EUROPEAN TELECOMMUNICATION STANDARD

ETS 300 607-3

October 1998

Eighth Edition

Source: SMG

Reference: RE/SMG-071110PR6-3

ICS: 33.020

Key words: Digital cellular telecommunications system, Global System for Mobile Communication (GSM)

GLOBAL SYSTEM FOR MOBILE COMMUNICATIONS

Digital cellular telecommunications system (Phase 2); Mobile Station (MS) conformance specification; Part 3: Layer 3 (L3) Abstract Test Suite (ATS) (GSM 11.10-3 version 4.22.1)

ETSI

European Telecommunications Standards Institute

ETSI Secretariat

Postal address: F-06921 Sophia Antipolis CEDEX - FRANCE
Office address: 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE
Internet: secretariat@etsi.org - http://www.etsi.org

Tel.: +33 4 92 94 42 00 - Fax: +33 4 93 65 47 16

Copyright Notification: No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 1998. All rights reserved.

Whilst every care has been taken in the preparation and publication of this document, errors in content, typographical or otherwise, may occur. If you have comments concerning its accuracy, please write to "ETSI Standards Making Support Dept." at the address shown on the title page.

Contents

Foreword 7

1 Scope 9

2 Normative references 9

3 Definitions and abbreviations 12
3.1 Definitions 12
3.2 Abbreviations 12

4 Test Suite Structure (TSS) 13
4.1 Test suite naming convention 13
4.2 Suite Overview 14
4.3 Test groups 14
4.3.1 Valid Behaviour tests (BV) 14
4.3.2 Invalid Behaviour/Inopportune Behaviour tests (BI-BO) 14
4.4 Test Step Structure 15
4.4.1 Preambles 15
4.4.2 Postambles 15

5 Test Purposes (TP) 15
5.1 TP and test case naming convention 15

6 Abstract test method and test configurations 15
6.1 Test system model 15
6.2 Test Method 16

Annex A (normative): Layer 3 Abstract Test Suite 17

Suite overview 17
Suite structure 17
Test case index 18
Test step index 28
Default index 36

Declarations Part 37
Test suite type definitions 37
Simple type definitions 37
Structured Type definitions 39
ASN1 type definitions 63
Test suite operation definitions 88
Test suite parameter declarations 146
Test case selection expression definitions 166
Test suite constant declarations 169
Test case variable declarations 189
PCO declarations 194
Timer declarations 194
ASP type definitions 195
TTCN ASP Type definitions 195
PDU type definitions 222
TTCN PDU Type definitions 222

Constraints Part 249
Test suite type constraint declarations 249
Structured type constraint declarations 249
ASN1 type constraints 506

ASP constraint declarations 520
TTCN ASP constraint declarations 520
PDU constraint declarations 552
TTCN PDU constraint declarations 552

Dynamic Part 662
Test Cases 662
Test Group General 662
Test Group InitialTest 677
Test Group IdleMode 703
Test Group BiBo 706
Test Group RR 744
Test Group MM 907
Test Group CC 971
Test Group StructureProc 1078
Test Group EGSMSignalling 1100
Test Group SS 1123
Test Group SM 1216
Test Step Library 1244
Test Step Group management 1244
Test Step Group Miscellaneous 1272
Test Step Group OperatorOP 1329
Test Step Group Postambles 1330
Test Step Group Preambles 1333
Defaults Library 1429

Annex B (normative): Partial IXIT proforma 1437

Introduction 1437

B.1 Identification Summary 1438

B.2 Abstract Test Suite Summary 1438

B.3 Test Laboratory 1438
B.3.1 Test Laboratory Identification 1438
B.3.2 Accreditation status of the test service 1439
B.3.3 Manager of Test Laboratory 1439
B.3.4 Contact person of Test Laboratory 1439
B.3.5 Means of Testing 1439
B.3.6 Instructions for Completion 1440

B.4 Client 1441
B.4.1 Client Identification 1441
B.4.2 Client Test Manager 1442
B.4.3 Client Contact person 1442
B.4.4 Test Facilities Required 1442

B.5 System Under Test 1443
B.5.1 SUT Information 1443
B.5.2 Limitations of the SUT 1444
B.5.3 Environmental Conditions 1445

B.6 Ancillary Protocols 1446
B.6.1 Ancillary Protocols 1 1447
B.6.2 Ancillary Protocols 2 1447

B.7 Protocol Layer Information for L3 of Mobile Station 1447
B.7.1 Information provided for test purposes by the MS supplier 1447
B.7.2 MMI information 1460
B.7.3 Test house specified parameters 1465

Annex C (normative): PCTR Proforma 1497

Annex D (normative): Enhanced Full Rate Abstract Test Suite 1498

Suite overview 1499
Suite structure 1499
Test case index 1500
Test step index 1501
Default index 1504

Declarations Part 1505
Test suite type definitions 1505
Simple type definitions 1505
Structured Type definitions 1507
ASN1 type definitions 1531
Test suite operation definitions 1556
Test suite parameter declarations 1614
Test case selection expression definitions 1634
Test suite constant declarations 1635
Test case variable declarations 1654
PCO declarations 1659
Timer declarations 1659
ASP type definitions 1659
TTCN ASP Type definitions 1659
PDU type definitions 1687
TTCN PDU Type definitions 1687

Constraints Part 1714
Test suite type constraint declarations 1714
Structured type constraint declarations 1714
ASN1 type constraints 1766
ASP constraint declarations 1780
TTCN ASP constraint declarations 1780
PDU constraint declarations 1795
TTCN PDU constraint declarations 1795

Dynamic Part 1823
Test Cases 1823
Test Group EFR 1823
Test Step Library 1886
Test Step Group management 1886
Test Step Group ChConfig 1886
Test Step Group Miscellaneous 1904
Test Step Group SysInfo 1918
Test Step Group OperatorOP 1930
Test Step Group Postambles 1931
Test Step Group Preambles 1933
Defaults Library 1951

Annex E (informative): Cell Selection Abstract Test Suite 1958

Suite overview 1958
Suite structure 1958
Test case index 1959
Test step index 1960
Default index 1963

Declarations Part 1964
Test suite type definitions 1964
Simple type definitions 1964
Structured Type definitions 1966
ASN1 type definitions 1990
Test suite operation definitions 2015
Test suite parameter declarations 2074
Test case selection expression definitions 2094

Test suite constant declarations 2097
Test case variable declarations 2117
PCO declarations 2122
Timer declarations 2122
ASP type definitions 2122
TTCN ASP Type definitions 2122
PDU type definitions 2150
TTCN PDU Type definitions 2150

Constraints Part 2177
Test suite type constraint declarations 2177
Structured type constraint declarations 2177
ASN1 type constraints 2247
ASP constraint declarations 2261
TTCN ASP constraint declarations 2261
PDU constraint declarations 2285
TTCN PDU constraint declarations 2285

Dynamic Part 2348
Test Cases 2348
Test Group CellSelection 2348
Test Step Library 2386
Test Step Group management 2386
Test Step Group ChConfig 2387
Test Step Group Miscellaneous 2400
Test Step Group SysInfo 2411
Test Step Group OperatorOP 2432
Test Step Group Postambles 2432
Test Step Group Preambles 2434
Defaults Library 2464

History 2470

Foreword

This European Telecommunication Standard (ETS) has been produced by the Special Mobile Group (SMG) Technical Committee of the European Telecommunications Standards Institute (ETSI).

This ETS describes the technical characteristics and methods of test for Mobile Stations (MSs), operating in the 900 MHz and 1 800 MHz frequency band (GSM 900 and DCS 1 800) within the digital cellular telecommunications system.

This ETS corresponds to GSM technical specification GSM 11.10-3 version 4.22.1.

This part of the ETS (Part 3), contains Tree and Tabular Combined Notation (TTCN) for Layer 3, signalling for MS supporting Enhanced Full Rate Speech, and cell selection conformity specifications for which Mobile Stations, within the digital cellular telecommunications system (Phase 2), are tested for compliance.

The graphical form ATS

The TTCN.GR representation of these ATSs are contained in annex A, annex D and annex E of this ETS.

The machine processable ATS

The electronic forms of the machine processable files (TTCN MP format) corresponding to the ATS for Layer 3 are contained in the file L3-4m1.mp (Layer 3), EF-4m1.mp (Enhanced Full Rate), and CS-4m1.mp (Cell Selection) **1)** on the diskettes included as a part of this ETS (Part 3).

ETS 300 607 consists of three parts, which have the following ETS numbers and titles:

ETS 300 607-1 Digital cellular telecommunications system (Phase 2);
Mobile Station (MS) conformance specification;
Part 1: Conformance specification

Reference: GSM 11.10-1.

ETS 300 607-2 Digital cellular telecommunications system (Phase 2);
Mobile Station (MS) conformance specification;
Part 2: Protocol Implementation Conformance Statement (ICS) proforma specification

Reference: GSM 11.10-2.

**ETS 300 607-3 Digital cellular telecommunications system (Phase 1);
Mobile Station (MS) conformance specification;
Part 3: Layer 3 (L3) Abstract Test Suite (ATS)**

Reference: GSM 11.10-3.

Transposition dates	
Date of adoption of this ETS:	23 October 1998
Date of latest announcement of this ETS (doa):	31 January 1999
Date of latest publication of new National Standard or endorsement of this ETS (dop/e):	31 July 1999
Date of withdrawal of any conflicting National Standard (dow):	31 July 1999

1) These files are located in an archive file named 6073_E8.ZIP. Other file formats are available on request.

Blank page

1 Scope

This European Telecommunication Standard (ETS) specifies the Abstract Test Suite (ATS) and partial IXIT proforma for the Network Layer (Layer 3) at the mobile radio interface of the GSM or DCS (Phase 2) mobile stations (MS) conforming to the ETSs for the Radio Resource management, the Mobility Management, the circuit-switched Call Control, the Supplementary Services and Short Message Services for the digital cellular telecommunications systems (Phase 2).

The ISO standards for the methodology of conformance testing are used as the basis for the test specifications.

2 Normative references

This ETS incorporates by dated and undated reference, provisions from other publications. These normative references are cited at the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this ETS only when incorporated in it by amendment or revision. For undated references, the latest edition of the publication referred to applies.

[1] ISO/IEC 9646-1: "Information Technology-OSI- Conformance Testing Methodology and Framework, Part 1: General Concepts".

[2] ISO/IEC 9646-2: "Information Technology-OSI- Conformance Testing Methodology and Framework, Part 2: Abstract Test Suite Specification".

[3] ISO/IEC 9646-3: "Information Technology-OSI- Conformance Testing Methodology and Framework, Part 3: The Tree and Tabular Combined Notation".

[4] ISO/IEC 9646-5: "Information Technology-OSI- Conformance Testing Methodology and Framework, Part 5: Requirements on test laboratories and clients for the conformance assessment process".

[5] ISO/IEC 8824: "Information Technology-OSI- Specification of Abstract Syntax Notation One (ASN.1)".

[6] ISO/IEC 8825: "Information Technology-OSI- Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)".

[7] ITU-T Recommendation Q.773: "Specifications of Signalling System No.7; Transaction capabilities formats and encoding".

[8] ETS 300 287: "Integrated Services Digital Network (ISDN); Signalling System No.7, Transaction Capabilities Application Part (TCAP) version 2".

[9] ETS 300 406 (January 1995): "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

[10] ETS 300 501 (GSM 02.02 Version 4.2.2): "Digital cellular telecommunication system (Phase 2); Bearer Services (BS) supported by a GSM Public Land Mobile Network (PLMN)".

[11] ETS 300 502 (GSM 02.03 Version 4.3.1): "Digital cellular telecommunication system (Phase 2); Teleservices supported by a GSM Public Land Mobile Network (PLMN)".

[12] ETS 300 504 Edition 4 (GSM 02.06 Version 4.5.1): "Digital cellular telecommunication system (Phase 2); Types of Mobile Stations (MS)".

[13] ETS 300 505 Edition 3 (GSM 02.07 Version 4.8.1): "Digital cellular telecommunication system (Phase 2); Mobile Station (MS) features".

[14] ETS 300 511 Edition 2 (GSM 02.30 Version 4.13.0): "Digital cellular telecommunication system (Phase 2); Man-Machine Interface (MMI) of the Mobile Station (MS)".

[15] ETS 300 523 Edition 2 (GSM 03.03 Version 4.9.0): "Digital cellular telecommunication system (Phase 2); Numbering, addressing and identification".

[16] ETS 300 527 Edition 2 (GSM 03.09 Version 4.6.0): "Digital cellular telecommunications system (Phase 2); Handover procedures".

[17] ETS 300 528 (GSM 03.10 Version 4.3.1): "Digital cellular telecommunications system (Phase 2); GSM Public Land Mobile Network (PLMN) connection types".

[18] ETS 300 529 Edition 3 (GSM 03.11 Version 4.10.1): "Digital cellular telecommunication system (Phase 2); Technical realization of supplementary services".

[19] ETS 300 530 (GSM 03.12 Version 4.4.2): "Digital cellular telecommunications system (Phase 2); Location registration procedures".

[20] ETS 300 532 (GSM 03.14 Version 4.1.1): "Digital cellular telecommunications system (Phase 2); Support of Dual Tone Multi-Frequency signalling (DTMF) via the GSM system".

[21] ETS 300 535 Edition 4 (GSM 03.22 Version 4.11.0): "Digital cellular telecommunication system (Phase 2); Functions related to Mobile Station (MS) in idle mode".

[22] ETS 300 536 Edition 4 (GSM 03.40 Version 4.13.0): "Digital cellular telecommunication system (Phase 2); Technical realization of the Short Message Service (SMS) Point to Point (PP)".

[23] ETS 300 537 Edition 2 (GSM 03.41 Version 4.11.0): "Digital cellular telecommunication system (Phase 2); Technical realization of Short Message Service Cell Broadcast (SMSCB)".

[24] ETS 300 550 (GSM 04.01 Version 4.0.4): "Digital cellular telecommunications system (Phase 2); Mobile Station - Base Station System (MS - BSS) interface General aspects and principles".

[25] ETS 300 552 (GSM 04.03 Version 4.1.1): "Digital cellular telecommunications system (Phase 2); Mobile Station - Base Station System (MS - BSS) interface Channel structures and access capabilities".

[26] ETS 300 553 (GSM 04.04 Version 4.0.4): "Digital cellular telecommunication system (Phase 2); layer 1 General requirements".

[27] ETS 300 554 (GSM 04.05 Version 4.0.3): "Digital cellular telecommunication system (Phase 2); Data Link (DL) layer General aspects".

[28] ETS 300 555 (GSM 04.06 Version 4.4.0): "Digital cellular telecommunication system (Phase 2); Mobile Station - Base Station System (MS - BSS) interface Data Link (DL) layer specification".

[29] ETS 300 556 (GSM 04.07 Version 4.3.1): "Digital cellular telecommunication system (Phase 2); Mobile radio interface signalling layer 3 General aspects".

[30] ETS 300 557 Edition 9 (GSM 04.08 Version 4.19.1): "Digital cellular telecommunication system (Phase 2); Mobile radio interface layer 3 specification".

[31] ETS 300 558 Edition 2 (GSM 04.10 Version 4.10.1): "Digital cellular telecommunication system (Phase 2); Mobile radio interface layer 3 Supplementary services specification General aspects".

[32] ETS 300 559 Edition 4 (GSM 04.11 Version 4.10.0): "Digital cellular telecommunication system (Phase 2); Point-to-Point (PP) Short Message Service (SMS) support on mobile radio interface".

[33] ETS 300 560 Edition 3 (GSM 04.12 Version 4.6.0): "Digital cellular telecommunication system (Phase 2); Short Message Service Cell Broadcast (SMSCB) support on the mobile radio interface".

[34] ETS 300 564 Edition 3 (GSM 04.80 Version 4.11.1): "Digital cellular telecommunication system (Phase 2); Mobile radio interface layer 3 supplementary services specification Formats and coding".

[35] ETS 300 565 (GSM 04.81 Version 4.4.1): "Digital cellular telecommunication system (Phase 2); Line identification supplementary services - Stage 3".

[36] ETS 300 566 Edition 2 (GSM 04.82 Version 4.9.1): "Digital cellular telecommunication system (Phase 2); Call Forwarding (CF) supplementary services - Stage 3".

[37] ETS 300 567 Edition 2 (GSM 04.83 Version 4.6.1): "Digital cellular telecommunication system (Phase 2); Call Waiting (CW) and Call Hold (HOLD) supplementary services - Stage 3".

[38] ETS 300 568 (GSM 04.84 Version 4.3.2): "Digital cellular telecommunication system (Phase 2); MultiParty (MPTY) supplementary services - Stage 3".

[39] ETS 300 569 Edition 2 (GSM 04.85 Version 4.1.1): "Digital cellular telecommunications system (Phase 2); Closed User Group (CUG) supplementary services - Stage 3".

[40] ETS 300 570 (GSM 04.86 Version 4.5.2): "Digital cellular telecommunication system (Phase 2); Advice of Charge (AoC) supplementary services - Stage 3".

[41] ETS 300 571 Edition 2 (GSM 04.88 Version 4.7.1): "Digital cellular telecommunication system (Phase 2); Call Barring (CB) supplementary services - Stage 3".

[42] ETS 300 572 (GSM 04.90 Version 4.1.1): "Digital cellular telecommunication system (Phase 2); Unstructured supplementary services operation - Stage 3".

[43] ETS 300 573 Edition 4 (GSM 05.01 Version 4.6.0): "Digital cellular telecommunications system (Phase 2); Physical layer on the radio path General description".

[44] ETS 300 574 Edition 6 (GSM 05.02 Version 4.9.0): "Digital cellular telecommunication system (Phase 2); Multiplexing and multiple access on the radio path".

[45] ETS 300 575 Edition 4 (GSM 05.03 Version 4.5.0): "Digital cellular telecommunication system (Phase 2); Channel coding".

[46] ETS 300 576 (GSM 05.04 Version 4.0.3): "Digital cellular telecommunication system (Phase 2); Modulation".

[47] ETS 300 577 Edition 12 (GSM 05.05 Version 4.20.0): "Digital cellular telecommunication system (Phase 2); Radio transmission and reception".

[48] ETS 300 578 Edition 10 (GSM 05.08 Version 4.19.0): "Digital cellular telecommunication system (Phase 2); Radio subsystem link control".

[49] ETS 300 579 Edition 6 (GSM 05.10 Version 4.9.0): "Digital cellular telecommunication system (Phase 2); Radio subsystem synchronisation".

[50] ETS 300 582 Edition 4 (GSM 07.01 Version 4.10.0): "Digital cellular telecommunication system (Phase 2); General on Terminal Adaptation Functions (TAF) for Mobile Stations (MS)".

[51] ETS 300 590 Edition 5 (GSM 08.08 Version 4.11.0): "Digital cellular telecommunications system (Phase 2); Mobile-services Switching Centre - Base Station System (MSC - BSS) interface Layer 3 specification".

[52] ETS 300 596 Edition 4 (GSM 08.58 Version 4.9.0): "Digital cellular telecommunications system (Phase 2); Base Station Controller - Base Transceiver Station (BSC - BTS) interface Layer 3 specification".

[53] ETS 300 599 Edition 7 (GSM 09.02 Version 4.17.0): "Digital cellular telecommunication system (Phase 2); Mobile Application Part (MAP) specification".

[54] ETS 300 607-1 Edition 7 (GSM 11.10-1 Version 4.20.0): "Digital cellular telecommunications system (Phase 2); Mobile Station (MS) conformance specification; Part 1: Conformance specification".

[55] ETS 300 607-2 Edition 2 (GSM 11.10-2 Version 4.15.0): "Digital cellular telecommunications system (Phase 2); Mobile Station (MS) conformance specification; Part 2: Protocol Implementation Conformance Statement (PICS) Proforma specification".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of this ETS, the following definitions apply:

Abstract Test Suite (ATS): Refer to ISO/IEC 9646-1 [1].

Implementation Under Test (IUT): Refer to ISO/IEC 9646-1 [1].

Partial Protocol Implementation eXtra Information for Testing (IXIT): Refer to ISO/IEC 9646-1 [1].

Point of Controls and Observations (PCO): Refer to ISO/IEC 9646-1 [1].

Protocol Implementation Conformance Statement (ICS): Refer to ISO/IEC 9646-1 [1].

System Under Test (SUT): Refer to ISO/IEC 9646-1 [1].

3.2 Abbreviations

For the purposes of this ETS, the following abbreviations apply:

ATS	Abstract Test Suite
BI	Invalid Behaviour tests
BO	Inopportune Behaviour tests
BV	Valid Behaviour tests
CA	CApability tests
EDP-N	Event Detection Point - Notification
EDP-R	Event Detection Point - Request
ETS	European Telecommunication Standard
FE	Functional Entity
FSM	Finite State Machine
ICS	Implementation Conformance Statement
IUT	Implementation Under Test

IXIT	Implementation eXtra Information for Testing
PDU	Protocol Data Unit
SUT	System Under Test
TP	Test Purpose
TSS	Test Suite Structure

Further abbreviations used within GSM related ETS may be found in ETR 100.

Abbreviations for test case group names may be found in figure 1 and 2.

4 Test Suite Structure (TSS)

4.1 Test suite naming convention

The test group identifier for each group and subgroup is built according to the scheme in figure 1.

Identifier: L3<c><s><g0><g1><g2><nn>

<c> = category:	BIT	BIT, Basic Interconnection tests (not used)
	CA	CA, Capability tests (not used)
	BV	BV, Valid Behaviour tests
	BI-BO	BI and BO, Invalid and Inopportune Behaviour tests
<g0> = group:	IN	Initial tests
	ID	Idle mode tests
	RR	Radio Resource management
	MM	Mobility Management
	CC	Call Control
	SP	Structured Procedures
	SS	Supplementary services
	LLF	Lower Layer Failures
	UUE	Unknown, Unforeseen or erroneous Elements
<g1> = group:	CR	Channel Request
	AT	IMSI detach and IMSI attach
	SMT	Sequenced MM / CC message transfer
	EC	Establishment Cause
	IA	Immediate assignment
	PG	Test of paging
	MR	Test of measurement report
	ASS	Test of the channel
	HO	Test of handover
	FR	Test of frequency redefinition
	CMM	Test of the channel mode modify procedure
	CY	Test of ciphering mode setting
	CM	Test of classmark
	CHR	Test of channel release
	START	Test of starting time
	IDAU	Identification and authentication
	LU	Location updating
	CON	MM connection
	CCSMO	State Machine Verification, Mobile Originating Call
	CCSMT	State Machine Verification, Mobile Terminating Call
	SMICF	State machine verification, In Call Functions
	CRE	Call Re-establishment
	UUS	User to user signalling
	UPD	Unknown protocol discriminator
	TIS	TI and skip indicator
	UMT	Undefined or unexpected message type
	UIE	Unforeseen information elements in the non-imperative
	NMIE	Non-semantical mandatory IE errors
	CNR	Unknown IE, comprehension not required
	SB	Spare bits

<g2> = group:	AC Handover / successful / active call
	CUE Handover / successful / call under establishment
	FSY Handover / successful / active call / finely synchronized
	PRS Pre-synchronized handovers
	PRF Protocol failures
	ACC Location updating / accepted
	REJ Location updating / rejected
	ABN Location updating / abnormal cases
	REL Location updating / release
	PER Location updating / periodic
	HPER Location updating / periodic HPLMN search
	IWAT Location updating / interworking of attach and periodic
	EST MM connection / establishment
	EXP MM connection / expiry
	NWAB MM connection / abortion by the network
	FRQP MM connection / follow-on request pending
	U0 Mobile Originating Call U0 State
	U0.1 Mobile Originating Call U0.1 MM Connection pending
	U1 Mobile Originating Call U1 State
	U3 Mobile Originating Call U3 State
	U4 Mobile Originating Call U4 State
	U10 Mobile Originating Call U10 State
	U11 Mobile Originating Call U11 State
	U12 Mobile Originating Call U12 State
	U19 Mobile Originating Call U19 State
	U0 Mobile Terminating Call U0 State
	U6 Mobile Terminating Call U6 State
	U9 Mobile Terminating Call U9 State
	U7 Mobile Terminating Call U7 State
	U8 Mobile Terminating Call U8 State
	DTMF State machine verification, In Call Functions / transfer
	CHC State machine verification, In Call Functions /
	TICM State machine verification, In Call Functions / in-
	OICM State machine verification, In Call Functions / in-

Figure 1: Test group identifier naming convention scheme

4.2 Suite Overview

Figure 2 shows the structure of the test suites for L3.

L3 ATS									
BV								BI/O	
IN	ID	RR	MM	CC	SP	SS	SMS	LLF	UUE

Figure 2: Test suite structure of the L3 tests

4.3 Test groups

4.3.1 Valid Behaviour tests (BV)

Predefined state transitions are considered as valid. The test purposes in the valid behaviour test subgroup cover the verification of the normal and exceptional procedures of the various Finite State Machines (FSMs), i.e. a valid behaviour test is a test where the message sequence and the message contents are considered as valid.

4.3.2 Invalid Behaviour/Inopportune Behaviour tests (BI-BO)

This test sub group verifies that the Implementation Under Test (IUT) is able to react properly having received an invalid Protocol Data Unit (PDU) or in the case an inopportune protocol event occurs. An invalid PDU is defined as a syntactically incorrect message. An inopportune event is syntactically correct but occurs when it is not expected, e.g. a correctly coded operation is received in a wrong state (the IUT may respond Error UnexpectedComponentSequence).

4.4 Test Step Structure

4.4.1 Preambles

The preamble is defined for each test purpose.

4.4.2 Postambles

After each test case the IUT shall be brought to the state as defined in the postamble for each test purpose.

5 Test Purposes (TP)

For each conformance requirement a Test Purpose (TP) is defined. The test purposes are specified in the ATS Dynamic part (annex A).

5.1 TP and test case naming convention

In order clearly to map the conformance requirements specified in the prETS 300 607-1 and TTCN test cases in the ATS, the section numbers in the prETS 300 607-1 are used as test case names.

The identifier of each TP is identical to the name of the implemented TTCN test case.

"Test Purpose Identifier" = "Test Case Name" = TPI = TC_NN_A_B_C_D_E, where NN, A, B, C, D and E are digits used in the corresponding section numbers of prETS 300 607-1. For example, the test case name TC_26_5_6_2 is the TTCN specification corresponding to the conformance requirements in the section 26.5.6.1.2 in the prETS 300 607-1. In case where the section has been implemented in more than one test case sub numbering has been introduced. For example, the section 26.7.4.2.4 of prETS 300 607-1, Location updating/rejected/national roaming testing is split into 5 test cases. The corresponding TCs are identified as TC_26_7_4_2_4_1, TC_26_7_4_2_4_2, TC_26_7_4_2_4_3, TC_26_7_4_2_4_4 and TC_26_7_4_2_4_5, respectively.

6 Abstract test method and test configurations

The distributed test method applies to the L3 MS testing. The test method uses a lower tester and a Man-Machine Interface (MMI) as an upper tester at the SUT.

6.1 Test system model

The model of the L3 test system is based on the original protocol architecture at the air interface. The test system consists conceptually of a lower tester LT, the L3 test programme (executable test suite), a L2 radio link emulator, a management functional unit, the L1 service provider and a TRx set (see figure 1).

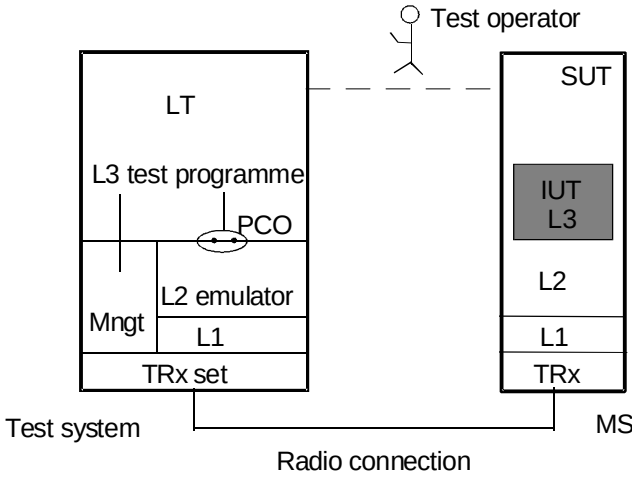

Figure 3: Test system and distributed test method

The LT provides the test environment and for test execution and the means of control and observation at the L3 lower service boundary within the test system.

The L3 TTCN test specification uses the three LT interfaces to communicate with the MS, the system under test, and with the other parts of the test system:

- Interface to the L2 emulator via the PCO;
- Interface to the management functional unit via TTCN test suite operations;
- Interface to the MS Man-Machine Interface (MMI) via a test operator.

6.2 Test Method

The PCO in the LT is defined as L2 SAP (SAP 0 + 3). The PCO has two FIFO queues (data buffers) to store all sending and receiving test events. The L2 primitives in the ATS which constitute mainly the interface to the L2 emulator are specified via the L2 primitives. In order to simulate multicell testing as required in some test cases, the defined primitives are able to address individual cells of the test system and the logic channels of each cell for the L3 message exchanges. The L2 emulator together with the underlying L1 and the TRx set support all message exchanges via correct radio links.

The management function unit has three management functions:

- L2 and L1 management;
- Channel management;
- TRx management;

The interface to the management function unit is presented in the ATS via a set of test suite operations. The major functions of the test suite management operations are:

- To load configuration parameters necessary for the test system.
- To control and get the necessary values of radio resources/ channels for tests.

The SUT (MS) has a more or less standardized MMI, such as keys, digital display, tones, etc. The ATS uses such kind of functions to provoke some procedures or to observe simple results at the SUT side. A human operator is needed during the test. The test system shall have an interface to the human operator to enable the test co-ordination.

Annex A (normative): Layer 3 Abstract Test Suite

The machine processable ATS

The electronic forms of the machine processable files (TTCN MP format) corresponding to the ATS for EFR are contained in the file l3-4m1.mp[2)] (Layer 3) on the diskette included as a part of this ETS (Part 3).

The graphical form ATS

The TTCN.GR representation of this ATS is contained in the remainder of this annex.

Suite overview

Suite structure

Test Suite Structure	
Suite Name:	GSM_L3_MS_v4220
Standards Ref:	ETS 300 557
PICS Ref:	
PIXIT Ref:	
Test Method(s):	Distributed Single Layer Test Method
Comments:	Version 4.22.0.

Structure and Objectives			
Test Group Reference	**Selection Ref**	**Test Group Objective**	**Page Nr**
GSM_L3_MS_v4220/General/	SelExpr_0000	To verify the supported and non-supported services.	649
GSM_L3_MS_v4220/InitialTest/	SelExpr_0100	To verify random access procedure, IMSI attatch and detach procedure, sequenced MM/CC message transfer and establishment causes.	664
GSM_L3_MS_v4220/IdleMode/	SelExpr_0200	To verify MS functions in idle mode	690
GSM_L3_MS_v4220/BiBo/	SelExpr_0300	To verify the MS handling unknown, unforeseen and erroneous protocol data, and parallel transactions	693
GSM_L3_MS_v4220/RR/	SelExpr_0400	To verify the elementary procedures for radio resource management	731
GSM_L3_MS_v4220/MM/	SelExpr_0500	To verify the elementary procedures of mobility management.	894
GSM_L3_MS_v4220/CC/	SelExpr_0600	To verify the circuit switched call cntrol functions.	959
GSM_L3_MS_v4220/StructureProc/	SelExpr_0700	To verify the structured procedures.	1066
GSM_L3_MS_v4220/EGSMSignalling/	SelExpr_1000	To verify the different procedures which may be impacted when some channel uses E- GSM frequencies.	1088
GSM_L3_MS_v4220/SS/	SelExpr_0800	To verify the functions of supplementary services.	1111
GSM_L3_MS_v4220/SM/	SelExpr_0900	To verify the functions of short message service.	1204
Detailed Comments:			

2) This file is located in an archive file named 6073_E8.ZIP. Other file formats are available on request.

Test case index

Test Case Index				
Test Group Reference	**Test Case Id**	**Selection ref**	**Description**	**Page Nr**
GSM_L3_MS_v4220/ General/	TC_11_1_1	SelExpr_0002	Verification of support and non-support of services (MT).	649
GSM_L3_MS_v4220/ General/	TC_11_1_2	SelExpr_0007	Verification of support and non-support of services (MO).	651
GSM_L3_MS_v4220/ General/	TC_11_2	SelExpr_0002	Verification of support of the single numbering scheme.	656
GSM_L3_MS_v4220/ General/	TC_11_3	SelExpr_0003	Verification of non-support of services. (Advice of Charge Charging, AOCC)	657
GSM_L3_MS_v4220/ General/	TC_11_4	SelExpr_0004	Verification of non-support of services. (Call Hold)	660
GSM_L3_MS_v4220/ General/	TC_11_5	SelExpr_0005	Verification of non-support of services. (MultiParty)	661
GSM_L3_MS_v4220/ General/	TC_11_6	SelExpr_0006	Verification of non-support of feature. (Fixed dialling number)	663
GSM_L3_MS_v4220/InitialTest/	TC_26_2_1_1	SelExpr_0101	Initial Layer 3 tests - Channel request / initial time. 11.10 Ref. ver.4.10; CR	664
GSM_L3_MS_v4220/InitialTest/	TC_26_2_1_2	SelExpr_0101	Initial Layer 3 tests - Channel request / repetition time. 11.10 Ref. ver.4.10; CR: C46	666
GSM_L3_MS_v4220/InitialTest/	TC_26_2_1_3	SelExpr_0101	Initial Layer 3 tests - Channel request / random reference. 11.10 Ref. ver.4.10; CR	668
GSM_L3_MS_v4220/InitialTest/	TC_26_2_2	SelExpr_0101	IMSI detach and IMSI attach. 11.10 Ref. ver 4.10.0; CR. 0295-6r1. CR.C62r1.	669
GSM_L3_MS_v4220/InitialTest/	TC_26_2_3	SelExpr_0101	Sequenced MM / CM message transfer.	671
GSM_L3_MS_v4220/InitialTest/	TC_26_2_4_1	SelExpr_0102	Establishment Cause /pr1 (TCH)	673
GSM_L3_MS_v4220/InitialTest/	TC_26_2_4_2	SelExpr_0103	Establishment Cause /pr2 (/H)	675
GSM_L3_MS_v4220/InitialTest/	TC_26_2_4_3	SelExpr_0107	Establishment Cause /pr3 (TCH/FS)	677
GSM_L3_MS_v4220/InitialTest/	TC_26_2_4_4	SelExpr_0104	Establishment Cause /pr4 (data)	679
GSM_L3_MS_v4220/InitialTest/	TC_26_2_4_5	SelExpr_0101	Establishment Cause /pr5	681
GSM_L3_MS_v4220/InitialTest/	TC_26_2_4_6	SelExpr_0101	Establishment Cause /pr6	684
GSM_L3_MS_v4220/InitialTest/	TC_26_2_4_7	SelExpr_0105	Establishment Cause /pr7 (non-call-SS)	687
GSM_L3_MS_v4220/InitialTest/	TC_26_2_4_8	SelExpr_0106	Establishment Cause /pr8 (SMS/PP MO)	689
GSM_L3_MS_v4220/IdleMode/	TC_26_3_2	SelExpr_0201	MS indication of available PLMNs	690
GSM_L3_MS_v4220/IdleMode/	TC_26_3_3	SelExpr_0201	MS will send only if BSS is "on air".	691
GSM_L3_MS_v4220/IdleMode/	TC_26_3_4	SelExpr_0201	Manual mode of PLMN selection 11.10 Ref. ver. 4.10.0; CR. 11.10-661. C64	692
GSM_L3_MS_v4220/ BiBo/	TC_26_5_1	SelExpr_0302	Handling of unknown protocol discriminator	693
GSM_L3_MS_v4220/ BiBo/	TC_26_5_2_1_1	SelExpr_0302	Handling of unknown TI and skip indicator / RR	694
GSM_L3_MS_v4220/ BiBo/	TC_26_5_2_1_2	SelExpr_0302	TI Skip indicator / RR / RR Connection established	695
GSM_L3_MS_v4220/ BiBo/	TC_26_5_2_2	SelExpr_0302	TI and skip indicator / MM	697
GSM_L3_MS_v4220/ BiBo/	TC_26_5_2_3	SelExpr_0301	TI and skip indicator / CC	699
GSM_L3_MS_v4220/ BiBo/	TC_26_5_3_1	SelExpr_0301	Undefined or unexpected Message type / undefined message type / CC	701
GSM_L3_MS_v4220/ BiBo/	TC_26_5_3_2	SelExpr_0301	Undefined or unexpected message type / undefined message type / MM	702

GSM_L3_MS_v4220/ BiBo/	TC_26_5_3_3	SelExpr_0302	Undefined or unexpected message type / undefined message type / RR	703
GSM_L3_MS_v4220/ BiBo/	TC_26_5_3_4	SelExpr_0301	Undefined or unexpected message type / unexpected message type / CC	704
GSM_L3_MS_v4220/ BiBo/	TC_26_5_4_1	SelExpr_0302	Unforeseen info elements in non-imperative message part / duplicated info elements	705
GSM_L3_MS_v4220/ BiBo/	TC_26_5_5_1_1_1	SelExpr_0302	Non-semantical mandatory IE errors / RR / missing mandatory IE error / special case	706
GSM_L3_MS_v4220/ BiBo/	TC_26_5_5_1_1_2	SelExpr_0302	Non-semantical mandatory IE errors / RR / missing mandatory IE error / general case	707
GSM_L3_MS_v4220/ BiBo/	TC_26_5_5_1_2	SelExpr_0302	Non-semantical mandatory ie errors / RR / comprehension required	708
GSM_L3_MS_v4220/ BiBo/	TC_26_5_5_2_1	SelExpr_0301	Non-semantical mandatory IE errors / MM / syntactically incorrect mandatory IE	709
GSM_L3_MS_v4220/ BiBo/	TC_26_5_5_2_2	SelExpr_0302	Non-semantical mandatory IE errors / MM / syntactically incorrect mandatory IE 11.10 Ref. ver. 4.10.0; CR. 11.10-688	710
GSM_L3_MS_v4220/ BiBo/	TC_26_5_5_2_3	SelExpr_0302	Non-semantical mandatory IE errors / MM / syntactically incorrect mandatory IE	711
GSM_L3_MS_v4220/ BiBo/	TC_26_5_5_3_1_1	SelExpr_0301	Non-semantical mandatory IE errors / CC / missing mandatory IE / disconnect message	713
GSM_L3_MS_v4220/ BiBo/	TC_26_5_5_3_1_2	SelExpr_0301	Non-semantical mandatory IE errors / CC / missing mandatory IE / general case	714
GSM_L3_MS_v4220/ BiBo/	TC_26_5_5_3_2	SelExpr_0301	Non-semantical mandatory IE errors / CC / comprehension required	715
GSM_L3_MS_v4220/ BiBo/	TC_26_5_6_1_1	SelExpr_0302	Unknown IE, comprehension not required / MM / IE unknown in the protocol	716
GSM_L3_MS_v4220/ BiBo/	TC_26_5_6_1_2	SelExpr_0302	Unknown IE, comprehension not required / MM / IE unknown in the message	718
GSM_L3_MS_v4220/ BiBo/	TC_26_5_6_2_1	SelExpr_0301	Unknown info elements in the non-imperative message part / CC / Call establishment	719
GSM_L3_MS_v4220/ BiBo/	TC_26_5_6_2_2	SelExpr_0301	Unknown information elements in the non-imperative message part / CC / disconnect	720
GSM_L3_MS_v4220/ BiBo/	TC_26_5_6_2_3	SelExpr_0301	Unknown information elements in the non-imperative message part / CC / release	721
GSM_L3_MS_v4220/ BiBo/	TC_26_5_6_2_4	SelExpr_0301	Unknown info elements in the non-imperative message part / CC / release complete	722
GSM_L3_MS_v4220/ BiBo/	TC_26_5_6_3	SelExpr_0302	Unknown IE in the non-imperative message part, comprehension not required / RR / unknown in the protocol.	723
GSM_L3_MS_v4220/ BiBo/	TC_26_5_7_1_1	SelExpr_0302	Spare bits / RR / paging channel	724
GSM_L3_MS_v4220/ BiBo/	TC_26_5_7_1_2	SelExpr_0302	Spare bits / RR / BCCH	725
GSM_L3_MS_v4220/ BiBo/	TC_26_5_7_1_3	SelExpr_0302	pare bits / RR / AGCH	726
GSM_L3_MS_v4220/ BiBo/	TC_26_5_7_1_4	SelExpr_0302	Spare bits / RR / connected mode	727
GSM_L3_MS_v4220/ BiBo/	TC_26_5_7_2	SelExpr_0302	Spare bits / MM	728
GSM_L3_MS_v4220/ BiBo/	TC_26_5_7_3	SelExpr_0301	Spare bits / CC	729
GSM_L3_MS_v4220/ RR/	TC_26_6_1_1	SelExpr_0401	Immediate Assignment / SDCCH or TCH assignment	731
GSM_L3_MS_v4220/ RR/	TC_26_6_1_2	SelExpr_0401	Immediate Assignment / extended assignment	733
GSM_L3_MS_v4220/ RR/	TC_26_6_1_3	SelExpr_0401	Immediate Assignment / assignment rejection	736
GSM_L3_MS_v4220/ RR/	TC_26_6_1_4	SelExpr_0401	Immediate Assignment / ignore assignment	739
GSM_L3_MS_v4220/ RR/	TC_26_6_1_5	SelExpr_0401	Immediate Assignment after immediate assignment reject	741
GSM_L3_MS_v4220/ RR/	TC_26_6_2_1_1	SelExpr_0401	Paging / normal / type 1	742
GSM_L3_MS_v4220/ RR/	TC_26_6_2_1_2	SelExpr_0401	Paging / normal / type 2	744
GSM_L3_MS_v4220/ RR/	TC_26_6_2_1_3	SelExpr_0401	Paging / normal / type 3	746
GSM_L3_MS_v4220/ RR/	TC_26_6_2_2	SelExpr_0401	Paging / extended	748
GSM_L3_MS_v4220/	TC_26_6_2_3_1	SelExpr_0401	Paging / reorganisation / procedure 1	752

RR/				
GSM_L3_MS_v4220/ RR/	TC_26_6_2_3_2	SelExpr_0401	Paging / reorganisation / procedure 2	754
GSM_L3_MS_v4220/ RR/	TC_26_6_2_4	SelExpr_0401	Paging / same as before	756
GSM_L3_MS_v4220/ RR/	TC_26_6_2_5	SelExpr_0401	Multislot CCCH handling	757
GSM_L3_MS_v4220/ RR/	TC_26_6_3_1	SelExpr_0402	Measurement / no neighbour	759
GSM_L3_MS_v4220/ RR/	TC_26_6_3_2	SelExpr_0402	Measurement / all neighbours present	761
GSM_L3_MS_v4220/ RR/	TC_26_6_3_3	SelExpr_0402	Measurement / barred cells and non-permitted NCCs	763
GSM_L3_MS_v4220/ RR/	TC_26_6_3_4	SelExpr_0402	Measurement / DTX	765
GSM_L3_MS_v4220/ RR/	TC_26_6_3_5	SelExpr_0402	Measurement / frequency formats	769
GSM_L3_MS_v4220/ RR/	TC_26_6_4_1	SelExpr_0401	Dedicated assignment / Successful case	772
GSM_L3_MS_v4220/ RR/	TC_26_6_4_2_1	SelExpr_0402	Dedicated assignment / failure / failure during active state	778
GSM_L3_MS_v4220/ RR/	TC_26_6_4_2_2	SelExpr_0401	Dedicated assignment / failure / genaral case	780
GSM_L3_MS_v4220/ RR/	TC_26_6_5_1	SelExpr_0402	Handover / successful / active call / non-synchronized	782
GSM_L3_MS_v4220/ RR/	TC_26_6_5_2_1	SelExpr_0410	Retransmit L3-msg during non synchr. HO from SDCCH/4_nonFH to TCH/F_FH.	787
GSM_L3_MS_v4220/ RR/	TC_26_6_5_2_2	SelExpr_0408	Retransmit L3-msg during non synchr. HO from SDCCH/4_nonFH to TCH/H_FH.	789
GSM_L3_MS_v4220/ RR/	TC_26_6_5_2_3	SelExpr_0410	Retransmit L3-msg during non synchr. HO from SDCCH/4_nonFH to SDCCH/8_FH.	791
GSM_L3_MS_v4220/ RR/	TC_26_6_5_2_4	SelExpr_0410	Retransmit L3-msg during non synchr. HO from SDCCH/8_nonFH in cell A to SDCCH/8_FH in cellB.	793
GSM_L3_MS_v4220/ RR/	TC_26_6_5_2_5	SelExpr_0408	Retransmit L3-msg during non synchr. HO from TCH/F_NonFH in cell A to TCH/H_NonFH in cellB.	796
GSM_L3_MS_v4220/ RR/	TC_26_6_5_2_6	SelExpr_0408	Retransmit L3-msg during non synchr. HO from TCH/H_FH in cell A to TCH/F_FH in cellB.	798
GSM_L3_MS_v4220/ RR/	TC_26_6_5_2_7	SelExpr_0410	Retransmit L3-msg during non synchr. HO from TCH/F_FH in cell A to TCH/F_FH in cellB.	800
GSM_L3_MS_v4220/ RR/	TC_26_6_5_2_8	SelExpr_0410	Retransmit L3-msg during non synchr. HO from TCH/SDCCH8_FH in cell A to TCH/F_NonFH in cellB.	802
GSM_L3_MS_v4220/ RR/	TC_26_6_5_2_9	SelExpr_0410	Retransmit L3-msg during non synchr. HO from SDCCH8_NoFH in cell A to TCH/F_FH in cellB.	804
GSM_L3_MS_v4220/ RR/	TC_26_6_5_2_10	SelExpr_0408	Retransmit L3-msg during non synchr. HO from TCH/SDCCH8_NoFH in cell A to TCH/H_FH in cellB.	807
GSM_L3_MS_v4220/ RR/	TC_26_6_5_3_1	SelExpr_0402	synchronised HO from TCH/F_FH in cellA to TCH/F_nonFH in cellB.	809
GSM_L3_MS_v4220/ RR/	TC_26_6_5_3_2	SelExpr_0408	synchronised HO from TCH/H_FH in cellA to TCH/H_nonFH in cellB.	811
GSM_L3_MS_v4220/ RR/	TC_26_6_5_4_1	SelExpr_0410	Handover/ successful/ call under establishment/ finely synchronized (HO: SDCCH8/FH -> SDCCH/8_FH)	813
GSM_L3_MS_v4220/ RR/	TC_26_6_5_4_2	SelExpr_0410	Handover/ successful/ call under establishment/ finely synchronized (HO: SDCCH8/FH -> SDCCH/4_noFH)	815
GSM_L3_MS_v4220/ RR/	TC_26_6_5_4_3	SelExpr_0410	Handover/ successful/ call under establishment/ finely synchronized (HO: TCH/F_NoFH -> TCH/F_FH)	817
GSM_L3_MS_v4220/ RR/	TC_26_6_5_4_4	SelExpr_0410	Handover/ successful/ call under establishment/ finely synchronized (HO: SDCCH8_NoFH -> TCH/F_NoFH)	819
GSM_L3_MS_v4220/ RR/	TC_26_6_5_5_1	SelExpr_0402	Handover / successful / active call / pre-synchronised / Timing Advance IE not included	821

GSM_L3_MS_v4220/ RR/	TC_26_6_5_5_2	SelExpr_0410	Handover / successful / call being estab. / pre-synch. /Tim. Advance IE is included / reporting of obsrvd. time diff requested	824
GSM_L3_MS_v4220/ RR/	TC_26_6_5_6	SelExpr_0409	Handover / successful / active call / pseudo - synchronised	827
GSM_L3_MS_v4220/ RR/	TC_26_6_5_7	SelExpr_0402	Handover / successful / active call / non-synchronised / reporting of obsrvd. Time diff requested	830
GSM_L3_MS_v4220/ RR/	TC_26_6_5_8	SelExpr_0402	Handover / L3-failure 11.10 Ref. ver.4.10.0; CR 11.10-720	832
GSM_L3_MS_v4220/ RR/	TC_26_6_5_9	SelExpr_0402	Handover / L1-failure 11.10 Ref. ver.4.10.0; CR 11.10-720	834
GSM_L3_MS_v4220/ RR/	TC_26_6_6_1	SelExpr_0401	Frequency redefinition	837
GSM_L3_MS_v4220/ RR/	TC_26_6_7_1	SelExpr_0403	Channel mode modify / full rate	841
GSM_L3_MS_v4220/ RR/	TC_26_6_7_2	SelExpr_0404	Channel mode modify / half rate 11.10 Ref ver. 4.10.0;CR.0295-17r1; 11.10-780	844
GSM_L3_MS_v4220/ RR/	TC_26_6_8_1	SelExpr_0405	Ciphering mode / start ciphering	847
GSM_L3_MS_v4220/ RR/	TC_26_6_8_2	SelExpr_0402	Ciphering mode / no ciphering	849
GSM_L3_MS_v4220/ RR/	TC_26_6_8_3	SelExpr_0405	Ciphering mode / old cipher key	850
GSM_L3_MS_v4220/ RR/	TC_26_6_8_4	SelExpr_0401	Ciphering mode / Change of mode, algorithm and key	851
GSM_L3_MS_v4220/ RR/	TC_26_6_8_5	SelExpr_0401	Ciphering mode / IMEI request 11.10 REF. VER.4.10.0; CR. C92	858
GSM_L3_MS_v4220/ RR/	TC_26_6_11_1	SelExpr_0406	Test of Classmark change	859
GSM_L3_MS_v4220/ RR/	TC_26_6_11_2	SelExpr_0401	Test of Classmark Interrogation	861
GSM_L3_MS_v4220/ RR/	TC_26_6_12_1	SelExpr_0401	Channel release / SDCCH	862
GSM_L3_MS_v4220/ RR/	TC_26_6_12_2	SelExpr_0401	Channel release / SDCCH - no L2 ACK	864
GSM_L3_MS_v4220/ RR/	TC_26_6_12_3	SelExpr_0403	Channel release / TCH-F	866
GSM_L3_MS_v4220/ RR/	TC_26_6_12_4	SelExpr_0403	Channel release / TCH-F - no L2 ACK	868
GSM_L3_MS_v4220/ RR/	TC_26_6_13_1	SelExpr_0401	Dedicated assignment with starting time / successful case / time not elapsed	870
GSM_L3_MS_v4220/ RR/	TC_26_6_13_2	SelExpr_0401	Dedicated assignment with starting time / successful case / time elapsed	872
GSM_L3_MS_v4220/ RR/	TC_26_6_13_3	SelExpr_0401	Dedicated assignment with starting time and frequency redefinition/ failure case / time not elapsed	874
GSM_L3_MS_v4220/ RR/	TC_26_6_13_4	SelExpr_0401	Dedicated assignment with starting time and frequency redefinition/ failure case / time elapsed	876
GSM_L3_MS_v4220/ RR/	TC_26_6_13_5	SelExpr_0401	Handover with starting time / successful case / time not elapsed	878
GSM_L3_MS_v4220/ RR/	TC_26_6_13_6	SelExpr_0401	Handover with starting time / successful case / time elapsed	881
GSM_L3_MS_v4220/ RR/	TC_26_6_13_7	SelExpr_0401	Handover with starting time and frequency redefinition / failure case / time not elapsed	884
GSM_L3_MS_v4220/ RR/	TC_26_6_13_8	SelExpr_0401	Handover with starting time and frequency redefinition / failure case / time elapsed	887
GSM_L3_MS_v4220/ RR/	TC_26_6_13_9	SelExpr_0401	Immediate assignment with starting time / successful case / time not elapsed	890
GSM_L3_MS_v4220/ RR/	TC_26_6_13_10	SelExpr_0401	Immediate assignment with starting time / successful case / time elapsed	892
GSM_L3_MS_v4220/ MM/	TC_26_7_1	SelExpr_0501	TMSI reallocation.	894
GSM_L3_MS_v4220/ MM/	TC_26_7_2_1	SelExpr_0501	Authentication accepted.	895
GSM_L3_MS_v4220/ MM/	TC_26_7_2_2	SelExpr_0501	Authentication Reject	896
GSM_L3_MS_v4220/ MM/	TC_26_7_3_1	SelExpr_0501	General identification with IMSI and TMSI in non ciphered mode and IMEI in ciphered	898

			mode.	
GSM_L3_MS_v4220/ MM/	TC_26_7_3_2	SelExpr_0501	Handling of IMSI shorter than the maximum length.	899
GSM_L3_MS_v4220/ MM/	TC_26_7_4_1	SelExpr_0501	Location Updating Accepted	902
GSM_L3_MS_v4220/ MM/	TC_26_7_4_2_1	SelExpr_0501	Location Update Reject with the cause "IMSI unknown in HLR", "illegal MS" or "Illegal ME".	904
GSM_L3_MS_v4220/ MM/	TC_26_7_4_2_2_1	SelExpr_0501	Location Update Rejected with the cause "PLMN not allowed" - test 1.	906
GSM_L3_MS_v4220/ MM/	TC_26_7_4_2_2_2	SelExpr_0501	Location Update Rejected with the cause "PLM not allowed" - test 2.	908
GSM_L3_MS_v4220/ MM/	TC_26_7_4_2_3	SelExpr_0501	Location Update Rejected with the cause "Location area not allowed".	910
GSM_L3_MS_v4220/ MM/	TC_26_7_4_2_4_1	SelExpr_0501	Location Updating / rejected / roaming / procedure1	912
GSM_L3_MS_v4220/ MM/	TC_26_7_4_2_4_2	SelExpr_0501	Location Updating / rejected / roaming / procedure2	913
GSM_L3_MS_v4220/ MM/	TC_26_7_4_2_4_3	SelExpr_0501	Location Updating / rejected / roaming / procedure3	915
GSM_L3_MS_v4220/ MM/	TC_26_7_4_2_4_4	SelExpr_0501	Location Updating / rejected / roaming / procedure4	917
GSM_L3_MS_v4220/ MM/	TC_26_7_4_2_4_5	SelExpr_0502	Location Updating / rejected / roaming / procedure5	918
GSM_L3_MS_v4220/ MM/	TC_26_7_4_3_1	SelExpr_0501	Location Updating / abnormal cases / random access fails	919
GSM_L3_MS_v4220/ MM/	TC_26_7_4_3_2	SelExpr_0501	Location updating / abnormal cases / attempt counter less or equal to 4, LAI different	921
GSM_L3_MS_v4220/ MM/	TC_26_7_4_3_3	SelExpr_0501	Location Updating/Abnormal cases/Attempt Counter Equal to 4	925
GSM_L3_MS_v4220/ MM/	TC_26_7_4_3_4	SelExpr_0501	Location updating/Abnormal cases/Attempt Counter Equal to 4, stored LAI equal to broadcast LAI	929
GSM_L3_MS_v4220/ MM/	TC_26_7_4_4	SelExpr_0502	To verify that the MS aborts the RR-connection at the expiry of timer T3240.	934
GSM_L3_MS_v4220/ MM/	TC_26_7_4_5_1	SelExpr_0501	Periodic Updating Procedure/ Change of T3212 value.	935
GSM_L3_MS_v4220/ MM/	TC_26_7_4_5_2	SelExpr_0501	Periodic Updating Procedure/ Reset of T3212 after receiving of the first L3 message.	937
GSM_L3_MS_v4220/ MM/	TC_26_7_4_5_3	SelExpr_0501	Periodic Updating Procedure/ Reset of T3212 after generic LUP or IMSI attach procedure	939
GSM_L3_MS_v4220/ MM/	TC_26_7_4_5_4_1	SelExpr_0501	Location updating/ periodic HPLMN search/ MS waits time T.	940
GSM_L3_MS_v4220/ MM/	TC_26_7_4_5_4_2	SelExpr_0501	Location updating/ periodic HPLMN search/ MS in manual mode.	941
GSM_L3_MS_v4220/ MM/	TC_26_7_4_5_4_3	SelExpr_0501	Location updating/ periodic HPLMN search/ MS waits at least two minutes	942
GSM_L3_MS_v4220/ MM/	TC_26_7_4_6	SelExpr_0501	Location updating/ interworking of attach and periodic.	944
GSM_L3_MS_v4220/ MM/	TC_26_7_5_2	SelExpr_0501	MM connection/ establishment with cipher.	947
GSM_L3_MS_v4220/ MM/	TC_26_7_5_3	SelExpr_0501	MM connection/ establishment without cipher.	949
GSM_L3_MS_v4220/ MM/	TC_26_7_5_4	SelExpr_0501	MM connection/ establishment rejected.	950
GSM_L3_MS_v4220/ MM/	TC_26_7_5_5	SelExpr_0501	MM connection/ establishment rejected cause 4.	951
GSM_L3_MS_v4220/ MM/	TC_26_7_5_6	SelExpr_0501	MM connection/ expiry T3230	952
GSM_L3_MS_v4220/ MM/	TC_26_7_5_7_1	SelExpr_0501	MM connection/ abortion by the network/ cause #6	953
GSM_L3_MS_v4220/ MM/	TC_26_7_5_7_2	SelExpr_0504	MM connection/ abortion by the network/ cause not equal #6	955
GSM_L3_MS_v4220/ MM/	TC_26_7_5_8_1	SelExpr_0501	MM connection / follow-on request pending / test1	956
GSM_L3_MS_v4220/ MM/	TC_26_7_5_8_2	SelExpr_0501	MM connection / follow-on request pending / test2	957
GSM_L3_MS_v4220/ MM/	TC_26_7_5_8_3	SelExpr_0501	MM connection / follow-on request pending / test3	958
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_1_1	SelExpr_0601	Outgoing call/ / U10 null state.	959
GSM_L3_MS_v4220/	TC_26_8_1_2_2_1	SelExpr_0601	Outgoing call / U0.1 MM connection pending /	960

CC/			CM service rejected	
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_2_2	SelExpr_0601	Outgoing call / U0.1 MM connection pending / CM service accepted	961
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_2_3	SelExpr_0601	Outgoing call / U0.1 MM connection pending / lower layer failure	962
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_3_1	SelExpr_0601	Outgoing call / U1 call initiated / receiving CALL PROCEEDING	963
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_3_2	SelExpr_0601	Outgoing call / U1 call initiated / rejecting with RELEASE COMPLETE	964
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_3_3	SelExpr_0601	Outgoing call / U1 call initiated / T303 expiry	965
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_3_4	SelExpr_0601	Outgoing call / U1 call initiated / lower layer failure	966
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_3_5	SelExpr_0601	Outgoing call / U1 call initiated / receiving ALERTING	967
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_3_6	SelExpr_0601	Outgoing call / U1 call initiated / entering state U10	968
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_3_7	SelExpr_0601	Outgoing call / U1 call initiated / unknown message received	969
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_4_1	SelExpr_0601	Outgoing call / U3 MS originating call proceeding / ALERTING received	970
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_4_2	SelExpr_0601	Outgoing call / U3 MS originating call proceeding / CONNECT received	971
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_4_3	SelExpr_0601	Outgoing call / U3 MS origintg. call proceeding / PROGRESS received without in band information.	972
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_4_4	SelExpr_0601	Outgoing call / U3 MS originating call proceeding / PROGRESS with in band information	973
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_4_5	SelExpr_0601	Outgoing call / U3 MS originating call proceeding / DISCONNECT with in band tones	975
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_4_6	SelExpr_0601	Outgoing call / U3 MS originating call proceeding / DISCONNECT without in band tones	976
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_4_7	SelExpr_0601	Outgoing call / U3 MS originating call proceeding / RELEASE received	977
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_4_8	SelExpr_0601	Outgoing call / U3 MS originating call proceeding / termination requested by the user	978
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_4_9	SelExpr_0601	Outgoing call / U3 MS originating call proceeding / traffic channel allocation	979
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_4_10	SelExpr_0601	Outgoing call / U3 MS originating call proceeding / timer T310 timeout	980
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_4_11	SelExpr_0601	Outgoing call / U3 MS originating call proceeding / lower layer failure	981
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_4_12	SelExpr_0601	Outgoing call / U3 MS originating call proceeding / unknown message received	982
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_4_13	SelExpr_0605	Outgoing call / U3 MS originating call proceeding / Internal alerting indication	983
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_5_1	SelExpr_0601	Outgoing call / U4 call delivered / CONNECT received	984
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_5_2	SelExpr_0601	Outgoing call / U4 call delivered / termination requested by the user	985
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_5_3	SelExpr_0601	Outgoing call / U4 call delivered / DISCONNECT with in band tones	986
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_5_4	SelExpr_0601	Outgoing call / U4 call delivered / DISCONNECT without in band tones	987
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_5_5	SelExpr_0601	Outgoing call / U4 call delivered / RELEASE received	988
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_5_6	SelExpr_0601	Outgoing call / U4 call delivered / lower layer failure	989
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_5_7	SelExpr_0601	Outgoing call / U4 call delivered / traffic channel allocation	990
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_5_8	SelExpr_0601	Outgoing call / U4 call delivered / unknown message received	991
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_6_1	SelExpr_0601	U10 call active / termination requested by the user	992
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_6_2	SelExpr_0601	10 call active / RELEASE received	993
GSM_L3_MS_v4220/	TC_26_8_1_2_6_3	SelExpr_0601	U10 call active / DISCONNECT with in band	994

CC/			tones	
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_6_4	SelExpr_0601	U10 call active / DISCONNECT without in band tones	995
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_6_5	SelExpr_0601	U10 call active / RELEASE COMPLETE received	996
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_6_6	SelExpr_0601	U10 call active / SETUP received	997
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_7_1	SelExpr_0601	U11 disconnect request / clear collision	998
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_7_2	SelExpr_0601	U11 disconnect request / RELEASE received	999
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_7_3	SelExpr_0601	U11 disconnect request / timer T305 timeout	1000
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_7_4	SelExpr_0601	U11 disconnect request / lower layer failure	1001
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_7_5	SelExpr_0601	U11 disconnect request / unknown message received	1002
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_8_1	SelExpr_0604	U12 disconnect indication / call releasing requested by the user	1003
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_8_2	SelExpr_0604	U12 disconnect indication / RELEASE received	1004
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_8_3	SelExpr_0604	U12 disconnect indication / lower layer failure	1005
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_8_4	SelExpr_0604	U12 disconnect indication / unknown message received	1006
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_9_1	SelExpr_0601	Outgoing call / U19 release request / timer T308 timeout	1007
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_9_2	SelExpr_0601	Outgoing call / U19 release request / 2nd timer T308 timeout	1008
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_9_3	SelExpr_0601	Outgoing call / U19 release request / RELEASE received	1009
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_9_4	SelExpr_0601	Outgoing call / U19 release request / RELEASE COMPLETE received	1010
GSM_L3_MS_v4220/ CC/	TC_26_8_1_2_9_5	SelExpr_0601	Outgoing call / U19 release request / lower layer failure	1011
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_1_1	SelExpr_0600	Incoming call / U0 null state / SETUP received with a non supported bearer capability	1012
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_2_1	SelExpr_0603	Incoming call / U6 call present / automatic call rejection	1013
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_3_1	SelExpr_0606	Incoming call / U9 mobile terminating call confirmed / alerting or immediate connecting 11.10 Ref. ver. 4.10.0; CR.	1014
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_3_2	SelExpr_0602	Incoming call / U9 mobile terminating call confirmed / TCH assignment	1015
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_3_3	SelExpr_0602	Incoming call / U9 mobile terminating call confirmed / termination requested by the user	1016
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_3_4	SelExpr_0602	Incoming call / U9 mobile terminating call confirmed / DISCONNECT received	1017
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_3_5	SelExpr_0602	Incoming call / U9 mobile terminating call confirmed / RELEASE received	1018
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_3_6	SelExpr_0602	Incoming call / U9 mobile terminating call confirmed / lower layer failure	1019
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_3_7	SelExpr_0602	Incoming call / U9 mobile terminating call confirmed / unknown message received	1020
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_4_1	SelExpr_0602	Incoming call / U7 call received / call accepted	1021
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_4_2	SelExpr_0602	Incoming call / U7 call received / termination requested by the user 11.10 Ref. ver. 4.10.0; CR.	1022
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_4_3	SelExpr_0602	Incoming call / U7 call received / DISCONNECT received	1023
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_4_4	SelExpr_0602	Incoming call / U7 call received / RELEASE received	1024
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_4_5	SelExpr_0602	Incoming call / U7 call received / lower layer failure	1025
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_4_6	SelExpr_0602	Incoming call / U7 call received / unknown message received	1026
GSM_L3_MS_v4220/ CC/	TC_26_8_1_3_4_7	SelExpr_0602	Incoming call / U7 call received / TCH assignment	1027

GSM_L3_MS_v4220/CC/	TC_26_8_1_3_4_8	SelExpr_0602	Incoming call / U7 call received / RELEASE COMPLETE received	1028
GSM_L3_MS_v4220/CC/	TC_26_8_1_3_5_1	SelExpr_0602	Incoming call / U8 connect request / CONNECT acknowledged	1029
GSM_L3_MS_v4220/CC/	TC_26_8_1_3_5_2	SelExpr_0602	Incoming call / U8 connect request / timer T313 timeout	1030
GSM_L3_MS_v4220/CC/	TC_26_8_1_3_5_3	SelExpr_0602	Incoming call / U8 connect request / termination requested by the user	1031
GSM_L3_MS_v4220/CC/	TC_26_8_1_3_5_4	SelExpr_0602	Incoming call / U8 connect request / DISCONNECT received with in-band information	1032
GSM_L3_MS_v4220/CC/	TC_26_8_1_3_5_5	SelExpr_0602	Incoming call / U8 connect request / DISCONNECT received without in-band information	1033
GSM_L3_MS_v4220/CC/	TC_26_8_1_3_5_6	SelExpr_0602	Incoming call / U8 connect request / RELEASE received	1034
GSM_L3_MS_v4220/CC/	TC_26_8_1_3_5_7	SelExpr_0602	Incoming call / U8 connect request / lower layer failure	1035
GSM_L3_MS_v4220/CC/	TC_26_8_1_3_5_8	SelExpr_0602	Incoming call / U8 connect request / TCH assignment	1036
GSM_L3_MS_v4220/CC/	TC_26_8_1_3_5_9	SelExpr_0602	Incoming call / U8 connect request / unknown message received	1037
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_1_1	SelExpr_0604	In-call functions / DTMF information transfer / basic procedure	1038
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_2_1	SelExpr_0606	In-call functions / User notification / MS terminated	1040
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_3_1	SelExpr_0606	In-call functions / Channel changes / A successful channel change in active state/ Handover and assignment command	1041
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_3_2	SelExpr_0606	In-call functions / Channel changes / An unsuccessful channel change in active mode/ Handover and assignment command	1043
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_4_1	SelExpr_0607	In-call functions / MS terminated in-call modification / modify when new mode is not supported	1045
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_5_1	SelExpr_0607	In-call functions / MS originated in-call modification / A successful case of modifying	1047
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_5_2	SelExpr_0607	In-call functions / MS originated in-call modification / modify rejected	1048
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_5_3	SelExpr_0607	In-call functions / MS originated in-call modification / an abnormal case of acceptance	1049
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_5_4	SelExpr_0607	In-call functions / MS originated in-call modification / an abnormal case of rejection	1050
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_5_5	SelExpr_0607	In-call functions / MS originated in-call modification / timeout of timer T323	1051
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_5_6	SelExpr_0607	In-call functions / MS orignated in-call modification / A successful channel change in state mobile originating modify	1052
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_5_7	SelExpr_0607	In-call functions / MS originated in-call modification / An unsuccessful channel change in state mobile originating modify	1054
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_5_8	SelExpr_0607	In-call functions / MS originated in-call modification / unknown message received	1056
GSM_L3_MS_v4220/CC/	TC_26_8_1_4_5_9	SelExpr_0607	In-call functions / MS originated in-call modification / a release complete received	1057
GSM_L3_MS_v4220/CC/	TC_26_8_2_1	SelExpr_0601	Call Re-establishment / Call Present, re-establishment allowed	1058
GSM_L3_MS_v4220/CC/	TC_26_8_2_2	SelExpr_0601	Call Re-establishment / Call Present, re-establishment not allowed 11.10 Ref. ver. 4.10.0; CR.	1061
GSM_L3_MS_v4220/CC/	TC_26_8_2_3	SelExpr_0601	Call Re-establishment / Call under establishment, transmission stopped 11.10 Ref. ver. 4.10.0; CR.	1063
GSM_L3_MS_v4220/CC/	TC_26_8_3	SelExpr_0606	User to user signalling	1064
GSM_L3_MS_v4220/StructureProc/	TC_26_9_2	SelExpr_0701	Structured procedures / MS originated call / early assignment	1066
GSM_L3_MS_v4220/StructureProc/	TC_26_9_3	SelExpr_0701	Structured procedures / MS originated call / late assignment	1069
GSM_L3_MS_v4220/	TC_26_9_4	SelExpr_0701	Structured procedures / MS terminated call /	1072

StructureProc/			early assignment	
GSM_L3_MS_v4220/ StructureProc/	TC_26_9_5	SelExpr_0701	Structured procedures / MS terminated call / late assignment	1076
GSM_L3_MS_v4220/ StructureProc/	TC_26_9_6_1_1	SelExpr_0702	Structured procedures / emergency call / idle updated / preferred channel rate	1079
GSM_L3_MS_v4220/ StructureProc/	TC_26_9_6_1_2	SelExpr_0703	Structured procedures / emergency call / idle updated, non-preferred channel rate	1082
GSM_L3_MS_v4220/ StructureProc/	TC_26_9_6_2_1	SelExpr_0702	Structured procedures / emergency call / idle, no IMSI / accept case	1085
GSM_L3_MS_v4220/ StructureProc/	TC_26_9_6_2_2	SelExpr_0702	Structured procedures / emergency call / idle, no IMSI / reject case	1087
GSM_L3_MS_v4220/ EGSMSignalling/	TC_26_10_2_1	SelExpr_1001	E-GSM signalling/ RR / measurement	1088
GSM_L3_MS_v4220/ EGSMSignalling/	TC_26_10_2_2	SelExpr_1000	E-GSM signalling/ RR / immediate assignment	1092
GSM_L3_MS_v4220/ EGSMSignalling/	TC_26_10_2_3	SelExpr_1000	E-GSM signalling/ RR / channel assignment procedure	1094
GSM_L3_MS_v4220/ EGSMSignalling/	TC_26_10_2_4_1	SelExpr_1001	E-GSM signalling/ RR / Handover / successful case	1098
GSM_L3_MS_v4220/ EGSMSignalling/	TC_26_10_2_4_2	SelExpr_1001	E-GSM signalling/ RR / Handover / layer 1 failure	1102
GSM_L3_MS_v4220/ EGSMSignalling/	TC_26_10_2_5	SelExpr_1000	E-GSM signalling/ RR / frequency redifinition	1104
GSM_L3_MS_v4220/ EGSMSignalling/	TC_26_10_3_1	SelExpr_1002	E-GSM signalling/ structured procedure / mobile originating call	1107
GSM_L3_MS_v4220/ EGSMSignalling/	TC_26_10_3_2	SelExpr_1003	E-GSM signalling/ structured procedure / emergency call	1109
GSM_L3_MS_v4220/ SS/	TC_31_2_1_1_1	SelExpr_0809	Call forwarding supplementary services / Registration - Reg. accepted	1111
GSM_L3_MS_v4220/ SS/	TC_31_2_1_1_2	SelExpr_0810	Call forwarding supplementary services / Registration - Reg. rejected	1113
GSM_L3_MS_v4220/ SS/	TC_31_2_1_2_1	SelExpr_0811	Call forwarding supplementary services / Erasure by the subscriber - Erasure Accepted	1116
GSM_L3_MS_v4220/ SS/	TC_31_2_1_2_2	SelExpr_0809	Call forwarding supplementary services / Erasure by the subscriber - Erasure rejected	1118
GSM_L3_MS_v4220/ SS/	TC_31_2_1_3	SelExpr_0810	Call forwarding supplementary services / Activation	1120
GSM_L3_MS_v4220/ SS/	TC_31_2_1_4	SelExpr_0811	Call forwarding supplementary services / Deactivation	1122
GSM_L3_MS_v4220/ SS/	TC_31_2_1_6_1	SelExpr_0811	Call forwarding supplementary services / Interrogation - Interrogation Accepted	1124
GSM_L3_MS_v4220/ SS/	TC_31_2_1_6_2	SelExpr_0812	Call forwarding supplementary services / Interrogation - Interrogation Rejected	1126
GSM_L3_MS_v4220/ SS/	TC_31_2_1_7_1_1	SelExpr_0813	Normal operation - Served mobile subscriber side / Notification during an incomming call	1128
GSM_L3_MS_v4220/ SS/	TC_31_2_1_7_1_2	SelExpr_0810	Normal operation / served mobile subscriber side / Notification during outgoing call	1130
GSM_L3_MS_v4220/ SS/	TC_31_2_1_7_2	SelExpr_0810	Normal operation / Forwarded-to mobile subscriber side	1132
GSM_L3_MS_v4220/ SS/	TC_31_6_1_1	SelExpr_0802	AOC time related charging / MS originated call	1134
GSM_L3_MS_v4220/ SS/	TC_31_6_1_2	SelExpr_0802	AOC time related charging / MS terminated call	1136
GSM_L3_MS_v4220/ SS/	TC_31_6_1_5	SelExpr_0802	Change in charging information during a call	1138
GSM_L3_MS_v4220/ SS/	TC_31_6_1_6	SelExpr_0802	Different formats of charging information	1140
GSM_L3_MS_v4220/ SS/	TC_31_6_1_7	SelExpr_0819	OC on a Call Hold call	1143
GSM_L3_MS_v4220/ SS/	TC_31_6_1_8	SelExpr_0820	AOC on a MultiParty call	1145
GSM_L3_MS_v4220/ SS/	TC_31_6_2_1	SelExpr_0818	Charge Storage - Removal of SIM during an active call	1147
GSM_L3_MS_v4220/ SS/	TC_31_6_2_2	SelExpr_0821	Charge Storage - Interruption of power supply during an active call	1149
GSM_L3_MS_v4220/ SS/	TC_31_6_2_3	SelExpr_0802	Charge Storage - MS going out of coverage during an active AOCC Call	1150
GSM_L3_MS_v4220/ SS/	TC_31_6_2_4	SelExpr_0802	Charge Storage - ACMM operation / Mobile Originating	1151
GSM_L3_MS_v4220/ SS/	TC_31_6_2_5	SelExpr_0802	Charge Storage - ACMM operation / Mobile Terminating	1154

GSM_L3_MS_v4220/ SS/	TC_31_8_1_1	SelExpr_0801	Call restriction supplementary services / Registration of a password/ Registration accepted	1157
GSM_L3_MS_v4220/ SS/	TC_31_8_1_2_1	SelExpr_0801	Rejection after invoke of the RegisterPassword operation	1159
GSM_L3_MS_v4220/ SS/	TC_31_8_1_2_2	SelExpr_0801	Rejection after password check with negative result	1161
GSM_L3_MS_v4220/ SS/	TC_31_8_1_2_3	SelExpr_0801	Rejection after new password mismatch	1163
GSM_L3_MS_v4220/ SS/	TC_31_8_3_1	SelExpr_0814	Activation accepted	1165
GSM_L3_MS_v4220/ SS/	TC_31_8_3_2_1	SelExpr_0803	Rejection after invoke of ActivateSS operation	1168
GSM_L3_MS_v4220/ SS/	TC_31_8_3_2_2	SelExpr_0804	Rejection after use of password procedure	1170
GSM_L3_MS_v4220/ SS/	TC_31_8_4_1	SelExpr_0801	Deactivation accepted	1172
GSM_L3_MS_v4220/ SS/	TC_31_8_4_2_1	SelExpr_0808	Deactivation rejected / Rejection after invoke of DeactivateSS operation	1175
GSM_L3_MS_v4220/ SS/	TC_31_8_4_2_2	SelExpr_0805	Deactivation rejected / Rejection after use of password procedure	1177
GSM_L3_MS_v4220/ SS/	TC_31_8_6_1	SelExpr_0806	Interrogation / Interrogation accepted	1179
GSM_L3_MS_v4220/ SS/	TC_31_8_6_2	SelExpr_0807	Interrogation / Interrogation rejected	1182
GSM_L3_MS_v4220/ SS/	TC_31_8_7	SelExpr_0808	Normal operation	1185
GSM_L3_MS_v4220/ SS/	TC_31_9_1_1	SelExpr_0816	mobile stattion initiated / unstructured supplementary service request accepted	1187
GSM_L3_MS_v4220/ SS/	TC_31_9_1_2	SelExpr_0817	mobile station initiated / unstructured supplementary service data request rejected	1191
GSM_L3_MS_v4220/ SS/	TC_31_9_2_1	SelExpr_0817	network initaited / unstructured supplementary service request accepted	1196
GSM_L3_MS_v4220/ SS/	TC_31_9_2_2	SelExpr_0817	network initiated / unstructured supplementary service request rejected	1198
GSM_L3_MS_v4220/ SS/	TC_31_9_2_3	SelExpr_0817	network initiated / unstructured supplementary service data notification accepted	1199
GSM_L3_MS_v4220/ SS/	TC_31_9_2_4	SelExpr_0817	network initiated / unstructured supplementary service data notification rejected	1201
GSM_L3_MS_v4220/ SS/	TC_31_10	SelExpr_0815	MMI input for USSD	1202
GSM_L3_MS_v4220/ SM/	TC_34_2_1	SelExpr_0901	Short message service / SMS point to point - SMS mobile terminated	1204
GSM_L3_MS_v4220/ SM/	TC_34_2_2	SelExpr_0902	Short message service / SMS point to point - SMS mobile originated	1208
GSM_L3_MS_v4220/ SM/	TC_34_2_3	SelExpr_0903	Short message service / SMS point to point - Test of the memory available notification	1212
GSM_L3_MS_v4220/ SM/	TC_34_2_4	SelExpr_0904	Short message service / Test of the status report capabilities and of SMS-COMMAND	1217
GSM_L3_MS_v4220/ SM/	TC_34_2_5_1	SelExpr_0905	Short message service / Test of message class 0 to 3 - Test of Class 0 Short Messages	1220
GSM_L3_MS_v4220/ SM/	TC_34_2_5_2	SelExpr_0906	Short message service / Test of message class 0 to 3 - Test of Class 1 Short Messages	1222
GSM_L3_MS_v4220/ SM/	TC_34_2_5_3	SelExpr_0907	Short message service / Test of message class 0 to 3 - Test of Class 2 Short Messages	1224
GSM_L3_MS_v4220/ SM/	TC_34_2_7	SelExpr_0908	Short message service / Test of the replace mechanism for SM type 1-7	1226
GSM_L3_MS_v4220/ SM/	TC_34_2_8	SelExpr_0909	Short message service / Test of the reply path scheme	1228
GSM_L3_MS_v4220/ SM/	TC_34_3	SelExpr_0900	Short message service cell broadcast	1231
Detailed Comments:				

Test step index

Test Step Index			
Test Step Group Reference	**Test Step Id**	**Description**	**Page Nr**
GSM_L3_MS_v4220/management/	StopAllBCCH	To stop the RF transmission of all BCCH channels in all active cells.	1233
GSM_L3_MS_v4220/management/	Stopmaindcch	To stop the transmission of main dcch channel.	1233
GSM_L3_MS_v4220/management/	CCConfigTCH		1234
GSM_L3_MS_v4220/management/	CCConfigTCH_B		1234
GSM_L3_MS_v4220/management/	CCConfigTCH_nociph		1235
GSM_L3_MS_v4220/management/	CCConfigTCH_pwr		1236
GSM_L3_MS_v4220/management/	Config_CBMS_A_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH, SDCCH4 and CBCH onto the physical channel which represents cell A.	1237
GSM_L3_MS_v4220/management/	Config_FACCHF_A_1	To set one physical channel used as TCHF_ACCH's of cell A.	1237
GSM_L3_MS_v4220/management/	Config_FACCHF_A_2	To set one physical channel used as TCHF_ACCH's for instance 2 of cell A.	1237
GSM_L3_MS_v4220/management/	Config_FACCHF_B_1	To set one physical channel used as TCHF_ACCH's of cell B.	1238
GSM_L3_MS_v4220/management/	Config_FACCHF_H_1	To set one physical channel used as TCHF_ACCH's of cell H.	1238
GSM_L3_MS_v4220/management/	Config_FACCHH_A_1	To set one physical channel used as TCHH_ACCH's of cell A.	1238
GSM_L3_MS_v4220/management/	Config_FACCHH_A_2	To set one physical channel used as TCHH_ACCH's of cell A.	1239
GSM_L3_MS_v4220/management/	Config_FACCHH_B_1	To set one physical channel used as TCHH_ACCH's of cell B.	1239
GSM_L3_MS_v4220/management/	Config_BCCH_CCCH_A_1	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell A.	1239
GSM_L3_MS_v4220/management/	Config_BCCH_CCCH_A_2	To set a physical channel and map the second BCCH, CCCH onto the physical channel which represents cell A.	1240
GSM_L3_MS_v4220/management/	Config_BCCH_CCCH_A_3	To set a physical channel and map the third BCCH, CCCH onto the physical channel which represents cell A.	1240
GSM_L3_MS_v4220/management/	Config_BCCH_CCCH_A_4	To set a physical channel and map the fourth BCCH, CCCH onto the physical channel which represents cell A.	1240
GSM_L3_MS_v4220/management/	Config_BCCH_CCCH_B_1	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell B.	1241
GSM_L3_MS_v4220/management/	Config_BCCH_CCCH_B_ho_1	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell B with controlable td and fn.	1241
GSM_L3_MS_v4220/management/	Config_SDCCH4_A_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.	1241
GSM_L3_MS_v4220/management/	Config_SDCCH4_B_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.	1242
GSM_L3_MS_v4220/management/	Config_SDCCH4_B_ho_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B with controlable td and fn.	1242
GSM_L3_MS_v4220/management/	Config_SDCCH4_C_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell C	1243
GSM_L3_MS_v4220/management/	Config_SDCCH4_D_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell D	1243
GSM_L3_MS_v4220/management/	Config_SDCCH4_E_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell E	1244
GSM_L3_MS_v4220/management/	Config_SDCCH4_F_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell F	1244
GSM_L3_MS_v4220/management/	Config_SDCCH4_G_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell G	1245

GSM_L3_MS_v4220/management/	Config_SDCCH4_H_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell H	1245
GSM_L3_MS_v4220/management/	Config_SDCCH8_A_1	To set one physical channel used as SDCCH8 channel for instance 1 of cell A.	1246
GSM_L3_MS_v4220/management/	Config_SDCCH8_A_2	To set one physical channel used as SDCCH8 channel for instance 2 of cell A.	1246
GSM_L3_MS_v4220/management/	Config_SDCCH8_B_1	To set one physical channel used as SDCCH8 channel for instance 1 of cell B.	1247
GSM_L3_MS_v4220/management/	CombinedBCCH_A_CBMS	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.	1247
GSM_L3_MS_v4220/management/	CombinedBCCH_A	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.	1248
GSM_L3_MS_v4220/management/	CombinedBCCH_B	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.	1248
GSM_L3_MS_v4220/management/	CombinedBCCH_B_ho	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.	1248
GSM_L3_MS_v4220/management/	CombinedBCCH_C	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell C.	1249
GSM_L3_MS_v4220/management/	CombinedBCCH_D	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell D.	1249
GSM_L3_MS_v4220/management/	CombinedBCCH_E	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell E.	1249
GSM_L3_MS_v4220/management/	CombinedBCCH_F	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell F.	1250
GSM_L3_MS_v4220/management/	CombinedBCCH_G	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell G.	1250
GSM_L3_MS_v4220/management/	CombinedBCCH_G_sp	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell G.	1251
GSM_L3_MS_v4220/management/	CombinedBCCH_H	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell H.	1251
GSM_L3_MS_v4220/management/	NonCombinedBCCH_A	To set one physical channel used as FCHH_SCH_BCCH_CCCH for cell A.	1252
GSM_L3_MS_v4220/management/	NonCombinedBCCH_A_2	To set a physical channel and map the second BCCH, CCCH onto the physical channel which represents cell A.	1252
GSM_L3_MS_v4220/management/	NonCombinedBCCH_A_3	To set a physical channel and map the third BCCH, CCCH onto the physical channel which represents cell A.	1252
GSM_L3_MS_v4220/management/	NonCombinedBCCH_A_4	To set a physical channel and map the fourth BCCH, CCCH onto the physical channel which represents cell A.	1253
GSM_L3_MS_v4220/management/	NonCombinedBCCH_B	To set a physical channel and map FCCH, SCH, BCCH, CCCH onto the physical channel which represents cell B for RR testing.	1253
GSM_L3_MS_v4220/management/	NonCombinedBCCH_B_ho	To set a physical channel and map FCCH, SCH, BCCH, CCCH onto the physical channel which represents cell B for RR testing.	1254
GSM_L3_MS_v4220/management/	FullRateCh_A_1	To set one physical channel used as TCHF_ACCH's for instance 1 of cell A.	1254
GSM_L3_MS_v4220/management/	FullRateCh_A_1_nociph	To set one physical channel used as TCHF_ACCH's for instance 1 of cell A.	1255
GSM_L3_MS_v4220/management/	FullRateCh_A_2	To set one physical channel used as TCHF_ACCH's for instance 2 of cell A for TC_26_6_13_1.	1255
GSM_L3_MS_v4220/management/	FullRateCh_A_2_nociph	To set one physical channel used as TCHF_ACCH's for instance 2 of cell A for TC_26_6_13_1.	1255
GSM_L3_MS_v4220/management/	FullRateCh_B_1	To set one physical channel used as TCHF_ACCH's for instance 1 of cell B.	1256
GSM_L3_MS_v4220/management/	FullRateCh_B_1_nociph	To set one physical channel used as TCHF_ACCH's for instance 1 of cell B.	1256
GSM_L3_MS_v4220/management/	HalfRateCh_A_1	To set one physical channel used as TCHH_ACCH's	1256

		for instance 1 of cell A.	
GSM_L3_MS_v4220/management/	HalfRateCh_A_1_nociph	To set one physical channel used as TCHH_ACCH's for instance 1 of cell A.	1257
GSM_L3_MS_v4220/management/	HalfRateCh_A_2	To set one physical channel used as TCHH_ACCH's for instance 2 of cell A for TC_26_6_13_1.	1257
GSM_L3_MS_v4220/management/	HalfRateCh_A_2_nociph	To set one physical channel used as TCHH_ACCH's for instance 2 of cell A for TC_26_6_13_1.	1258
GSM_L3_MS_v4220/management/	HalfRateCh_B_1	To set one physical channel used as TCHH_ACCH's for instance 1 of cell B	1258
GSM_L3_MS_v4220/management/	HalfRateCh_B_1_nociph	To set one physical channel used as TCHH_ACCH's for instance 1 of cell B	1259
GSM_L3_MS_v4220/management/	SDCCH8_A_1	To set one physical channel used as SDCCH8 channel for instance 1 of cell A.	1259
GSM_L3_MS_v4220/management/	SDCCH8_A_1_nociph	To set one physical channel used as SDCCH8 channel for instance 1 of cell A.	1259
GSM_L3_MS_v4220/management/	SDCCH8_A_2_nociph	To set one physical channel used as hopping SDCCH8 channel for instance 2 of cell A, for TC_26_6_13_1.	1260
GSM_L3_MS_v4220/management/	SDCCH8_B_1	To set one physical channel used as hopping SDCCH8 channel for instance 1 of cell B .	1260
GSM_L3_MS_v4220/management/	SDCCH8_B_1_nociph	To set one physical channel used as hopping SDCCH8 channel for instance 1 of cell B For TC_26_6_13_5.	1261
GSM_L3_MS_v4220/Miscellaneous/	Adjust_gsmanddcs_powerlvl		1261
GSM_L3_MS_v4220/Miscellaneous/	AOC_CHK_FAC	Check the receive of FACILITY within one second at AOCC	1262
GSM_L3_MS_v4220/Miscellaneous/	AssCh_complete		1262
GSM_L3_MS_v4220/Miscellaneous/	AssCh_failure		1263
GSM_L3_MS_v4220/Miscellaneous/	AssCmdGen		1264
GSM_L3_MS_v4220/Miscellaneous/	AssCmdGen_fh		1265
GSM_L3_MS_v4220/Miscellaneous/	Authentication	To execute the authentication procedure.	1265
GSM_L3_MS_v4220/Miscellaneous/	CallProcGen	To generate a CALL PROCEEDING message .	1266
GSM_L3_MS_v4220/Miscellaneous/	CCAssignTCH		1267
GSM_L3_MS_v4220/Miscellaneous/	CCCH_group_Paging_group		1268
GSM_L3_MS_v4220/Miscellaneous/	CCEstablishMO_SDCCH4		1269
GSM_L3_MS_v4220/Miscellaneous/	CCEstablishMO_TCH		1269
GSM_L3_MS_v4220/Miscellaneous/	CCEstablishMT_SDCCH4		1269
GSM_L3_MS_v4220/Miscellaneous/	CCEstablishMT_TCH		1270
GSM_L3_MS_v4220/Miscellaneous/	CC_Est_MT_Call		1270
GSM_L3_MS_v4220/Miscellaneous/	CC_EstMsTermCall		1271
GSM_L3_MS_v4220/Miscellaneous/	CCModifyTCH		1271
GSM_L3_MS_v4220/Miscellaneous/	CCPage		1272
GSM_L3_MS_v4220/Miscellaneous/	CCstatuschk_01	To check whether the MS under test is in the CC state `st`.	1272
GSM_L3_MS_v4220/Miscellaneous/	CCstatuschk_02	To check whether the MS under test is in the CC state `st`.	1272
GSM_L3_MS_v4220/Miscellaneous/	CCstatuschk_03	To check whether the MS under test is in the CC state `st`.	1273
GSM_L3_MS_v4220/Miscellaneous/	CCstatuschk_05	To check whether the MS under test is in the CC state `st`.	1273
GSM_L3_MS_v4220/Miscellaneous/	Check_hoaccPwr		1274
GSM_L3_MS_v4220/Miscellaneous/	Check_hoaccTiming		1275
GSM_L3_MS_v4220/Miscellaneous/	Check_Time		1275
GSM_L3_MS_v4220/Miscellaneous/	CheckMTDif		1275
GSM_L3_MS_v4220/Miscellaneous/	CheckTIsInStateU0		1276
GSM_L3_MS_v4220/Miscellaneous/	ChkMsrmntRpt		1276
GSM_L3_MS_v4220/Miscellaneous/	Ciphering_off	To switch off the ciphering. In presteps the ciphering shall be switched on.	1277
GSM_L3_MS_v4220/Miscellaneous/	Ciphering_on	To switch on the ciphering.	1277
GSM_L3_MS_v4220/Miscellaneous/	CMsrvcRq	To request a CM service	1277
GSM_L3_MS_v4220/Miscellaneous/	DTMFSignalling	To handle DTMF signalling initiated by the MS, which sends 'n' tones to the network.	1278
GSM_L3_MS_v4220/Miscellaneous/	EstablishFacMO		1278
GSM_L3_MS_v4220/Miscellaneous/	FnArith	To calcurate the difference of two frame numbers `fn1` and `fn`. The test step uses TCV_Cnt1,	1279

		TCV_Cnt2, TCV_K and TCV_Time.	
GSM_L3_MS_v4220/Miscellaneous/	IdentityRequest	To execute the identity procedure depending on the given identity type. Used var's: TCV_ch, TCV_tmsi	1279
GSM_L3_MS_v4220/Miscellaneous/	ImsiAttach		1280
GSM_L3_MS_v4220/Miscellaneous/	ImsiAttachIni		1281
GSM_L3_MS_v4220/Miscellaneous/	ImsiAttachNoReaction		1282
GSM_L3_MS_v4220/Miscellaneous/	ImsiDetach		1283
GSM_L3_MS_v4220/Miscellaneous/	ImsiDetachNoReaction		1284
GSM_L3_MS_v4220/Miscellaneous/	InCallModi1		1284
GSM_L3_MS_v4220/Miscellaneous/	MM_LUP	To execute the Location Update Procedure. The parameter lup_mi is the actual mi of MS before MM_LUP.	1285
GSM_L3_MS_v4220/Miscellaneous/	MM_LUP2	To execute the Location Update Procedure. The parameter lup_mi is the actual mi of MS before MM_LUP.	1285
GSM_L3_MS_v4220/Miscellaneous/	MM_LUP3	To execute the normal Location Update Procedure. No check of parameter.	1285
GSM_L3_MS_v4220/Miscellaneous/	MM_LupAndStop	To execute the initialisation of the Location Update Procedure.	1286
GSM_L3_MS_v4220/Miscellaneous/	MM_LupAuthRpt	To initialise Location Update and repeated Authentication procedures till T3210 times out.	1286
GSM_L3_MS_v4220/Miscellaneous/	MM_LUPauth1	To execute the Location Update Procedure with authentication. Check of the parameter LAI, MSClassMark and Mobile Identity is not required and are not checked.	1287
GSM_L3_MS_v4220/Miscellaneous/	MM_LUPauth2	To execute the Location Update Procedure with authentication. Checking of the parameter LAI, MSClassMark and Mobile Identity is required.	1287
GSM_L3_MS_v4220/Miscellaneous/	MM_LUP_imsi	To execute the Location Update Procedure only for IMSI. The parameter lup_mi is the actual mi of MS before MM_LUP. It is called by TC_26_7_4_3_1.	1288
GSM_L3_MS_v4220/Miscellaneous/	MM_LupInit	To execute the initialisation of the Location Update Procedure.	1288
GSM_L3_MS_v4220/Miscellaneous/	MM_LupInit2	To execute the initialisation of theLocation Update Procedure.	1289
GSM_L3_MS_v4220/Miscellaneous/	MM_LupInit3	To execute the initialisation of the Location Update Procedure.	1289
GSM_L3_MS_v4220/Miscellaneous/	MM_LupInit4	To execute the initialisation of theLocation Update Procedure.	1290
GSM_L3_MS_v4220/Miscellaneous/	MM_LupInit5	To execute the initialisation of the Location Update Procedure.	1290
GSM_L3_MS_v4220/Miscellaneous/	MM_LupInit6	To execute the initialisation of the Location Update Procedure.	1291
GSM_L3_MS_v4220/Miscellaneous/	MM_LUP_tmsirealloc	To execute the Location Update Procedure. The parameter expectedlup_mi is the actual mi of MS before MM_LUP. The expectedlup_mi is not used in this teststep, it is not required to check it. The test step is called in TC_26_7_1.	1291
GSM_L3_MS_v4220/Miscellaneous/	MM_LUPper	To execute the periodic Location Update Procedure.	1291
GSM_L3_MS_v4220/Miscellaneous/	MM_LUPper2	To execute the periodic Location Update Procedure.	1292
GSM_L3_MS_v4220/Miscellaneous/	MM_LUPperauth	To execute the periodic Location Update Procedure.	1292
GSM_L3_MS_v4220/Miscellaneous/	MM_LUPperrej	To execute the periodic Location Update Procedure.	1292
GSM_L3_MS_v4220/Miscellaneous/	MM_LUPperrej2	To execute the Location Update Procedure, which shall be rejected.	1293
GSM_L3_MS_v4220/Miscellaneous/	MM_LUPperrej3	To execute the Location Updating Procedure, which shall be stopped with channel release.	1293
GSM_L3_MS_v4220/Miscellaneous/	MM_LupRej	To execute the Location Update Procedure, which shall be rejected.	1293
GSM_L3_MS_v4220/Miscellaneous/	MM_LupRej2	To execute the Location Update Procedure, which shall be rejected.	1294
GSM_L3_MS_v4220/Miscellaneous/	MM_PwrOrSimOff		1294
GSM_L3_MS_v4220/Miscellaneous/	MM_PwrOrSimOn		1294
GSM_L3_MS_v4220/Miscellaneous/	MM_check_ecall1	To check, if MS execute the emergency call.	1295
GSM_L3_MS_v4220/Miscellaneous/	MM_check_ecall2	To check, if MS execute the emergency call with IMSI	1296
GSM_L3_MS_v4220/Miscellaneous/	MM_no_cmservices	To check, if MS doesn't execute a MO CM connection procedure.	1296

GSM_L3_MS_v4220/Miscellaneous/	MM_no_paging	To check, if MS doesn't initiate a RR connection. The duration of the checking is set in given par.	1297
GSM_L3_MS_v4220/Miscellaneous/	NoReaction	To check whether MS doesn't initiate any RR connections.	1297
GSM_L3_MS_v4220/Miscellaneous/	RR_hocomp1	To finish the HO-procedure in non-synchronized HO cases.	1298
GSM_L3_MS_v4220/Miscellaneous/	RR_hocomp3	To finish the HO-procedure in synchronized HO cases.	1299
GSM_L3_MS_v4220/Miscellaneous/	RRmtcallprepare	To establish a full rate call with non hopping in cell A for GSM900 and DCS1800. IUT should be in idle updated state.	1299
GSM_L3_MS_v4220/Miscellaneous/	RcvSetupOrEsetup		1300
GSM_L3_MS_v4220/Miscellaneous/	SendSeqNo_chk	check sending sequence number.	1300
GSM_L3_MS_v4220/Miscellaneous/	SelectPagingCh		1301
GSM_L3_MS_v4220/Miscellaneous/	Set_CellChDescr	To set the cell channel description	1302
GSM_L3_MS_v4220/Miscellaneous/	SetupRcvMo2		1302
GSM_L3_MS_v4220/Miscellaneous/	Switchcell		1302
GSM_L3_MS_v4220/Miscellaneous/	SvcSupportedChk	To check wether the Basic Service svc is supported against PICS answers.	1303
GSM_L3_MS_v4220/Miscellaneous/	Timadv_Pwrlvl_chk	To check the power level and the timing advance in L1 head.	1304
GSM_L3_MS_v4220/Miscellaneous/	TmsiReallocation	To execute the TMSI reallocation procedure. Used var's: TCV_ch	1304
GSM_L3_MS_v4220/Miscellaneous/	Varinit_tch		1304
GSM_L3_MS_v4220/Miscellaneous/	Varinit_fix		1305
GSM_L3_MS_v4220/Miscellaneous/	Wait		1305
GSM_L3_MS_v4220/Miscellaneous/	WaitForInService	To wait until the MS enters the Idle and updated state.	1305
GSM_L3_MS_v4220/Miscellaneous/	ChgLAC_A	To change the LAC of cell A.	1306
GSM_L3_MS_v4220/Miscellaneous/	ChgLAC_B	To change the LAC of cell B.	1306
GSM_L3_MS_v4220/Miscellaneous/	ChgLAI_C	To change the LAI of cell C to HPLMN.	1307
GSM_L3_MS_v4220/Miscellaneous/	SysInfoSending_cbms	To send system information messages for the CBMS tests.	1308
GSM_L3_MS_v4220/Miscellaneous/	SysInfoSending_fh	To send system information messages for the L3 tests. The following parameters specified by input parameters.	1310
GSM_L3_MS_v4220/Miscellaneous/	SysInfoSending_nfh	To send system information messages with default parameters defined for L3 tests for which no other special parameters indicated.	1312
GSM_L3_MS_v4220/Miscellaneous/	SysInfoSending_nfhCb	To send system information messages with default parameters defined for L3 tests except that the cell is barred.	1314
GSM_L3_MS_v4220/Miscellaneous/	SysInfoSending_e	Sending of Systeminformation for cell A in EGSM test case TC_26_10_2_2.	1315
GSM_L3_MS_v4220/Miscellaneous/	SysInfo_SacchSending	To send SYSTEM INFORMATION 5 and 6 messages defined by parameters 'sysinfo5_pdu' and 'sysinfo6_pdu' in the parametrised 'ch' channel.	1316
GSM_L3_MS_v4220/Miscellaneous/	SysInfo_5bisSending	To send SYSTEM INFORMATION 5bis message defined by parameters 'sysinfo5bis_pdu' in the parametrised 'ch' channel.	1316
GSM_L3_MS_v4220/Miscellaneous/	SetNECI	To set the NECI =1.	1317
GSM_L3_MS_v4220/Miscellaneous/	SetATT	To set the ATT flag to "MS's in the cell should apply IMSI attach and detach procedure"	1318
GSM_L3_MS_v4220/OperatorOP/	InitCall	To initiate a call of the basic service `srv`.	1319
GSM_L3_MS_v4220/OperatorOP/	InitCM		1319
GSM_L3_MS_v4220/OperatorOP/	InitNonCallSupp	To attempt a non call related supplementary service at the MS under test.	1319
GSM_L3_MS_v4220/OperatorOP/	AtmpShortMsg	To attempt an MO short message service transaction at the MS under test.	1319
GSM_L3_MS_v4220/OperatorOP/	CheckUssdStringDisplayed	To check whether the correct USSD String is displayed on the MS	1320
GSM_L3_MS_v4220/OperatorOP/	PLMNsCHK	To check whether the MS presents a list of available PLMNs.	1320
GSM_L3_MS_v4220/OperatorOP/	RFtransCHK	To check whether the MS transmits any radio signal.	1320
GSM_L3_MS_v4220/Postambles/	PostLinkRelEnd	To release the RR connection and bring the MS back to Idle state. This teststep decides the final verdict and therefor it shall be used only at the end of testcases.	1320
GSM_L3_MS_v4220/Postambles/	PostMainLinkRel	To release the main signalling link `ch`, and bring the	1321

		MS back to Idle state.	
GSM_L3_MS_v4220/Postambles/	ChanRel	To release the RR connection on the channel TCV_ch and bring the MS back to Idle state.	1321
GSM_L3_MS_v4220/Postambles/	ChanRel_P	To release the RR connection on the channel TCV_chmaindcch and bring the MS back to Idle state. This teststep decides the verdict (P).	1321
GSM_L3_MS_v4220/Postambles/	ChanRel_end	To release the RR connection and bring the MS back to Idle state. This teststep decides the final verdict and therefor it shall be used only at the end of testcases.	1321
GSM_L3_MS_v4220/Postambles/	RmvForbiddenPlmn	To remove a forbidden PLMN (C_PLMN_2) of cell B in the list on SIM.	1322
GSM_L3_MS_v4220/Postambles/	RmvForbiddenPlmn1		1322
GSM_L3_MS_v4220/Preambles/	BasicServiceMO	1. This test step generates an appropriate MO SETUP message according to the IXIT parameters of Bearer services/Teleservices for an MO call. 2. The detailed algorithms for Bcap derivation for each Bearer service/Teleservice are described in test steps BS2xMO, BS3xMO, BS4xMO, BS5xMO, BS61or81MO, TS61MO and TS62MO.	1323
GSM_L3_MS_v4220/Preambles/	BasicServiceMOorTelephony	To get a MO SETUP message with right BC IE.	1325
GSM_L3_MS_v4220/Preambles/	BasicServiceMT	1. This test step generates an appropriate MT SETUP message according to the IXIT parameters of Bearer services/Teleservices for an MT call. If possible the test step generates a 2nd MT SETUP message for TC_11_1_1. 2. The Bearer Capability in the 2nd SETUP message is chosen in such a way that as many parameters as possible are different from the Bcap in the first MT SETUP message. The detailed algorithms for Bcap derivation for each Bearer service/Teleservice are described in test steps BS2xMT, BS3xMT, BS61or81MT, TS61MT and TS62MT.	1326
GSM_L3_MS_v4220/Preambles/	BasicServiceMTorTelephony	To get a MT SETUP message with right BC IE.	1327
GSM_L3_MS_v4220/Preambles/	BS2xMO	MO SETUP message with right Bcap for BS2x service.	1328
GSM_L3_MS_v4220/Preambles/	BS2xMT	MT SETUP message with right Bcap for BS2x service.	1331
GSM_L3_MS_v4220/Preambles/	BS3xMO	MO SETUP message with right Bcap for BS3x service.	1334
GSM_L3_MS_v4220/Preambles/	BS3xMT	MT SETUP message with right Bcap for BS3x service.	1338
GSM_L3_MS_v4220/Preambles/	BS4xMO	MO SETUP message with right Bcap for BS4x service.	1343
GSM_L3_MS_v4220/Preambles/	BS5xMO	MO SETUP message with right Bcap for BS5x service.	1345
GSM_L3_MS_v4220/Preambles/	BS61or81MO	To get a MO SETUP message with right BC IE for BS61 or BS81 service.	1346
GSM_L3_MS_v4220/Preambles/	BS61or81MT	To get a MT SETUP message with right BC IE for BS61 or BS81 service.	1350
GSM_L3_MS_v4220/Preambles/	CallCfmGen	generate a CALL COMFIRMED for CALL COMFIRMED message verifying	1355
GSM_L3_MS_v4220/Preambles/	Chmod	To assign correct values to the test case variables TCV_ChMod and TCV_ChModb.	1361
GSM_L3_MS_v4220/Preambles/	Est_MO_Call	To establish a MO call in any cell.	1362
GSM_L3_MS_v4220/Preambles/	Est_MO_Call_init	To initiate a mobile originating call for the supported bearer capability. Only in HO cases.	1363
GSM_L3_MS_v4220/Preambles/	Est_MT_Call_FH	To establish a call with FH (in CELL A or B).	1365
GSM_L3_MS_v4220/Preambles/	Est_MT_CallNonFH	To establish a call with non hopping.	1365
GSM_L3_MS_v4220/Preambles/	IdleUpdated	This test step is called at the preamble of each test case.	1366
GSM_L3_MS_v4220/Preambles/	PreEnterIdleState_r02	To broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with some options in cell A, and wait for the SUT (MS) entering the Idle updated state. 2 parameters: max retransmission, and tx integer can be set/assigned and legal combination of CCCH-CONF, BS-AG-BLKS-RES, BS-PA-MFRMS are specified by parameter `ccd`.	1374
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU01_21	To bring the MS into CC state U0.1 by procedure in table 26.8.1.2/1. This is used in CC testing.	1375

GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU1_21	To bring the MS into CC state U1 by procedure in table 26.8.1.2/1. This is used in CC testing.	1376
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU1_22	To bring the MS into CC state U1 by procedure in table 26.8.1.2/2. This is used in CC testing.	1376
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU1_22Timer	To bring the MS into CC state U1 by procedure in table 26.8.1.2/2. This is used in CC testing.	1377
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU1_24	To bring the MS into CC state U1 by procedure in table 26.8.1.2/4. This is used in CC testing.	1377
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU1	To establish a mobile originating call and put the MS under test in the CC state U1.	1378
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU3	To establish a mobile originating call and put the MS under test in the CC state U3.	1378
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU3_21	To bring the MS into CC state U3 by procedure in table 26.8.1.2/1. This is used in CC testing.	1379
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU3_22	To bring the MS into CC state U3 by procedure in table 26.8.1.2/2. This is used in CC testing.	1379
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU3_23	To bring the MS into CC state U3 by procedure in table 26.8.1.2/3. This is used in CC testing.	1379
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU3_24	To bring the MS into CC state U3 by procedure in table 26.8.1.2/4. This is used in CC testing.	1380
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU4_21	To bring the MS into CC state U4 by procedure in table 26.8.1.2/1. This is used in CC testing.	1380
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU4_22	To bring the MS into CC state U4 by procedure in table 26.8.1.2/2. This is used in CC testing.	1380
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU4_23	To bring the MS into CC state U4 by procedure in table 26.8.1.2/3. This is used in CC testing.	1381
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU4_24	To bring the MS into CC state U4 by procedure in table 26.8.1.2/4. This is used in CC testing.	1381
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU6_32	To bring the MS into CC state U6 by procedure in table 26.8.1.3/2. This is used in CC testing.	1381
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU7_31	To bring the MS into CC state U7 by procedure in table 26.8.1.3/1. This is used in CC testing.	1382
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU7_32	To bring the MS into CC state U7 by procedure in table 26.8.1.3/2. This is used in CC testing.	1382
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU7_33	To bring the MS into CC state U7 by procedure in table 26.8.1.3/3. This is used in CC testing.	1382
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU8_31	To bring the MS into CC state U8 by procedure in table 26.8.1.3/1. This is used in CC testing.	1383
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU8_32	To bring the MS into CC state U8 by procedure in table 26.8.1.3/2. This is used in CC testing.	1383
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU8_33	To bring the MS into CC state U8 by procedure in table 26.8.1.3/3. This is used in CC testing.	1384
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU8_34_SS	Entering U8, non hopping, no ciphering.	1384
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU9_31		1385
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU9_32	To bring the MS into CC state U9 by procedure in table 26.8.1.3/2. This is used in CC testing.	1385
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU9_33	To bring the MS into CC state U9 by procedure in table 26.8.1.3/3. This is used in CC testing.	1385
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU9_34	To bring the MS into CC state U9 by procedure in table 26.8.1.3/4. This is used in CC testing.	1386
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU10	To establish a mobile terminating call for the supported bearer capability and put the MS under test in the CC state U10. The supported bearer capability is specified in the input partameter setup. With early assigment, SETUP message does not contain SIGNAL information element.	1387
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU10_late	To establish a mobile terminating call for the supported bearer capability and put the MS under test in the CC state U10. The supported bearer capability is specified in the input partameter setup. Late assigment only. The SETUP message shall contain a SIGNAL information element.	1388
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU10_r01	To establish a mobile terminating call for the supported bearer capability and put the MS in the CC state U10 with late assignment.	1389
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU10_21	To bring the MS into CC state U10 by procedure in table 26.8.1.2/1. This is used in CC testing.	1390
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU10_22	To bring the MS into CC state U10 by procedure in table 26.8.1.2/2. This is used in CC testing.	1390

GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU11_23	To bring the MS into CC state U11 by procedure in table 26.8.1.2/3. This is used in CC testing.	1391
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU11_23Timer	To bring the MS into CC state U11 by procedure in table 26.8.1.2/3. This is used in CC testing.	1391
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU11_24	To bring the MS into CC state U11 by procedure in table 26.8.1.2/4. This is used in CC testing.	1392
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU12_21	To bring the MS into CC state U12 by procedure in table 26.8.1.2/1. This is used in CC testing.	1392
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU12_22	To bring the MS into CC state U12 by procedure in table 26.8.1.2/2. This is used in CC testing.	1392
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU12_23	To bring the MS into CC state U12 by procedure in table 26.8.1.2/3. This is used in CC testing.	1393
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU19_21	To bring the MS into CC state U19 by procedure in table 26.8.1.2/1. This is used in CC testing.	1393
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU19_24	To bring the MS into CC state U19 by procedure in table 26.8.1.2/4. This is used in CC testing.	1394
GSM_L3_MS_v4220/Preambles/	PreEnterCCstateU19_24Timer	To bring the MS into CC state U19 by procedure in table 26.8.1.2/4. This is used in CC testing.	1394
GSM_L3_MS_v4220/Preambles/	PreEstRRConn	To establish a RR connection on TSPX_SDCCH4SubDef	1395
GSM_L3_MS_v4220/Preambles/	PreEstRRC_MM	To establish a RR connection for MM testcases.	1395
GSM_L3_MS_v4220/Preambles/	PreModifySetup	To setup dual mode call and initiate MO incall modification.	1396
GSM_L3_MS_v4220/Preambles/	SpeechService	To selecte a speech service.	1396
GSM_L3_MS_v4220/Preambles/	StartCellA	To broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with default parameters in cell A, frequency hopping.	1397
GSM_L3_MS_v4220/Preambles/	StartCellA_1	To start cell A without hopping.	1397
GSM_L3_MS_v4220/Preambles/	StartCellB	To broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with default parameters in cell B.	1398
GSM_L3_MS_v4220/Preambles/	StartCellB_1	To start cell B with default parameters.	1399
GSM_L3_MS_v4220/Preambles/	StartCellB_egsm	To broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with default parameters in cell B.	1400
GSM_L3_MS_v4220/Preambles/	StartCellB_ho	To broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with default parameters in cell B. Cell B has a frame number difference 'fn' and the BCCH of cell B has a delay of 'td' bits to the BCCH of cell A.	1401
GSM_L3_MS_v4220/Preambles/	StartCellC	To broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with default parameters in cell C.	1402
GSM_L3_MS_v4220/Preambles/	Start_2cells		1402
GSM_L3_MS_v4220/Preambles/	StartMultiCells_01	To broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 in multiple cells for idle mode testing.	1403
GSM_L3_MS_v4220/Preambles/	StartMultiCells_02	To broadcast SYSTEM INFORMATION messages 1, 2, 3, 4, 5 and 6 in multiple cells for measurement testing. Neighbour cells description for cell S1 is a formal parameter.	1406
GSM_L3_MS_v4220/Preambles/	StartMultiCells_02e	To broadcast SYSTEM INFORMATION messages 1, 2, 3, 4, 5 and 6 in multiple cells for measurement testing. Neighbour cells description for cell S1 is a formal parameter.	1408
GSM_L3_MS_v4220/Preambles/	StartMultiCells_03	To broadcast SYSTEM INFORMATION messages 1, 2, 3, 4, 5 and 6 in multiple cells for measurement testing. Neighbour cells description for cell S1 is a formal parameter.	1410
GSM_L3_MS_v4220/Preambles/	TS61MO	MO SETUP message with right Bcap for TS61 service.	1412
GSM_L3_MS_v4220/Preambles/	TS62MO	MO SETUP message with right Bcap for TS62 service.	1414
GSM_L3_MS_v4220/Preambles/	TS61MT	MT SETUP message with right Bcap for TS61 service.	1416
GSM_L3_MS_v4220/Preambles/	TS62MT	MT SETUP message with right Bcap for TS62 service.	1418

Detailed Comments:

Default index

Default Index			
Default Group Reference	**Default Id**	**Description**	**Page Nr**
GSM_L3_MS_v4220/	DTMFsig		1420
GSM_L3_MS_v4220/	OtherEvents	To match unexpected events and sign final verdict for preambles.	1421
GSM_L3_MS_v4220/	OtherEventsFail	To match unexpected events and fail the test case.	1422
GSM_L3_MS_v4220/	OtherEventsFail_01	To match irrelevant CHANNEL REQUEST msg and return or match other unexpected events and fail the test case.	1423
GSM_L3_MS_v4220/	OtherEventsFail_02	To match unexpected events and fail the test case but ignore channel request messages that are sent before the lower tester has sent (and the MS received) the Immediate Assigment message.	1424
GSM_L3_MS_v4220/	OtherEvents_01	To match irrelevant messages and return	1425
GSM_L3_MS_v4220/	OtherEvents_02	To match unexpected events and sign final verdict for preambles but ignore channel request messages that are sent before the lower tester has sent (and the MS received) the Immediate Assigment message.	1426
GSM_L3_MS_v4220/	RcvHdOvAcc	To match any HANDOVER ACCESS then return to calling tree.	1427
Detailed Comments:			

Declarations Part

Test suite type definitions

Simple type definitions

Simple Type Definitions		
Type Name	**Type Definition**	**Comments**
B_1	BITSTRING [1]	Generic type for 1 bit value
B_2	BITSTRING [2]	Generic type for 2 bits value
B_3	BITSTRING [3]	Generic type for 3 bits value
B_4	BITSTRING [4]	Generic type for 4 bits value
B_5	BITSTRING [5]	Generic type for 5 bits value
B_6	BITSTRING [6]	Generic type for 6 bits value
B_8	BITSTRING [8]	Generic type for 8 bits value
BCCHFRQ	BITSTRING [5]	position of a bcch carrier in the bcch channel list
BCC	BITSTRING [3]	BS colour code
BCDN	OCTETSTRING [1..10]	BCD numbers, GSM 04.08, 10.5.4.7, octet 4-13
BSIC	BITSTRING [6]	base station identity code
CCCH_CON	BITSTRING[3]	number of physical channels for ccch GSM 04.08, 10.5.2.11
CCSTATE	INTEGER(1,3,4,6,7,8,9,10,11,12,19,26,27)	
CellID	IA5String	Cell identifier
CHANNEL	BITSTRING [2]	needed channel type
CHMOD_VAL	BITSTRING[8]	value for channel mode GSM 04.08, 10.5.2.6, 10.5.2.7
CH_TDMA	BITSTRING[5]	channel type and TDMA offset GSM 04.08, 10.5.2.5
CI	OCTETSTRING [2]	cell identity, GSM 04.08, 10.5.1.1
CKSN	BITSTRING[3]	ciphering key sequence number(only key sequence) GSM 04.08, 10.5.1.2
CLRSUP	BITSTRING('10100001'B)	CLIR suppression GSM 04.08, 10.5.4.11a
CLRINV	BITSTRING('10100010'B)	CLIR invocation GSM 04.08, 10.5.4.11b
CMSVTYPE	BITSTRING [4]	CM service type
CP_CAU	OCTETSTRING [1]	cp cause GSM 04.11, 8.1.4.2
CPHALG	BITSTRING[3]	Ciphering algorithm identifier GSM 04.08, 10.5.2.9
CS	BITSTRING [2]	coding standard
EstMode	OCTETSTRING [1]	establish mode
EXTB	BITSTRING [1]	extension bit
FCS	HEXSTRING [2]	TP failure cause
FLWCNTL	INTEGER(0, 1, 2)	type of flow control. 0--outband flow control; 1--inband flow control; 2-- no flow control
HORF	BITSTRING [8]	handover reference
HSN	BITSTRING[6]	Hopping sequence number e.g.GSM 04.08 10.5.2.5
IDTYPE	B_4	identity type
IEI_4	BITSTRING [4]	information element identifier, type 1
IEI_8	BITSTRING [8]	information element identifier, type 2-4
L2FMTYPE	INTEGER(1..3)	L2 frame type
LENGTH	OCTETSTRING [1]	IE length or L2 pseudo length. The L2 pseudo length is composed of a L2 pseudo length value, which is 6 bits long, and of 2 additional bits. See GSM 04.08 section 10.5.2.19
LOGICCH	IA5String	

LOGCH	IA5String	
MAC	OCTETSTRING[3]	mac, e.g. GSM 04.08 10.5.2.21
MAIO	BITSTRING[6]	MAIO,GSM 04.08 10.5.2.5
MAXTXPOW	INTEGER(0..31)	Max Tx Power Level GSM 04.08, 10.5.2.4
MR	OCTETSTRING [1]	SMS RP or TP message reference
MT	BITSTRING[8]	message type, structure of it see MT0
MTI	BITSTRING [3]	SMS RP message type indicator
NCC	BITSTRING [3]	PLMN colour code
NCCP	OCTETSTRING [1]	ncc permitted
NTI	BITSTRING('10000000'B, '10000001'B, '10000010'B)	notification indicator GSM 04.08, 10.5.4.20
PD	B_4	Protocol discriminator
PGG	BITSTRING [8]	the paging group of an MS to be paged in binary presentation
RATE	IA5String("F", "H")	
RAND	BITSTRING [128]	random number
RCSD	BITSTRING [8]	reverse call setup direction GSM 04.08, 10.5.4.22a
RelMode	OCTETSTRING [1]	release mode
REJCAU	OCTETSTRING [1]	reject cause GSM 04.08, 10.5.3.6
RPI	BITSTRING [8]	repeat indicator
RRCAU	BITSTRING [8]	RR cause GSM 04.08, 10.5.2.31
RXLEV	BITSTRING [6]	received signal strength
SAPID	OCTETSTRING [1]	sap identifier
SENDINGMODE	INTEGER(1..4)	the mode of sending two consecutive messages containing paging mode IE
SERVICES	IA5String("C_Telephony", "C_EmgCall", "C_AltSpchFax", "C_AutoFax", "C_Async300", "C_Async1200", "C_Async120075", "C_Async2400", "C_Async4800", "C_Async9600", "C_Sync1200", "C_Sync2400", "C_Sync4800", "C_Sync9600", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Packet2400", "C_Packet4800", "C_Packet9600", "C_AltSpchData", "C_SpchData")	
SHOCT	B_4	spare half octet for type 1 information element GSM 04.08, 10.5.1.8
SKI	B_4	Skip indicator
SN	BITSTRING [3]	all possible slot number, GSM 04.08 10.5.2.5
SPB	B_1	spare bit
SP2B	B_2	2 spare bits
SP3B	B_3	3 spare bits
SP5B	B_5	5 spare bits
SP6B	B_6	6 spare bits
SRES	OCTETSTRING [4]	authentication response signature
TPCD	OCTETSTRING [1..157]	TP command data
TPSCTS	HEXSTRING [14]	TP service centre time stamp
TPUD	OCTETSTRING [1..140]	TP user data
TSC	B_3	Training sequence code,GSM 04.08 10.5.2.5
T1	BITSTRING [11]	
T1_	B_5	T1'
T2	B_5	
T3	BITSTRING [6]	
T3_	B_3	T3'
TI_V	B_3	TI value
TMSI_V	OCTETSTRING[4]	TMSI value is octetstring of length 4
TZONES	INTEGER(-79..79)	time zones in 15 minute steps
WI	HEXSTRING [2]	wait indication, unit in second
Detailed Comments:		

Structured Type definitions

Structured Type Definition		
Type Name:	ACST	
Comments:	Auxiliary states (CC information element) GSM 04.08, 10.5.4.4	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb	EXTB	
sp3b	SP3B	
has	BITSTRING [2]	
mpas	BITSTRING [2]	
Detailed Comments:		

Structured Type Definition		
Type Name:	EFR_BCAP_3A_ETC	
Comments:	Speech Versions preferences as supported by the EFR mobile - see GSM 04.08/10.5.4.5/octet 3a etc/"speech version indication"	
Element Name	**Type Definition**	**Comments**
extb	EXTB	
cod	BITSTRING[1]	
spr2	BITSTRING[2]	
svi	BITSTRING [4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	BCAP	
Comments:	Bearer capability (CC information element) GSM 04.08, 10.5.4.5	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
rchr	BITSTRING [2]	
cs	BITSTRING [1]	
tm	BITSTRING [1]	
itc	BITSTRING [3]	
oct_3a_etc_1	EFR_BCAP_3A_ETC	
oct_3a_etc_2	EFR_BCAP_3A_ETC	
oct_3a_etc_3	EFR_BCAP_3A_ETC	
oct_3a_etc_4	EFR_BCAP_3A_ETC	
oct_3a_etc_5	EFR_BCAP_3A_ETC	
oct_3a_etc_6	EFR_BCAP_3A_ETC	
extb4	EXTB	
spb	SPB	
strc	BITSTRING [2]	
dplxm	BITSTRING [1]	
config	BITSTRING [1]	
nirr	BITSTRING [1]	
est	BITSTRING [1]	
extb5	EXTB	
accid	BITSTRING [2]	
ra	BITSTRING [2]	
sacp	BITSTRING [3]	
extb6	EXTB	
l1id	BITSTRING [2]	
uil1	BITSTRING [4]	
sb	BITSTRING [1]	
extb6a	EXTB	
nsb	BITSTRING [1]	
nb	BITSTRING [1]	
ndb	BITSTRING [1]	
ur	BITSTRING [4]	
extb6b	EXTB	
ir	BITSTRING [2]	
nictx	BITSTRING [1]	
nicrx	BITSTRING [1]	
pi	BITSTRING [3]	
extb6c	EXTB	
ce	BITSTRING [2]	
modemt	BITSTRING [5]	
extb7	EXTB	
l2id	BITSTRING [2]	
uil2	BITSTRING [5]	
Detailed Comments:		

Structured Type Definition		
Type Name:	BLOCKTYPE	
Comments:	Block type, GSM 04.12, 3.3.1	
Element Name	**Type Definition**	**Comments**
spare1	SPB	
lpd	BITSTRING[2]	
lb	BITSTRING[1]	
sequence_number	BITSTRING[4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CAU	
Comments:	Cause (CC information element) GSM 04.08, 10.5.4.11	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
cs	CS	
spb	SPB	
location	BITSTRING [4]	
extb3a	EXTB	
rec	BITSTRING [7]	
extb4	EXTB	
cau_class	BITSTRING [3]	
cau_va	BITSTRING [4]	
cau_di	OCTETSTRING [1..28]	
Detailed Comments:	&COMMON_U06	

Structured Type Definition		
Type Name:	CCCAP	
Comments:	Call Control Capabilities GSM 04.08, 10.5.4.5a	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
spr	BITSTRING [7]	
dtmf	BITSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CCD	
Comments:	Control channel description GSM 04.08, 10.5.2.11	
Element Name	**Type Definition**	**Comments**
spr1	BITSTRING [1]	
att	BITSTRING [1]	
babr	BITSTRING [3]	
ccch_con	BITSTRING [3]	
spr2	BITSTRING [5]	
bpm	BITSTRING [3]	
t3212	OCTETSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CCHD	
Comments:	Cell channel description GSM 04.08, 10.5.2.1	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
rfl	OCTETSTRING [16]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CD	
Comments:	Cell description GSM 04.08, 10.5.2.2	
Element Name	**Type Definition**	**Comments**
bcch_arfcn_h	BITSTRING [2]	
ncc	NCC	
bcc	BCC	
bcch_arfcn_l	BITSTRING [8]	
Detailed Comments:	The info element is two octets long.	

Structured Type Definition		
Type Name:	CDPN	
Comments:	Called party BCD number (CC information element) GSM 04.08, 10.5.4.7	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
tonnpi	TON_NPI	
digits	BCDN	
Detailed Comments:		

Structured Type Definition		
Type Name:	CDPS	
Comments:	Called party subaddress (CC information element) GSM 04.08, 10.5.4.8	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
subad	SUBAD	
Detailed Comments:		

Structured Type Definition		
Type Name:	CGPN	
Comments:	Calling party BCD number (CC information element) GSM 04.08, 10.5.4.9	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
tonnpi	TON_NPI	
pisi	PI_SI	
digits	BCDN	
Detailed Comments:		

Structured Type Definition		
Type Name:	CGPS	
Comments:	Calling party subaddress (CC information element) GSM 04.08, 10.5.4.10	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
subad	SUBAD	
Detailed Comments:		

Structured Type Definition		
Type Name:	CHD	
Comments:	Channel description GSM 04.08, 10.5.2.5	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
cht_schn	BITSTRING [5]	
tn	BITSTRING [3]	
tsc	BITSTRING [3]	
hch	BITSTRING [1]	
maio	BITSTRING [6]	
hsn	BITSTRING [6]	
spr	BITSTRING [2]	
arfcn	BITSTRING [10]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CHMOD	
Comments:	Channel mode, channel mode2 GSM 04.08, 10.5.2.6, 10.5.2.7	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
mode	B_8	
Detailed Comments:		

Structured Type Definition		
Type Name:	CHNEED	
Comments:	Channels needed GSM 04.08, 10.5.2.8	
Element Name	**Type Definition**	**Comments**
ch2	CHANNEL	
ch1	CHANNEL	
Detailed Comments:		

Structured Type Definition		
Type Name:	CNN	
Comments:	Connected number (CC information element) GSM 04.08, 10.5.4.13	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
tonnpi	TON_NPI	
pisi	PI_SI	
digits	BCDN	
Detailed Comments:		

Structured Type Definition		
Type Name:	CNS	
Comments:	Connected subaddress (CC information element) GSM 04.08, 10.5.4.14	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
subad	SUBAD	
Detailed Comments:		

Structured Type Definition		
Type Name:	CO	
Comments:	Cell options (BCCH / SACCH) GSM 04.08, 10.5.2.3	
Element Name	**Type Definition**	**Comments**
sprb	BITSTRING [1]	
pwrc	BITSTRING [1]	
dtx	BITSTRING [2]	
rlt	BITSTRING [4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	Component_T	
Comments:	Component for sending (downlink) GSM 04.80, 3.6	
Element Name	**Type Definition**	**Comments**
comp_part1	OCTETSTRING	
invokeId	OCTETSTRING [1]	
comp_part2	OCTETSTRING	
comp_part3	OCTETSTRING	
comp_part4	OCTETSTRING	
comp_part5	OCTETSTRING	
comp_part6	OCTETSTRING	
comp_part7	OCTETSTRING	
comp_part8	OCTETSTRING	
comp_part9	OCTETSTRING	
comp_part10	OCTETSTRING	
comp_part11	OCTETSTRING	
comp_part12	OCTETSTRING	
Detailed Comments:	This component is defined for the use in the downlink (from the test system to the IUT) direction.	

Structured Type Definition		
Type Name:	CPDATA	
Comments:	CP-User data element GSM 04.11, 8.1.4.1	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
rpack	RPACK	
rpdata	RPDATA	
rperr	RPERR	
rpsmma	RPSMMA	
Detailed Comments:	One of the 4 types of rpdu is contained in the CPDATA.	

Structured Type Definition		
Type Name:	CPHMS	
Comments:	Cipher mode setting GSM 04.08, 10.5.2.9	
Element Name	**Type Definition**	**Comments**
iei	IEI_4	
algid	BITSTRING [3]	
sc	BITSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CPHKSN	
Comments:	Ciphering key sequence number GSM 04.08, 10.5.1.2	
Element Name	**Type Definition**	**Comments**
sprb	BITSTRING [1]	
ks	BITSTRING [3]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CPH_RES	
Comments:	Cipher response GSM 04.08, 10.5.2.10	
Element Name	**Type Definition**	**Comments**
iei	IEI_4	
sprb	BITSTRING [3]	
cr	BITSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CSP	
Comments:	Cell selection parameters GSM 04.08, 10.5.2.4	
Element Name	**Type Definition**	**Comments**
crh	BITSTRING [3]	
mtmc	BITSTRING [5]	
acs	BITSTRING [1]	
neci	BITSTRING [1]	
ram	BITSTRING [6]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CST	
Comments:	Call status (CC information element) GSM 04.08, 10.5.4.6	
Element Name	**Type Definition**	**Comments**
cs	CS	
csv	BITSTRING [6]	
Detailed Comments:		

Structured Type Definition		
Type Name:	FIE	
Comments:	Facility information element GSM 04.80, 3.6	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
components_1	Components	
components_t	Component_T	
Detailed Comments:	When sending normally only one component is sent, but when receiving any number of components can be received even though normally we are only interested in one component during the testing.	

Structured Type Definition		
Type Name:	FN	
Comments:	The time in reference to the frame numbering corresponding to the absolute frame number modulo 42432.	
Element Name	**Type Definition**	**Comments**
t1_	T1_	
t3	T3	
t2	T2	
Detailed Comments:	In STRT and RQR	

Structured Type Definition		
Type Name:	FRQCHS	
Comments:	Frequency channel sequence GSM 04.08, 10.5.2.12	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
larfcn	OCTETSTRING [1]	
incs	OCTETSTRING [8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	FRQL	
Comments:	Frequency list, frequency short list GSM 04.08, 10.5.2.5, 10.5.2.6	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
fl	OCTETSTRING	
Detailed Comments:	frequency short list has a fixed length of 10 octets and does not contain the length indicator.	

Structured Type Definition		
Type Name:	FRQPARA	
Comments:	Parameters for Description of basic physical channel in frequency domain.	
Element Name	**Type Definition**	**Comments**
hch	BITSTRING[1]	
maio	BITSTRING [6]	
hsn	BITSTRING [6]	
spr	BITSTRING [2]	
arfcn	BITSTRING [10]	
maclength	LENGTH	
mac_8n	BITSTRING [8]	
mac_7n	BITSTRING [8]	
mac_6n	BITSTRING [8]	
mac_5n	BITSTRING [8]	
mac_4n	BITSTRING [8]	
mac_3n	BITSTRING [8]	
mac_2n	BITSTRING [8]	
mac_1n	BITSTRING [8]	
flst	FRQL	
Detailed Comments:		

Structured Type Definition		
Type Name:	HLCMP	
Comments:	High layer compatibility (CC information element) GSM 04.08, 10.5.4.16, ITU Q.931	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
cs	BITSTRING [2]	
in	BITSTRING [3]	
pmpp	BITSTRING [2]	
extb4	EXTB	
hlci	BITSTRING [7]	
extb4a	EXTB	
ehlci	BITSTRING [7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	HOACC_PARA	
Comments:	Handover Access parameters	
Element Name	**Type Definition**	**Comments**
num_bursts	OCTETSTRING[1]	
min_mspwr	OCTETSTRING[1]	
max_mspwr	OCTETSTRING[1]	
min_msta	BITSTRING[8]	
max_msta	BITSTRING[8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	IARESTOCT	
Comments:		
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iaroct1	OCTETSTRING[1..11]	
iaroct2	IARESTOCT2	
Detailed Comments:		

Structured Type Definition		
Type Name:	IARESTOCT2	
Comments:		
Element Name	**Type Definition**	**Comments**
p	BITSTRING[2]	
frqparalen	BITSTRING[6]	
spbt1	BITSTRING[2]	
maio	BITSTRING[6]	
ma	OCTETSTRING[0..8]	
spbt2	OCTETSTRING[0..8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	KPF (keypad facility)	
Comments:	keypad facility GSM 04.08 clause 10.5.4.17	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
extb	EXTB	
kpf_info	IA5String [1]	
Detailed Comments:	Is extb needed?	

Structured Type Definition		
Type Name:	L1HD	
Comments:	Layer 1 Header	
Element Name	**Type Definition**	**Comments**
spr1	BITSTRING[3]	
mspwrlvl	BITSTRING[5]	
spr2	BITSTRING[1]	
ta	BITSTRING[7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	LAI	
Comments:	Location area identification GSM 04.08, 10.5.1.3	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
mcc	OCTETSTRING [2]	
mnc	OCTETSTRING [1]	
lac	OCTETSTRING [2]	
Detailed Comments:		

Structured Type Definition		
Type Name:	LLCMP	
Comments:	Low layer compatibility (CC information element) GSM 04.08, 10.5.4.18 and ETS DE/SPS-5034-1 (update of ETS 300 102-1)	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
cs	BITSTRING [2]	
itc	BITSTRING [5]	
extb3a	EXTB	
negind	BITSTRING[1]	
spb3a	SP6B	
extb4	EXTB	
tm	BITSTRING [2]	
itr	BITSTRING [5]	
extb4_1	EXTB	
rateml	BITSTRING [7]	
extb5	EXTB	
l1id	BITSTRING [2]	
uil1	BITSTRING [5]	
extb5a	EXTB	
sb	BITSTRING [1]	
neg	BITSTRING [1]	
ur	BITSTRING [5]	
extb5b1	EXTB	
ir	BITSTRING [2]	
nictx	BITSTRING [1]	
nicrx	BITSTRING [1]	
fctx	BITSTRING [1]	
fcrx	BITSTRING [1]	
spb5b1	SPB	
extb5b2	EXTB	
hdrb	BITSTRING[1]	
mfs	BITSTRING[1]	
mode	BITSTRING[1]	
llineg	BITSTRING[1]	
ass	BITSTRING[1]	
ibob	BITSTRING[1]	
spb5b2	SPB	
extb5c	EXTB	
nsb	BITSTRING [2]	
ndb	BITSTRING [2]	
pi	BITSTRING [3]	
extb5d	EXTB	
dplxm	BITSTRING [1]	
modemt	BITSTRING [6]	
extb6	EXTB	
l2id	BITSTRING [2]	
uil2	BITSTRING [5]	
extb6a1	EXTB	
modl2	BITSTRING[2]	
spb6a1	SP3B	
q933	BITSTRING [2]	
extb6a2	EXTB	
usl2pi	BITSTRING[7]	
extb6b	EXTB	
ws	BITSTRING[7]	
extb7	EXTB	
l3id	BITSTRING [2]	
uil3	BITSTRING [5]	
extb7a1	EXTB	
modl3	BITSTRING[2]	
spb7a1	SP5B	

extb7a2	EXTB	
usl3pi	BITSTRING[7]	
extb7b	EXTB	
spb7b	SP3B	
dps	BITSTRING [4]	
extb7c	EXTB	
pws	BITSTRING[7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	LUT	
Comments:	Location updating type GSM 04.08, 10.5.3.5	
Element Name	**Type Definition**	**Comments**
foreq	BITSTRING [1]	
sprb	BITSTRING [1]	
lut	BITSTRING [2]	
Detailed Comments:		

Structured Type Definition		
Type Name:	MA	
Comments:	Mobile allocation GSM 04.08, 10.5.2.21	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
mac_8n	BITSTRING [8]	
mac_7n	BITSTRING [8]	
mac_6n	BITSTRING [8]	
mac_5n	BITSTRING [8]	
mac_4n	BITSTRING [8]	
mac_3n	BITSTRING [8]	
mac_2n	BITSTRING [8]	
mac_1n	BITSTRING [8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	MI	
Comments:	Mobile identity GSM 04.08, 10.5.1.4	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
idigit_1	BITSTRING [4]	
oei	BITSTRING [1]	
toi	BITSTRING [3]	
idigits_other	OCTETSTRING [0..7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	MSCLM1	
Comments:	mobile station classmark 1 GSM 04.08, 10.5.1.5	
Element Name	**Type Definition**	**Comments**
spr1	BITSTRING [1]	
rl	BITSTRING [2]	
spr2	BITSTRING [1]	
a5_1	BITSTRING [1]	
rfpc	BITSTRING [3]	
Detailed Comments:	2 octets long	

Structured Type Definition		
Type Name:	MSCLM2	
Comments:	mobile station classmark 2 GSM 04.08, 10.5.1.6	
Element Name	**Type Definition**	**Comments**
iel	LENGTH	
spr1	BITSTRING [1]	
rl	BITSTRING [2]	
spr2	BITSTRING [1]	
a5_1	BITSTRING [1]	
rfpc	BITSTRING [3]	
spr3	BITSTRING [1]	
psc	BITSTRING [1]	
sssi	BITSTRING [2]	
smc	BITSTRING [1]	
spr4	BITSTRING [2]	
fc	BITSTRING [1]	
cm3	BITSTRING [1]	
spr5	BITSTRING [5]	
a5_3	BITSTRING [1]	
a5_2	BITSTRING [1]	
Detailed Comments:	4 octets long	

Structured Type Definition		
Type Name:	MSCLM3	
Comments:	Mobile station classmark 3 GSM 04.08, 10.5.1.7	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
spr1	BITSTRING [4]	
a5_7	BITSTRING [1]	
a5_6	BITSTRING [1]	
a5_5	BITSTRING [1]	
a5_4	BITSTRING [1]	
spr2	OCTETSTRING [11]	
Detailed Comments:	The info element has 14 octets	

Structured Type Definition		
Type Name:	MSRR	
Comments:	measurement results GSM 04.08, 10.5.2.20	
Element Name	**Type Definition**	**Comments**
ba_used	BITSTRING [1]	
dtx_used	BITSTRING [1]	
rxlev_fsc	RXLEV	
spr1	BITSTRING [1]	
meas_valid	BITSTRING [1]	
rxlev_ssc	RXLEV	
spr2	BITSTRING [1]	
rxqual_fsc	BITSTRING [3]	
rxqual_ssc	BITSTRING [3]	
no_nc	BITSTRING [3]	
rxlev_nc1	RXLEV	
bcchfrq_nc1	BCCHFRQ	
bsic_nc1	BSIC	
rxlev_nc2	RXLEV	
bcchfrq_nc2	BCCHFRQ	
bsic_nc2	BSIC	
rxlev_nc3	RXLEV	
bcchfrq_nc3	BCCHFRQ	
bsic_nc3	BSIC	
rxlev_nc4	RXLEV	
bcchfrq_nc4	BCCHFRQ	
bsic_nc4	BSIC	
rxlev_nc5	RXLEV	
bcchfrq_nc5	BCCHFRQ	
bsic_nc5	BSIC	
rxlev_nc6	RXLEV	
bcchfrq_nc6	BCCHFRQ	
bsic_nc6	BSIC	
Detailed Comments:	The info element has a fixed length of 16 octets.	

Structured Type Definition		
Type Name:	MTDIF	
Comments:	Mobile time difference GSM 04.08, 10.5.1.21a	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
value	OCTETSTRING [3]	
Detailed Comments:		

Structured Type Definition		
Type Name:	NCD	
Comments:	Neighbour cells description GSM 04.08, 10.5.2.22	
Element Name	**Type Definition**	**Comments**
rfl2	BITSTRING [2]	
extind	BITSTRING [1]	
baind	BITSTRING [1]	
rfl4	BITSTRING [4]	
rfl	OCTETSTRING [15]	
Detailed Comments:		

Structured Type Definition		
Type Name:	NCD2	
Comments:	Neighbour cells description2 GSM 04.08, 10.5.2.22b	
Element Name	**Type Definition**	**Comments**
b128	BITSTRING [1]	
mbrpt	BITSTRING [2]	
baind	BITSTRING [1]	
b122_124	BITSTRING [3]	
rfl1	BITSTRING [1]	
rfl	OCTETSTRING [15]	
Detailed Comments:		

Structured Type Definition		
Type Name:	PCMD	
Comments:	Power command and access type GSM 04.08, 10.5.2.28, 10.5.2.28a	
Element Name	**Type Definition**	**Comments**
sprb	BITSTRING [3]	
pl	BITSTRING [5]	
Detailed Comments:		

Structured Type Definition		
Type Name:	PI	
Comments:	Progress indicator GSM 04.08, 10.5.4.21	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
cs	BITSTRING [2]	
spb	SPB	
loc	BITSTRING [4]	
extb4	EXTB	
prd	BITSTRING [7]	
Detailed Comments:	total 4 octets	

Structured Type Definition		
Type Name:	PI_SI	
Comments:	Presentation indicator & screening indicator GSM 04.08, 10.5.4.9, 10.5.4.13	
Element Name	**Type Definition**	**Comments**
extb	EXTB	
pi	BITSTRING [2]	
sp3b	SP3B	
si	BITSTRING [2]	
Detailed Comments:		

Structured Type Definition		
Type Name:	PM	
Comments:	Page mode GSM 04.08, 10.5.2.26	
Element Name	**Type Definition**	**Comments**
sprb	B_2	
pgm	B_2	
Detailed Comments:		

Structured Type Definition		
Type Name:	RACHCP	
Comments:	Rach control parameters GSM 04.08, 10.5.2.29	
Element Name	**Type Definition**	**Comments**
maxrtx	BITSTRING [2]	
txint	BITSTRING [4]	
cba	BITSTRING [1]	
re	BITSTRING [1]	
acc_2	BITSTRING [5]	
ec	BITSTRING [1]	
acc_1	BITSTRING [10]	
Detailed Comments:	The info element has a fixed length of 3 octets.	

Structured Type Definition		
Type Name:	RPACK	
Comments:	SMS RP ACKNOWLEDGEMENT ms <-> n GSM 04.11, 7.3.3	
Element Name	**Type Definition**	**Comments**
sprb	SP5B	
rpmti	MTI	
rpmr	MR	
Detailed Comments:		

Structured Type Definition		
Type Name:	RPCAU	
Comments:	RP user data element GSM 04.11, 8.2.5.4	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb2	EXTB	
rpcau_class	BITSTRING [3]	
rpcau_va	BITSTRING [4]	
rpcau_di	OCTETSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	RPDATA	
Comments:	SMS RP DATA ms <-> n GSM 04.11, 7.3.1	
Element Name	**Type Definition**	**Comments**
sprb	SP5B	
rpmti	MTI	
rpmr	MR	
rpOaddr	CDPN	
rpDaddr	CDPN	
rpusrdat	RPUSRDAT	
Detailed Comments:		

Structured Type Definition		
Type Name:	RPERR	
Comments:	SMS RP ERROR ms <-> n GSM 04.11, 7.3.4	
Element Name	**Type Definition**	**Comments**
sprb	SP5B	
rpmti	MTI	
rpmr	MR	
rpcau	RPCAU	
rpusrdat	RPUSRDAT	
Detailed Comments:		

Structured Type Definition		
Type Name:	RPSMMA	
Comments:	SMS RP SMMA ms -> n GSM 04.11, 7.3.2	
Element Name	**Type Definition**	**Comments**
sprb	SP5B	
rpmti	MTI	
rpmr	MR	
Detailed Comments:	SM memory available	

Structured Type Definition		
Type Name:	RPUSRDAT	
Comments:	RP user data element GSM 04.11, 8.2.5.3, GSM 03.40, 9.2.2	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
tpdeliver	SMDLVR	
tpsubmit	SMSBMT	
tpstatus_rpt	SMST_RPT	
tpcommand	SMCMD	
tpdlvr_sbmt_rpt	SMDLVR_RPT	
Detailed Comments:	One of the six tpdu is contained in the RPUSRDATA. Since the structures of the messages SMS-Deliver-Report and SMS-Submit-Report are identical, they have been combined to tpdlvr_sbmt_rpt and therefore only five tp message types exist in this type definition.	

Structured Type Definition		
Type Name:	RQR	
Comments:	Request reference GSM 04.08, 10.5.2.30	
Element Name	**Type Definition**	**Comments**
ra	BITSTRING [8]	
fn	FN	
Detailed Comments:		

Structured Type Definition		
Type Name:	SI3RO	
Comments:	SI3 Rest Octets GSM 04.08, 10.5.2.34	
Element Name	**Type Definition**	**Comments**
pi	BITSTRING [1]	
cbq	BITSTRING [1]	
cro	BITSTRING [6]	
to	BITSTRING [3]	
pt	BITSTRING [5]	
poi	BITSTRING [1]	
po	BITSTRING [2]	
ti	BITSTRING [1]	
ecsc	BITSTRING [1]	
spr1	BITSTRING [3]	
spr2	BITSTRING [8]	
Detailed Comments:	The info element has a fixed length of 4 octets.	

Structured Type Definition		
Type Name:	SI4RO	
Comments:	SI4 Rest Octets GSM 04.08, 10.5.2.35	
Element Name	**Type Definition**	**Comments**
pi	BITSTRING [1]	
cbq	BITSTRING [1]	
cro	BITSTRING [6]	
to	BITSTRING [3]	
pt	BITSTRING [5]	
poi	BITSTRING [1]	
po	BITSTRING [2]	
spr1	BITSTRING [5]	
spr2	BITSTRING [8]	
spr3	BITSTRING [8]	
spr4	BITSTRING [8]	
spr5	BITSTRING [8]	
spr6	BITSTRING [8]	
spr7	BITSTRING [8]	
spr8	BITSTRING [8]	
Detailed Comments:	The info element has a fixed length of 10 octets.	

Structured Type Definition		
Type Name:	SI7RO	
Comments:	SI7 Rest Octets GSM 04.08, 10.5.2.36	
Element Name	**Type Definition**	**Comments**
pi	BITSTRING [1]	
cbq	BITSTRING [1]	
cro	BITSTRING [6]	
to	BITSTRING [3]	
pt	BITSTRING [5]	
poi	BITSTRING [1]	
po	BITSTRING [2]	
spr1	BITSTRING [5]	
spr2	BITSTRING [8]	
spr3	BITSTRING [8]	
spr4	BITSTRING [8]	
spr5	BITSTRING [8]	
spr6	BITSTRING [8]	
spr7	BITSTRING [8]	
spr8	BITSTRING [8]	
spr9	BITSTRING [8]	
spr10	BITSTRING [8]	
spr11	BITSTRING [8]	
spr12	BITSTRING [8]	
spr13	BITSTRING [8]	
spr14	BITSTRING [8]	
spr15	BITSTRING [8]	
spr16	BITSTRING [8]	
spr17	BITSTRING [8]	
spr18	BITSTRING [8]	
Detailed Comments:	The info element has a fixed length of 20 octets.	

Structured Type Definition		
Type Name:	SIGNAL	
Comments:	Signal (CC information element) GSM 04.08, 10.5.4.23	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
sigv	BITSTRING [8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	SERIAL_NUMBER	
Comments:	Serial number for SMSCB, GSM 3.41, 9.3.2	
Element Name	**Type Definition**	**Comments**
gs	BITSTRING[2]	
message_code	BITSTRING[10]	
update_number	BITSTRING[4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMCMD	
Comments:	SMS COMMAND, ms - > n. GSM 03.40, 9.2.2.4	
Element Name	**Type Definition**	**Comments**
sprb1	SP2B	
srr	BITSTRING[1]	
sprb2	SP3B	
mti	BITSTRING [2]	
mr	MR	
pid	TPPID	
ct	OCTETSTRING [1]	
mn	OCTETSTRING [1]	
da	TPA	
cdl	LENGTH	
cd	TPCD	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMDLVR	
Comments:	SMS DELIVER, n - > ms. GSM 03.40, 9.2.2.1	
Element Name	**Type Definition**	**Comments**
rp	BITSTRING [1]	
udhi	BITSTRING [1]	
sri	BITSTRING [1]	
sprb2	SP2B	
mms	BITSTRING [1]	
mti	BITSTRING [2]	
oa	TPA	
pid	TPPID	
dcs	TPDCS	
scts	TPSCTS	
udl	LENGTH	
ud	TPUD	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMDLVR_RPT	
Comments:	SMS DELIVER or SUBMIT REPORT contained in the RP ERROR PDU, n <-> ms. GSM 03.40, 9.2.2.1a, 9.2.2.2a.	
Element Name	**Type Definition**	**Comments**
sprb1	SP6B	
mti	BITSTRING [2]	
fcs	FCS	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMSBMT	
Comments:	SMS SUBMIT, ms -> n. GSM 03.40, 9.2.2.2	
Element Name	**Type Definition**	**Comments**
rp	BITSTRING [1]	
udhi	BITSTRING [1]	
srr	BITSTRING [1]	
vpf	BITSTRING [2]	
rd	BITSTRING [1]	
mti	BITSTRING [2]	
mr	MR	
da	TPA	
pid	TPPID	
dcs	TPDCS	
vp1	OCTETSTRING [1]	
vp7	TPSCTS	
udl	LENGTH	
ud	TPUD	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMSCBpack	
Comments:	SMS cell broadcasting packing data n -> ms. GSM 03.38, 6.1.2.2.1	
Element Name	**Type Definition**	**Comments**
block1	OCTETSTRING [16]	
block2	OCTETSTRING [22]	
block3	OCTETSTRING [22]	
block4	OCTETSTRING [22]	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMST_RPT	
Comments:	SMS STATUS REPORT, n - > ms. GSM 03.40, 9.2.2.3	
Element Name	**Type Definition**	**Comments**
sprb1	SP5B	
mms	BITSTRING [1]	
mti	BITSTRING [2]	
mr	MR	
ra	TPA	
scts	TPSCTS	
dt	TPSCTS	
st	TPST	
Detailed Comments:		

Structured Type Definition		
Type Name:	SSVI	
Comments:	SS version indicator GSM 04.08, 10.5.4.24	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
ssv	OCTETSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	STRT	
Comments:	Starting time GSM 04.08, 10.5.2.38	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
fn	FN	
Detailed Comments:		

Structured Type Definition		
Type Name:	SUBAD	
Comments:	Subaddress GSM 04.08, 10.5.4.8, 10.5.4.10, 10.5.4.14	
Element Name	**Type Definition**	**Comments**
extb	EXTB	
tos	BITSTRING [3]	
oei	BITSTRING [1]	
sp3b	SP3B	
si	OCTETSTRING [2..23]	
Detailed Comments:		

Structured Type Definition		
Type Name:	SYNCHI	
Comments:	Synchronization indication GSM 04.08, 10.5.2.39	
Element Name	**Type Definition**	**Comments**
iei	IEI_4	
nci	B_1	
rot	B_1	
si	B_2	
Detailed Comments:		

Structured Type Definition		
Type Name:	TA	
Comments:	Timing advance GSM 04.08, 10.5.2.40	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
sprb	BITSTRING [2]	
value	BITSTRING [6]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TDIF	
Comments:	Time difference GSM 04.08, 10.5.1.41	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
value	OCTETSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TI	
Comments:	Transaction identifier GSM 04.08, 10.1	
Element Name	**Type Definition**	**Comments**
ti_f	BITSTRING [1]	
ti_v	TI_V	
Detailed Comments:		

Structured Type Definition		
Type Name:	TMSI	
Comments:	Temporary GSM 04.08, 10.5.2.42	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
tmsi_val	OCTETSTRING [4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TON_NPI	
Comments:	Type of number and numbering plan GSM 04.08, 10.5.4.7, 10.5.4.9, 10.5.4.13, GSM 03.40, 9.1.2.5	
Element Name	**Type Definition**	**Comments**
extb	EXTB	
ton	BITSTRING [3]	
npi	BITSTRING [4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TPA	
Comments:	TP address GSM 03.40, 9.1.2.5	
Element Name	**Type Definition**	**Comments**
iel	LENGTH	
tonnpi	TON_NPI	
digits	BCDN	
Detailed Comments:		

Structured Type Definition		
Type Name:	TPDCS	
Comments:	SMS data coding scheme GSM 03.38, 4, 5	
Element Name	**Type Definition**	**Comments**
cg	BITSTRING [4]	
code	BITSTRING [4]	
Detailed Comments:	Identifying the coding scheme within the TP user data.	

Structured Type Definition		
Type Name:	TPPID	
Comments:	TP protocol identifier GSM 03.40, 9.2.3.9	
Element Name	**Type Definition**	**Comments**
type	BITSTRING [2]	
value	BITSTRING [6]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TPST	
Comments:	TP status GSM 03.40, 9.2.3.15	
Element Name	**Type Definition**	**Comments**
sprb1	SPB	
value	BITSTRING [7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	UNKWN	
Comments:	unknown IE	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
contents	OCTETSTRING	
Detailed Comments:		

Structured Type Definition		
Type Name:	UU	
Comments:	User-user GSM 04.08, 10.5.4.25	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
uupd	BITSTRING [8]	
uui	OCTETSTRING [1..128]	
Detailed Comments:	In SETUP, ALERTING, CONNECT, DISCONNECT, RELEASE and RELEASE COMPLETE messages the uui length is of 0 - 32 bytes. In USER INFORMATION messages the uui length is of 1 - 128.	

ASN1 type definitions

ASN.1 Type Definition	
Type Name:	Component
Comments:	ASN1_Encoding: BER
Type Definition	

```
CHOICE {
   registerSSComponents                          RegisterSS_Components,
   eraseSSComponents                              EraseSS_Components,
   activateSSComponents                          ActivateSS_Components,
   deactivateSSComponents                      DeactivateSS_Components,
   interrogateSSComponents                     InterrogateSS_Components,
   notifySSComponents                             NotifySS_Components,
   registerPasswordComponents                RegisterPassword_Components,
   getPasswordComponents                       GetPassword_Components,
   processUnstructuredSSDataComponents     ProcessUnstructuredSSData_Components,
   forwardCheckSSIndicationComponents        ForwardCheckSSIndication_Components,
   processUnstructuredSSRequestComponents ProcessUnstructuredSSRequest_Components,
   unstructuredSSRequestComponents           UnstructuredSSRequest_Components,
   unstructuredSSNotifyComponents              UnstructuredSSNotify_Components,
   forwardCUGInfoComponents                       ForwardCUGInfo_Components,
   splitMPTYComponents                                 SplitMPTY_Components,
   retrieveMPTYComponents                           RetrieveMPTY_Components,
   holdMPTYComponents                                 HoldMPTY_Components,
   buildMPTYComponents                                BuildMPTY_Components,
   forwardChargeAdviceComponents             ForwardChargeAdvice_Components,
   generalComponents                                     General_Components
}
```

Detailed Comments:	Plural components as each type represents invoke, return result, return error etc.

ASN.1 Type Definition	
Type Name:	Components
Comments:	GSM 04.80, 3.6
Type Definition	

```
SET OF Component
```

Detailed Comments:	ASN.1 transfer encoding rules: BER is not wholy used for the type Components. The contents of Components, without the octets encoding the tag and the length of SET OF, is carried as the Components value.

ASN.1 Type Definition

Type Name: ActivateSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       activateSS_InvokeComp          [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (12),
                                ss_ForBS                    SS_ForBS_Code },

       activateSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                        localValue                          INTEGER (12),
                                        ss_Info              SS_Info        }      } ,

       activateSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| dataMissing

| unexpectedDataValue

| negativePW_Check

| numberOfPW_AttemptsViolation) },
              callBarredErr                        [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_ErrorStatusErr             [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_SubscriptionViolationeErr [3] IMPLICIT SEQUENCE { invokeID       InvokeIDType,

errorCode     ErrorLocalValues (ss_SubscriptionViolation),

parameter      SS_SubscriptionOption },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                 [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       activateSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m (15 - 30 s)

ASN.1 Type Definition

Type Name: BuildMPTY_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       buildMPTY_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (124)  },

       buildMPTY_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                         localValue                          INTEGER (124) }
} ,

       buildMPTY_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSS_Operation

| ss_NotAvailable

| maxNumberOfMPTY_ParticipantsExceeded

| resourcesNotAvailable ) },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       buildMPTY_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_buildMPTY (5 - 30 s)

ASN.1 Type Definition

Type Name: DeactivateSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       deactivateSS_InvokeComp          [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (13),
                                ss_ForBS                    SS_ForBS_Code },

       deactivateSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                   InvokeIDType,
                                 result                        SEQUENCE {
                                        localValue                        INTEGER (13),
                                        ss_Info              SS_Info       }      } ,

       deactivateSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| dataMissing

| unexpectedDataValue

| negativePW_Check

| numberOfPW_AttemptsViolation) },
              callBarredErr                       [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_ErrorStatusErr             [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_SubscriptionViolationeErr [3] IMPLICIT SEQUENCE { invokeID       InvokeIDType,

errorCode     ErrorLocalValues (ss_SubscriptionViolation),

parameter      SS_SubscriptionOption },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       deactivateSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: EraseSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       eraseSS_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (11),
                                ss_ForBS                    SS_ForBS_Code },

       eraseSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                        localValue                        INTEGER (11),
                                        ss_Info              SS_Info    }             } ,

       eraseSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues (  bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| dataMissing

| unexpectedDataValue ) },
              callBarredErr                       [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              systemFailureErr                [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },


       eraseSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition	
Type Name:	ForwardChargeAdvice_Components
Comments:	GSM 04.80, 4.2

Type Definition

```
CHOICE {
       forwardChargeAdvice_InvokeComp           [1] IMPLICIT SEQUENCE {
                           invokeID                                        InvokeIDType,
                           localValue                                       INTEGER (125),
                           forwardChargeAdviceArg              SEQUENCE {
                                                ss_Code                       [0] SS_Code,
                                                chargingInformation     [1]  SEQUENCE{
                                                      e1       [1] INTEGER
(0..max10TimesUnitsPerTime)                        OPTIONAL,
                                                      e2       [2]  INTEGER
(0..max10TimesTimeInterval)                         OPTIONAL,
                                                      e3       [3]  INTEGER
(0..max100TimesScalingFactor)                     OPTIONAL,
                                                      e4       [4] INTEGER
(0..max10TimesIncrement)                             OPTIONAL,
                                                      e5       [5]  INTEGER
(0..max10TimesIncrementPerDataInterval)    OPTIONAL,
                                                      e6       [6] INTEGER
(0..maxNumberOfSegmentsPerDataInterval) OPTIONAL,
                                                      e7       [7]  INTEGER
(0..max10TimesInitialTime)                           OPTIONAL } } },

       forwardChargeAdvice_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                           invokeID                    InvokeIDType,
                           result                        SEQUENCE {
                           localValue                   INTEGER (125)  }    } ,

       forwardChargeAdvice_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_aoc = 1 - 40 s

ASN.1 Type Definition	
Type Name:	ForwardCheckSSIndication_Components
Comments:	GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       forwardCheckSSIndication_InvokeComp           [1] IMPLICIT SEQUENCE {
                              invokeID                      InvokeIDType,
                              localValue                     INTEGER (38)    },

      forwardCheckSSIndication_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	ForwardCUGInfo_Components
Comments:	GSM 04.80, 4.2

Type Definition

```
CHOICE {
       forwardCUGInfo_InvokeComp          [1] IMPLICIT SEQUENCE {
                              invokeID                     InvokeIDType,
                              localValue                    INTEGER (120),
                              forwardCUGInfo_Arg            SEQUENCE{
                                       cug_Index             [0]  IMPLICIT CUG_Index
OPTIONAL,
                                       suppressPrefCUG   [1]  IMPLICIT NULL           OPTIONAL,
                                       suppressOA            [2]  IMPLICIT NULL
OPTIONAL } },

        forwardCUGInfo_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	General_Components
Comments:	Non specified components must match this type definition.

Type Definition

```
CHOICE {
  general_InvokeComp          [1] IMPLICIT General_InvokeComponent,
  general_ReturnResultComp [2] IMPLICIT General_ReturnResultComponent,
  general_ReturnErrorComp  [3] IMPLICIT General_ReturnErrorComponent,
  general_RejectComp           [4] IMPLICIT RejectComponent }


-- This is the General InvokeComponent --
General_InvokeComponent ::= SEQUENCE {
  invokeID                     InvokeIDType,
  linked_ID       [0] IMPLICIT InvokeIDType OPTIONAL,
  operation_value              Operation,
  argument                     ANY     OPTIONAL }

-- This is the General ReturnResultComponent --
General_ReturnResultComponent ::= SEQUENCE {
  invokeID       InvokeIDType,
  valueAndResult SEQUENCE {
                    operation_value Operation,
                    result          ANY     } OPTIONAL }

-- This is the General ReturnErrorComponent --
General_ReturnErrorComponent ::= SEQUENCE {
  invokeID InvokeIDType,
  error    ANY }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	GetPassword_Components
Comments:	GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       getPassword_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                linkedID                       [0] IMPLICIT InvokeIDType,
                                localValue                     INTEGER (18),
                                guidanceInfo                ENUMERATED {
                                                                    enterPW  (0),
                                                                    enterNewPW  (1),
                                                                    enterNewPW_Again  (2) }
},

       getPassword_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                        localValue                        INTEGER (18),
                                        currentPassword              Password }            } ,


       getPassword_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition	
Type Name:	HoldMPTY_Components
Comments:	GSM 04.80, 4.2

Type Definition

```
CHOICE {
       holdMPTY_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (123)  },

       holdMPTY_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                         localValue                          INTEGER (123) }
} ,

       holdMPTY_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSS_Operation

| facilityNotSupported ) },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       holdMPTY_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments:	Timer T_holdMPTY (5 - 30 s)

ASN.1 Type Definition

Type Name: InterrogateSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       interrogateSS_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (14),
                                ss_ForBS                    SS_ForBS_Code },

       interrogateSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                   InvokeIDType,
                                 result                        SEQUENCE {
                                         localValue                        INTEGER (14),
                                         interrogateSS_Res             CHOICE {
                                              ss_Status                     [0] IMPLICIT
SS_Status,
                                              basicServiceGroupList   [2] IMPLICIT
BasicServiceGroupList,
                                              forwardingFeatureList   [3] IMPLICIT
ForwardingFeatureList,
                                              cli_RestrictionInfo         [4] IMPLICIT SEQUENCE {
                                                                           ss_Status
SS_Status,
                                                                           cliRestrictionOption
CliRestrictionOption    OPTIONAL } } } } ,

       interrogateSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| ss_NotAvailable

| dataMissing

| unexpectedDataValue) },
              callBarredErr                       [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },


       interrogateSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition	
Type Name:	NotifySS_Components
Comments:	GSM 04.80, 4.2

Type Definition

```
CHOICE {
       notifySS_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (16),
                                notifySS_Arg              SEQUENCE{
                                        ss_Code                        [1] IMPLICIT SS_Code
OPTIONAL,
                                        ss_Status                      [4] IMPLICIT SS_Status
OPTIONAL,
                                        ss_Notification              [5] IMPLICIT OCTET STRING
(SIZE (1)) OPTIONAL,
                                        callIsWaiting_Indicator   [14] IMPLICIT NULL
OPTIONAL,
                                        callOnHold_Indicator     [15] IMPLICIT ENUMERATED {

callRetrieved (0),

callOnHold (1) }         OPTIONAL,
                                        mpty_Indicator              [16] IMPLICIT NULL
OPTIONAL,
                                        cug_Index                     [17]  IMPLICIT CUG_Index
OPTIONAL,
                                        clirSuppressionRejected [18]  IMPLICIT NULL
OPTIONAL }    },

      notifySS_RejectComp             [4] IMPLICIT RejectComponent        }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	ProcessUnstructuredSSData_Components
Comments:	GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       processUnstructuredSSData_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (19),
                                ss_UserData               SS_UserData },

       processUnstructuredSSData_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                        SEQUENCE {
                                        localValue                          INTEGER (19),
                                        ss_UserData                   SS_UserData   }  } ,

       processUnstructuredSSData_ReturnErrorComp   CHOICE {
              errorCodes                        [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues (unexpectedDataValue) },
              systemFailureErr                [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       processUnstructuredSSData_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: ProcessUnstructuredSSRequest_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       processUnstructuredSSRequest_InvokeComp           [1] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 localValue                   INTEGER (59),
                                 ussd_Arg                  USSD_Arg },

       processUnstructuredSSRequest_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                  invokeID                   InvokeIDType,
                                  result                        SEQUENCE {
                                          localValue                      INTEGER (59),
                                          ussd_Res                     USSD_Res}            }
,

       processUnstructuredSSRequest_ReturnErrorComp   CHOICE {
              errorCodes                        [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues (dataMissing

| unexpectedDataValue

| unknownAlphabet) },
              callBarredErr                      [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              systemFailureErr                [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       processUnstructuredSSRequest_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: RegisterPassword_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       registerPassword_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (17),
                                ss_Code                      SS_Code      },

       registerPassword_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                        localValue                          INTEGER (17),
                                        newPassword                 Password    }         } ,

       registerPassword_ReturnErrorComp   CHOICE {
              errorCodes                              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( dataMissing

| unexpectedDataValue

| negativePW_Check

| numberOfPW_AttemptsViolation) },
              callBarredErr                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode      ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_SubscriptionViolationeErr  [3] IMPLICIT SEQUENCE { invokeID    InvokeIDType,

errorCode      ErrorLocalValues (ss_SubscriptionViolation),

parameter      SS_SubscriptionOption },
              systemFailureErr                    [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter     NetworkResource},
              pw_RegistrationFailureErr       [3] IMPLICIT SEQUENCE { invokeID    InvokeIDType,

errorCode     ErrorLocalValues ( pw_RegistrationFailure ),

parameter     PW_RegistrationFailureCause }   },

       registerPassword_RejectComp      [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition	
Type Name:	RegisterSS_Components
Comments:	GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       registerSS_InvokeComp          [1] IMPLICIT SEQUENCE {
                                invokeID                     InvokeIDType,
                                localValue                    INTEGER (10),
                                registerSS_Arg           SEQUENCE{
                                        ss_Code                         SS_Code,
                                        basicService                    BasicServiceCode
OPTIONAL,
                                        forwardedToNumber         [4] IMPLICIT AddressString
OPTIONAL,
                                        forwardedToSubaddress   [6] IMPLICIT
ISDN_SubaddressString       OPTIONAL,
                                        noReplyConditionTime      [5] IMPLICIT
NoReplyConditionTime        OPTIONAL     }   },

       registerSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                     InvokeIDType,
                                 result                          SEQUENCE {
                                         localValue                        INTEGER (10),
                                         ss_Info              SS_Info          }         } ,

       registerSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| dataMissing

| unexpectedDataValue ) },
              callBarredErr                       [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },


       registerSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition	
Type Name:	RejectComponent
Comments:	ITU-T Q.767

Type Definition

```
SEQUENCE {
  invokedID CHOICE {
                      derivable            InvokeIDType,
                      notDerivable      NULL },
  problem    CHOICE {
                      generalProblem         [0] IMPLICIT GeneralProblem,
                      invokeProblem          [1] IMPLICIT InvokeProblem,
                      returnResultProblem [2] IMPLICIT ReturnResultProblem,
                      returnErrorProblem  [3] IMPLICIT ReturnErrorProblem } }
```

Detailed Comments: Reject Component is not specific to any particular operation. The invokeID may be used to identify a specific operation.

ASN.1 Type Definition

Type Name: RetrieveMPTY_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       retrieveMPTY_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (122)  },

       retrieveMPTY_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                     InvokeIDType,
                                 result                          SEQUENCE {
                                         localValue                          INTEGER (122) }
} ,

       retrieveMPTY_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSS_Operation

| facilityNotSupported ) },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       retrieveMPTY_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_retrieveMPTY (5 - 30 s)

ASN.1 Type Definition

Type Name: SplitMPTY_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       splitMPTY_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (121)  },

       splitMPTY_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                         localValue                          INTEGER (121) }
} ,

       splitMPTY_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSS_Operation

| facilityNotSupported ) },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       splitMPTY_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_splitMPTY (5 - 30 s)

ASN.1 Type Definition

Type Name: UnstructuredSSNotify_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       unstructuredSSNotify_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (61),
                                ussd_Arg                    USSD_Arg                  },

       unstructuredSSNotify_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                   InvokeIDType,
                                 result                        SEQUENCE {
                                         localValue                          INTEGER (61) }  } ,

       unstructuredSSNotify_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSubscriber

| illegalEquipment

| absentSubscriber

| dataMissing

| unexpectedDataValue

| unknownAlphabet

| ussd_Busy) },
              systemFailureErr                 [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       unstructuredSSNotify_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition	
Type Name:	UnstructuredSSRequest_Components
Comments:	GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       unstructuredSSRequest_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (60),
                                ussd_Arg                     USSD_Arg                  },

       unstructuredSSRequest_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                        SEQUENCE {
                                         localValue                          INTEGER (60),
                                        ussd_Res                          USSD_Res       }  } ,

       unstructuredSSRequest_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSubscriber

| illegalEquipment

| absentSubscriber

| dataMissing

| unexpectedDataValue

| unknownAlphabet

| ussd_Busy) },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       unstructuredSSRequest_RejectComp             [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition	
Type Name:	AddressString
Comments:	GSM 09.02, 14.7.8

Type Definition

```
OCTET STRING (SIZE ( 1 .. maxAddressLength ))
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	Asn1Integer
Comments:	INTEGER of ASN.1 type

Type Definition

```
INTEGER
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	BasicServiceCode
Comments:	GSM 09.02, 14.7.8

Type Definition

```
CHOICE {
        bearerService   [2] IMPLICIT BearerServiceCode,
        teleservice        [3] IMPLICIT TeleserviceCode       }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	BasicServiceGroupList
Comments:	GSM 09.02, 14.7.8
Type Definition	
`SEQUENCE SIZE ( 1 .. maxNumOfBasicServiceGroups ) OF BasicServiceCode`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	BearerServiceCode
Comments:	GSM 09.02, 14.7.10
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CallBarringCause
Comments:	GSM 09.02, 14.7.7
Type Definition	

`ENUMERATED {`
`  barringServiceActive  (0),`
`  operatorBarring  (1) }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CallBarringFeature
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`  basicService  BasicServiceCode  OPTIONAL,`
`  ss_Status  [4] IMPLICIT SS_Status  OPTIONAL }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CallBarringFeatureList
Comments:	GSM 09.02, 14.7.4
Type Definition	
`SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF CallBarringFeature`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CallBarringInfo
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`  ss_Code  SS_Code  OPTIONAL,`
`  callBarringFeatureList  CallBarringFeatureList  }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CliRestrictionOption
Comments:	GSM 09.02, 14.7.4
Type Definition	
ENUMERATED { permanent (0), temporaryDefaultRestricted (1), temporaryDefaultAllowed (2) }	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_Feature
Comments:	GSM 09.02, 14.7.4
Type Definition	
SEQUENCE { basicService BasicServiceCode OPTIONAL, preferentialCUG_Indicator CUG_Index OPTIONAL, interCUG_Restrictions InterCUG_Restrictions }	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_FeatureList
Comments:	GSM 09.02, 14.7.4
Type Definition	
SEQUENCE SIZE (1.. maxNumOfBasicServiceGroups) OF CUG_Feature	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_Index
Comments:	GSM 09.02, 14.7.4
Type Definition	
INTEGER (0..32767)	
Detailed Comments:	The internal structure is defined in ETS 300 138.

ASN.1 Type Definition	
Type Name:	CUG_Interlock
Comments:	GSM 09.02, 14.7.4
Type Definition	
OCTET STRING (SIZE (4))	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_Info
Comments:	GSM 09.02, 14.7.4
Type Definition	
SEQUENCE { cug_SubscriptionList CUG_SubscriptionList, cug_FeatureList CUG_FeatureList OPTIONAL }	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_Subscription
Comments:	GSM 09.02, 14.7.4

Type Definition

```
SEQUENCE {
        cug_Index                       CUG_Index,
        cug_Interlock                  CUG_Interlock,
        intraCUG_Options         IntraCUG_Options,
        basicServiceGroupList    BasicServiceGroupList    OPTIONAL }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	CUG_SubscriptionList
Comments:	GSM 09.02, 14.7.4

Type Definition

```
SEQUENCE SIZE (1.. maxNumOfCUG ) OF CUG_Subscription
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	ErrorLocalValues
Comments:	GSM 04.80, 4.5

Type Definition

```
INTEGER {
  unknownSubscriber                                 (1),
  illegalSubscriber                                    (9),
  bearerServiceNotProvisioned                    (10),
  teleserviceNotProvisioned                        (11),
  illegalEquipment                                    (12),
  callBarred                                             (13),
  illegalSS_Operation                                (16),
  ss_ErrorStatus                                      (17),
  ss_NotAvailable                                     (18),
  ss_SubscriptionViolation                         (19),
  ss_Incompatibility                                  (20),
  facilityNotSupported                             (21),
  absentSubscriber                                    (27),
  systemFailure                                          (34),
  dataMissing                                            (35),
  unexpectedDataValue                              (36),
  pw_RegistrationFailure                            (37),
  negativePW_Check                                 (38),
  numberOfPW_AttemptsViolation                (43),
  unknownAlphabet                                    (71),
  ussd_Busy                                              (72),
  maxNumberOfMPTY_ParticipantsExceeded  (126),
  resourcesNotAvailable                             (127)
 }
```

Detailed Comments: Elements of INTEGER are global for the ATS.

ASN.1 Type Definition	
Type Name:	ForwardingFeature
Comments:	GSM 09.02, 14.7.4

Type Definition

```
SEQUENCE {
        basicService                    BasicServiceCode                    OPTIONAL,
        ss_Status                        [4] IMPLICIT SS_Status
OPTIONAL,
        forwardedToNumber         [5] IMPLICIT ISDN_AddressString          OPTIONAL,
        forwardedToSubaddress   [8] IMPLICIT ISDN_SubaddressString     OPTIONAL,
        forwardingOptions           [6] IMPLICIT ForwardingOptions             OPTIONAL,
        noReplyConditionTime      [7] IMPLICIT NoReplyConditionTime         OPTIONAL }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	ForwardingFeatureList
Comments:	GSM 09.02, 14.7.4
Type Definition	
`SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF ForwardingFeature`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	ForwardingInfo
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`  ss_Code  SS_Code  OPTIONAL,`
`  forwardingFeatureList  ForwardingFeatureList }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	ForwardingOptions
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	GeneralProblem
Comments:	ITU-T Q.767
Type Definition	

`TCAP_Problems ( unrecognizedComponent`
`  | mistypedComponent`
`  | badlyStructuredComponent )`	

Detailed Comments:	Type restricted to these three.

ASN.1 Type Definition	
Type Name:	InterCUG_Restrictions
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	bits: 876543: 000000 (unused) bits 21: 00 CUG only facilities 01 CUG with outgoing access 10 CUG with incoming access 11 CUG with both outgoing and incoming access

ASN.1 Type Definition	
Type Name:	IntraCUG_Options
Comments:	GSM 09.02, 14.7.4
Type Definition	

`ENUMERATED {`
`  noCUG_Restrictions  (0),`
`  cugIC_CallBarred  (1),`
`  cugOG_CallBarred  (2) }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	InvokeIDType
Comments:	ITU-T, Q.773
Type Definition	
`INTEGER (-128 .. 127)`	
Detailed Comments:	Values: Sending Components: If it is an invoke component then use Test Case Variable (with default) to set value. If another invoke component is sent the TCV should be incremented beforehand. If it is a return result, error or reject component in response to a received invoke component then use TCV also, making sure the value is set to the value of the received component beforehand. Receiving Components: If it is an invoke comp then use '?'. If it is a return result, error or reject component in response to a sent invoke component then use TCV value (as used in sent invoke component).

ASN.1 Type Definition	
Type Name:	InvokeProblem
Comments:	ITU-T Q.767
Type Definition	

`TCAP_Problems ( duplicateInvokeID`
`| unrecognizedOperation`
`| mistypedArgument`
`| resourceLimitation`
`| initiatingRelease`
`| unrecognizedLinkedID`
`| linkedResponseUnexpected`
`| unexpectedLinkedOperation )`	

Detailed Comments:	Type restricted to these 8.

ASN.1 Type Definition	
Type Name:	ISDN_AddressString
Comments:	GSM 09.02, 14.7.8
Type Definition	
`AddressString (SIZE ( 1 .. maxISDN_AddressLength ))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	ISDN_SubaddressString
Comments:	GSM 09.02, 14.7.8
Type Definition	
`OCTET STRING (SIZE ( 1 .. maxISDN_SubaddressLength ))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	NetworkResource
Comments:	GSM 09.02, 14.7.8,
Type Definition	

`ENUMERATED {`
`plmn (0),`
`hlr (1),`
`vlr (2),`
`pvlr (3),`
`controllingMSC (4),`
`vmsc (5),`
`eir (6),`
`rss (7) }`	

Detailed Comments:	Elements of INTEGER are global for the ATS.

ASN.1 Type Definition	
Type Name:	NoReplyConditionTime
Comments:	GSM 09.02, 14.7.4
Type Definition	
`INTEGER (5 .. 30 )`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	Operation
Comments:	ITU-T Q.767
Type Definition	

`CHOICE {`
`  localValue  INTEGER,`
`  globalValue OBJECT IDENTIFIER}`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	OverrideCategory
Comments:	GSM 09.02, 14.7.4
Type Definition	

`ENUMERATED {`
`  overrideEnabled  (0),`
`  overrideDisabled  (1) }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	Password
Comments:	GSM 09.02, 14.7.4
Type Definition	
`NumericString (FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"))  (SIZE (4))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	PW_RegistrationFailureCause
Comments:	GSM 09.02, 14.7.7
Type Definition	

`ENUMERATED {`
`  undetermined  (0),`
`  invalidFormat  (1),`
`  newPasswordsMismatch  (2) }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	ReturnErrorProblem
Comments:	ITU-T Q.767
Type Definition	

`TCAP_Problems ( unrecognizedInvokeID`
`| returnErrorUnexpected`
`| unrecognizedError`
`| unexpectedError`
`| mistypedParameter )`	

Detailed Comments:	Type restricted to these 5.

ASN.1 Type Definition	
Type Name:	ReturnResultProblem
Comments:	ITU-T Q.767
Type Definition	

`TCAP_Problems ( unrecognizedInvokeID`
`| returnResultUnexpected`
`| mistypedResult )`	

Detailed Comments:	Type restricted to these three.

ASN.1 Type Definition	
Type Name:	SS_Code
Comments:	GSM 09.02, 14.7.5
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	group (bits 8765), and specific service (bits 4321)

ASN.1 Type Definition	
Type Name:	SS_Data
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`ss_Code SS_Code OPTIONAL,`
`ss_Status [4] IMPLICIT SS_Status OPTIONAL,`
`ss_SubscriptionOption SS_SubscriptionOption OPTIONAL,`
`basicServiceGroupList BasicServiceGroupList OPTIONAL }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_ForBS_Code
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`ss_Code SS_Code,`
`basicService BasicServiceCode OPTIONAL }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_IncompatibilityCause
Comments:	GSM 09.02, 14.7.8
Type Definition	

`SEQUENCE {`
`ss_Code [1] IMPLICIT SS_Code OPTIONAL,`
`basicService BasicServiceCode OPTIONAL,`
`ss_Status [4] SS_Status OPTIONAL }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_Info
Comments:	GSM 09.02, 14.7.4
Type Definition	

`CHOICE {`
`forwardingInfo [0] IMPLICIT ForwardingInfo,`
`callBarringInfo [1] IMPLICIT CallBarringInfo,`
`cug_Info [2] IMPLICIT CUG_Info,`
`ss_Data [3] IMPLICIT SS_Data }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_Status
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	bits 8765: 0000 unused, bits 4: SS state information Q bit, bits 3: SS state information P bit, bits 2: SS state information R bit, bits 1: SS state information A bit.

ASN.1 Type Definition	
Type Name:	SS_SubscriptionOption
Comments:	GSM 09.02, 14.7.4
Type Definition	

```
CHOICE {
        overrideCategory        [1] IMPLICIT OverrideCategory,
        cliRestrictionOption     [2] IMPLICIT CliRestrictionOption }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	SS_UserData
Comments:	GSM 09.02, 14.7.4
Type Definition	

```
IA5String (SIZE (1  .. maxSignalInfoLength))
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	TCAP_Problems
Comments:	ITU-T Q.767
Type Definition	

```
INTEGER {
  unrecognizedComponent      (0),     -- GeneralProblem
  unrecognizedInvokeID          (0),     -- ReturnResultProblem, ReturnErrorProblem
  duplicateInvokeID                (0),     -- InvokeProblem
  mistypedComponent             (1),     -- GeneralProblem
  returnErrorUnexpected         (1),     -- ReturnErrorProblem
  returnResultUnexpected       (1),     -- ReturnResultProblem
  unrecognizedOperation         (1),     -- InvokeProblem
  badlyStructuredComponent  (2),     -- GeneralProblem
  unrecognizedError               (2),     -- ReturnErrorProblem
  mistypedArgument               (2),     -- InvokeProblem, originally called mistypedParameter
in TCAP
  mistypedResult                    (2),     -- ReturnResultProblem
  resourceLimitation               (3),     -- InvokeProblem
  unexpectedError                 (3),     -- ReturnErrorProblem
  mistypedParameter               (4),     -- ReturnErrorProblem
  initiatingRelease                    (4),     -- InvokeProblem
  unrecognizedLinkedID          (5),     -- InvokeProblem
  linkedResponseUnexpected  (6),     -- InvokeProblem
  unexpectedLinkedOperation  (7)     -- InvokeProblem
  }
```

Detailed Comments: Errors of the same integer value are distinguished by their different parent types (General, Invoke, ReturnResult, ReturnError).

ASN.1 Type Definition	
Type Name:	TeleserviceCode
Comments:	GSM 09.02, 14.7.9
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	USSD_Arg
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`  ussd_DataCodingScheme  USSD_DataCodingScheme,`
`  ussd_String  USSD_String }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	USSD_Res
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`  ussd_DataCodingScheme  USSD_DataCodingScheme,`
`  ussd_String  USSD_String }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	USSD_DataCodingScheme
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	USSD_String
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1..maxUSSD_StringLength))`	
Detailed Comments:	The structure of the contents of the USSD-String is dependent on the USSD-DataCodingScheme as described in TS GSM03.38.

Test suite operation definitions

Test Suite Operation Definition	
Operation Name:	OC_Asn1intToOct(n:Asn1Integer; l: INTEGER)
Result Type:	OCTETSTRING
Comments:	
Description	
OC_Asn1intToOct converts the ASN.1 INTEGER `n` into OCTETSTRING with length = `l`. for example: OC_Asn1intToOct(14,1) = '0E'O; OC_Asn1intToOct(18,1) = '12'O; OC_Asn1intToOct(18,2) = '0012'O; OC_Asn1intToOct(-128,1) = '80'O (MSB (position p out of 1...p) represents -2exp(p-1)); OC_Asn1intToOct(-32768,2) = '8000'O (MSB (position p out of 1...p) represents -2exp(p-1)).	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_Bcap(setup:SETUP_MO_PDU; callproc:CALL_PROC_PDU; mem:INTEGER)
Result Type:	BCAP
Comments:	
Description	
OC_Bcap operation returns a bearer capability IE according the following rule: - for `mem` = 1: - if bearer capability IE were presented in `callproc` the returned BC is the bearer capability 1 of the `callproc`; - otherwise, the returned bearer capability is the bearer capability 1 of the `setup`. - for `mem` = 2: - if bearer capability IE were presented in `callproc` the returned BC is the bearer capability 2 of the `callproc`; - otherwise, the returned bearer capability is the bearer capability 2 of the `setup`.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_BinAdd(bitstr1:BITSTRING; bitstr2:BITSTRING)
Result Type:	BITSTRING
Comments:	
Description	
OC_BinAdd operation performs binary addition of two input parameters `bitstr1` and `bitstr2`, then returns the result of the addition. These two input parameters shall have the same length, the result of the operation has the same length as the input parameters. for example: OC_BinAdd('01000'B, '00110'B) = '01110'B; OC_BinAdd('01000'B, '00011'B) = '01011'B; OC_BinAdd('00100'B, '00010'B) = '00110'B.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_Bit7(bstring: B_8)
Result Type:	BITSTRING
Comments:	The input parameter bstring is of type BITSTRING[8].
Description	
OC_Bit7(bstring) returns the value of bit 7 in the `bstring`. for example: OC_Bit7('01010101'B) = '1'B, OC_Bit7('10101010'B) = '0'B	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_BCDtoInt(bcdstring:HEXSTRING; relevant_digits: INTEGER)
Result Type:	INTEGER
Comments:	
Description	
The operation OC_BCDtoInt converts last 'relevant_digits' of an HEXSTRING containing BCD coded digits to an integer representation of these relevant digits. Example: OC_BCDtoInt('12345'H, 3) := 345	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_CalledNumCHK(callednum:OCTETSTRING; dialnum:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
the operation OC_CalledNumCHK to check whether the called party number `callednum`, which is represented by OCTETSTRING, is the same as the dialed number 'dialnum`, which is represented by IA5String. It returns TRUE if they are the same, otherwise FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_ChkSRES(sres:OCTETSTRING; ki: BITSTRING; rand:BITSTRING)
Result Type:	BOOLEAN
Comments:	sres is 32 bits value.
Description	
OC_ChkSRES checks the input parameter `sres` according to the authentication algorithm defined in the following procedure. It returns TRUE if the `sres` is correct, otherwise it returns FALSE. - firstly the `ki` XOR to the `rand` results in RES1; - then compare the most significant 32 bits of the RES1 with the `sres`; - if they are equal, the `sres` is correct and the operation returns TRUE; - if they are not equal, the `sres` is wrong and the operation returns FALSE. NOTE: this procedure is the test algorithm for authentication defined by GSM.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_CnvtMax(max:INTEGER)
Result Type:	B_2
Comments:	
Description	
The allowed integer values for max are 1, 2, 4, 7 (maximum number of retransmissions). The result BITSTRING is 2 bits long. OC_CnvtMax converts the input integer `max` into a BITSTRING according to the following rule: 1. the result is '00'B if the `max` = 1, 2. the result is '01'B if the `max` = 2, 3. the result is '10'B if the `max` = 4, 4. the result is '11'B if the `max` = 7. for example: OC_CnvtMax(1) = '00'B, OC_CvntMax(7) = '11'B.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_CnvtTx(tx:INTEGER)
Result Type:	B_4
Comments:	The result BITSTRING is 4 bits long.

Description

OC_CnvtTx converts the input integer `tx` into a BITSTRING according the following rule:

1. the result is '0000'B if the `tx` = 3,
2. the result is '0001'B if the `tx` = 4,
3. the result is '0010'B if the `tx` = 5,
4. the result is '0011'B if the `tx` = 6,
5. the result is '0100'B if the `tx` = 7,
6. the result is '0101'B if the `tx` = 8,
7. the result is '0110'B if the `tx` = 9,
8. the result is '0111'B if the `tx` = 10,
9. the result is '1000'B if the `tx` = 11,
10. the result is '1001'B if the `tx` = 12,
11. the result is '1010'B if the `tx` = 14,
12. the result is '1011'B if the `tx` = 16,
13. the result is '1100'B if the `tx` = 20,
14. the result is '1101'B if the `tx` = 25,
15. the result is '1110'B if the `tx` = 32,
16. the result is '1111'B if the `tx` = 50.

for example:

OC_CnvtTx(3) = '0000'B,
OC_CvntTx(5) = '0010'B.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_CodeSMSCBMessage(firstoct: INTEGER; lastoct: INTEGER)
Result Type:	OCTETSTRING
Comments:	

Description

The operation codes a part of the contents for a cell broadcast short message. The cell broadcast short message, is 82 octets long, i.e. 93 characters, each represented by 7 bits. As many as possible different characters are sent, the characters are those corresponding to the 7-bit representation of the integers 0, 1, 2, ..., 92. The bits are arranged acc. to GSM 03.38, clause 6.1.2.1.1. The result of this operation is the octetstring of the octets 'firstoct' to 'lastoct' (16 octets for the first message block, 22 octets for the 2nd, 3rd and 4th blocks), with the octets of the cell broadcast short message being numbered from 1 to 82.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_CodingOfUssdString(text: IA5String)
Result Type:	OCTETSTRING
Comments:	

Description

This operation provides the coding for a USSD String 'text' in the default alpabet, each character represented by 7 bits. The bits are arranged acc. to GSM 03.38, clause 6.1.2.2.1.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_ComputeSMContents(NumberOfCharacters: INTEGER)
Result Type:	OCTETSTRING
Comments:	max. 160 characters, i.e. 140 octets.
Description	
This operation provides a short message's contents with a specified number of characters 'NumberOfCharacters', each represented by 7 bits. As possible different characters are sent, the characters are those corresponding to the 7-bit representation of 0, 1, 2, ... up to ('NumberOfCharacters' - 1). If more than 128 characters are sent, the rest of the characters is the corresponding to 0, 1, ... up to (NumberOfCharacters - 128 - 1), e.g. for 160 characters: 0, 1, ..., 127, 0, 1, ..., 31. The bits are arranged acc. to GSM 03.38, clause 6.1.2.1.1.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_ComputeSMContentsSpecText(NumbOfIA5chara: INTEGER; text: IA5String)
Result Type:	OCTETSTRING
Comments:	max. 160 characters, i.e. 140 octets.
Description	
This operation provides a short message's contents with a specified number of characters 'NumbOfIA5chara', each represented by 7 bits. 'text' is used as contents of the short message. If "text' contains less than 'NumberOfCharacters' characters, 'text' is repeated until the short message reaches the 'NumberOfCharacters' characters long. The bits packing is arranged according. to GSM 03.38, clause 6.1.2.1.1.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_CphKeyGen(ki:BITSTRING; rand:BITSTRING)
Result Type:	BITSTRING
Comments:	both ki and rand are 128 bits values, the result of the operation is 64 bits value
Description	
OC_CphKeyGen generates the ciphering key from the input parameters according to the following procedure: - firstly the `ki` XOR to the `rand` results in RES1; - then discard the most significant 32 bits of the RES; - the next 64 bits of RES1 are the ciphering key, the operation returns this value. - the 32 least singnificant bits are not used.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_FirstDigi(bcddigits:HEXSTRING)
Result Type:	B_4
Comments:	
Description	
The input parameter bcddigits shall be a BCD string (subset of HEXSTRING), the resut is a BITSTRING[4] of a binary representation of one BCD digit. The function of the OC_FirstDigi is to return the first (most significant) digit of the input parameter `bcddigits`. for example: OC_FirstDigi('12345') = '0001'B, OC_FirstDigi('012345678') = '0000'B.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_FnInc(fn:FN; inc:INTEGER)
Result Type:	FN
Comments:	

Description

The function of OC_FnInc operation to increase frame number with `inc`. The frame number to be incremented is the input parameter `fn` in FN type and the increment `inc` is in INTEGER type, the incremented frame number is returned in FN type.

```
OC_FnInc(fn, inc)
FN     fn;
INTEGER  inc;
{
int     fmrin;
FN     frmout;

   frmin = 51 * ((fn.t3 - fn.t2) MOD 26) + fn.t3 + 1326 * fn.t1_ ;

   frmout.t1_ = ((frmin + inc) DIV 1326) MOD 32;
   frmout.t2   = (frmin + inc) MOD 26;
   frmout.t3   = (frmin + inc) MOD 51;
   return (frmout);
}
```

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_GetSCTimeStamp(timezone:TZONES)
Result Type:	TPSCTS
Comments:	TPSCTS is HEXSTRING[14]

Description

This Operation provides the hexstring containing the service center time stamp (SCTS) according to GSM 03.40, clauses 9.2.2.1 and 9.2.3.11. The TSO reads the current time of the test systems clock and transforms the time in conbination with the input parameter 'timezone' into a service center time stamp.

Example:

1996 April 18, 15:32:46, timezone=4

OC_GetSCTimeStamp returns 69408151236440

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_IncTmsi(tmsi:TMSI_V; inc:OCTETSTRING)
Result Type:	TMSI_V
Comments:	the `tmsi` is 4 OCTETs long

Description

OC_IncTmsi operation adds the two input parameters and returns the result. An overflow of addition is allowed.

For example :

OC_IncTmsi('33542140'O + '01'O) = '33542141'O;
OC_IncTmsi('21322140'O + '08'O) = '21322148'O.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_InRang(tx: INTEGER; maxret:INTEGER; m: INTEGER)
Result Type:	BOOLEAN
Comments:	
Description	
The operation returns TRUE if (m DIV (maxret * ((230+maxret -1)/maxret))) is inside the following interval: [0.8 - ((tx+1)/2) DIV tx; 1.2 - ((tx+1)/2) DIV tx] where / is integer division, DIV is float division, m, tx and maxret are input parameters of the operation.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_IntToOct(n:INTEGER; l: INTEGER)
Result Type:	OCTETSTRING
Comments:	
Description	
OC_IntToOct converts the INTEGER `n` into OCTETSTRING with length = `l`. for example: OC_IntToOct(14,1) = '0E'O; OC_IntToOct(18,1) = '12'O; OC_IntToOct(18,2) = '0012'O.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LeastBits(bstring:BITSTRING; lg:INTEGER)
Result Type:	BITSTRING
Comments:	
Description	
OC_LeastBits operation returns the `lg` least significant bits of the original `bstring`. for example: OC_LeastBits('110011000101010'B, 3) = '010'B, OC_LeastBits('110011000101010'B, 6) = '101010'B.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LengthOf(identity_field :MI)
Result Type:	INTEGER
Comments:	
Description	
OC_LengthOf operation returns the actual length of the `identity_field`, the unit of length is in OCTET.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LengthOfBCDN(bcdn :BCDN)
Result Type:	INTEGER
Comments:	
Description	
OC_LengthOfBCDN operation returns the actual length of an IE of type BCDN 'bcdn', the unit of length is in OCTET.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LengthOfComp(comp: Component_T)
Result Type:	LENGTH
Comments:	
Description	
OC_LengthOfComp operation returns the actual length of the Component_T `comp`, the unit of length is in OCTET.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LengthOfString(strg: IA5String)
Result Type:	INTEGER
Comments:	
Description	
OC_LengthOfString operation returns the actual length (number of characters) of the string `strg`.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LookupS(t: INTEGER; combined :BOOLEAN)
Result Type:	INTEGER
Comments:	the algorithm is derived from Table 3.1/GSM 04.08 for values of parameter S.

Description

This operation returns an INTEGER according the following algorithm:

```
OC_LookupS( t, comb)
INTEGER t;
BOOLEAN combined;
{
    switch (t)
        {  case 3:
           case 8:
           case 14:
           case 50:
              if (combined) then
                      return(41)
                 else   return(55);
              break;
           case 4:
           case 9:
           case 16:
              if (combined) then
                      return(52)
                 else  return(76);
              break;
           case 5:
           case 10:
           case 20:
              if (combined) then
                      return(58)
                 else  return(109);
              break;
           case 6:
           case 11:
           case 25:
              if (combined) then
                      return(86)
                 else  return(163);
              break;
           case 7:
           case 12:
           case 32:
              if (combined) then
                      return(115)
                 else  return(217);
              break;
     }
}
```

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_ModifyBcap(bcap:BCAP; list_indicator:INTEGER)
Result Type:	BCAP
Comments:	

Description

Using the following lists

modem_list = ('00011'B, '00111'B)
transparent_list = ('00'B, '10'B, '11'B)
non_transparent_list ('01'B, '10'B, '11'B)

OC_ModifyBcap operation returns a bearer capability IE modified from the input IE according the following rule:

- for `list_indicator` =C_modemt_ce:
  - copy the IE unchanged.

- for `list_indicator` = C_modemt_Tlist:
  - replace the 'ce' value with the list 'transparent_list'

- for `list_indicator` = C_modemt_NTlist:
  - replace the 'ce' value with the list 'non_transparent_list'

- for `list_indicator` =C_modemList_ce:
  - replace the 'modemt' value with the list 'modem_list'.

- for `list_indicator` = C_modemList_Tlist:
  - replace the 'modemt' value with the list 'modem_list',
    and replace the 'ce' value with the list 'transparent_list'

- for `list_indicator` = C_modemList_NTlist:
  - replace the 'modemt' value with the list 'modem_list',
    and replace the 'ce' value with the list 'non_transparent_list'

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_MostBits(bstring:BITSTRING; lg :INTEGER)
Result Type:	BITSTRING
Comments:	

Description

OC_MostBits operation returns the `lg` most significant bits of the original `bstring`.

for example:

OC_LeastBits('110011000101010'B, 3) = '110'B,
OC_LeastBits('110011000101010'B, 6) = '110011'B.

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OC_MsrReptChk(msrres: MSRR; index:INTEGER)
Result Type:	BOOLEAN
Comments:	parameter index is used for selecting different match pattern

Description

OC_MsrReptChk operation checks whether the received measurement report contains correct values:

1. when the `index` = 1:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00000'B, '001011'B}, {'01111'B, '001001'B},
   {'10101'B, '001011'B}, {'10110'B, '001101'B},
   {'11100'B, '001111'B}, {'11111'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

2. when the `index` = 2:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00000'B, '001011'B}, {'00100'B, '001001'B},
   {'00110'B, '001011'B}, {'00101'B, '001101'B},
   {'00010'B, '001111'B}, {'00111'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

3. when the `index` = 3:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00001'B, '001011'B}, {'00101'B, '001001'B},
   {'00111'B, '001011'B}, {'00110'B, '001101'B},
   {'00011'B, '001111'B}, {'01000'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

4. when the `index` = 4 :
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   }
   equals to
   set { {'00000'B, '001011'B}, {'00010'B, '001111'B},
   {'00001'B, '001111'B}, {'00011'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

5. when the `index` = 5 :
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   }
   equals to
   set { {'00000'B, '001011'B}, {'00010'B, '001111'B},
   {'00001'B, '001111'B}, {'00100'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

6. when the `index` = 6:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00000'B, '001011'B}, {'00011'B, '001001'B},
   {'00100'B, '001011'B}, {'00101'B, '001101'B},
   {'00110'B, '001111'B}, {'00111'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

7. when the `index` = 7:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00000'B, '001011'B}, {'00100'B, '001001'B},
   {'00110'B, '001011'B}, {'00101'B, '001101'B},
   {'00010'B, '001111'B}, {'00111'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

8. when the `index` = 8 :
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2}
   }
   equals to
   set { {'00000'B, '001011'B}, {'00010'B, '001101'B}
   }
   then the operation returns TRUE otherwise FALSE.

9. when the `index` = 9 :
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2}
   }
   equals to
   set { {'00110'B, '001011'B}, {'00111'B, '001101'B}
   }
   then the operation returns TRUE otherwise FALSE.

10. when the `index` = 10 :
    if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2}
    }
    equals to
    set { {'00011'B, '001011'B}, {'00101'B, '001101'B}
    }
    then the operation returns TRUE otherwise FALSE.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_OeBit(bcddigits:HEXSTRING)
Result Type:	BITSTRING
Comments:	The input parameter `bcddigits` is really BCD string (subset of HEXSTRING), the result is BITSTRING[1].
Description	
The function of the OC_OeBit is as the following: 1. it returns '1'B, if the length of the `bcddigits` is odd, 2. it returns '0'B, if the length of the `bcddigits` is even. for example: OC_OeBit('12583') = '1'B, OC_OeBit('87259957') ='0'B.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_OctToInt(ostr: OCTETSTRING)
Result Type:	INTEGER
Comments:	
Description	
OC_OctToInt converts the OCTETSTRING `ostr` into INTEGER.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_OctToInvokeIDType(o:OCTETSTRING)
Result Type:	InvokeIDType
Comments:	
Description	
OC_OctToInvokeIDType converts the OCTETSTRING `o` into InvokeIDType, with the MSB of 'o' representing the negative value -2exp(n-1), for example '80'O->'10000000'B-> -128. The rest of the bits can code positive values up to +2exp(n-2)-1, for example '4F'O->'01111111'B-> +127. for example: OC_OctToInvokeIDType('80'O) = -128; OC_OctToInvokeIDType('81'O) = -128+1 = -127; OC_OctToInvokeIDType('40'O) = +64. OC_OctToInvokeIDType('7F'O) = +127. OC_OctToInvokeIDType('C0'O) = -128+64 = -64.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_OtherDigi(bcddigits:HEXSTRING)
Result Type:	OCTETSTRING
Comments:	The input parameter` bcddigits` is really BCD string (subset of HEXSTRING), the result is an even BCD digits except that the next to last may either be 'F'H or a BCD digit.

Description

The function of the OC_OtherDigi is as the following:

1. If the number of the `bcddigits` is odd, the operation removes the most significant digit, and then reverses the order of each pair of digits;
2. If the number of the `bcddigits` is even, first the operation suffixes the `bcddigits` with 'F'H, then removes the most significant digit, and then reverses the order of each pair of digits.

for example:

OC_OtherDigi('12345') = '3254',
OC_OtherDigi('12345678') ='325476F8'.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_PosinSet(set: Components; comp: Component)
Result Type:	INTEGER
Comments:	

Description

The operation OC_PosinSet returns the position of component `comp` within the SET `set`.

for example:

if the set = { registerSSComponents, eraseSSComponents, activateSSComponents, deactivateSSComponents }

OC_PosinSet(set, registerSSComponents) = 0,
OC_PosinSet(set, activateSSComponents) = 2.

Detailed Comments:

Test Suite Operation Definition

Operation Name: OC_RachSlots(fn1:FN; fn2:FN; cmbn: BOOLEAN; mode:INTEGER)

Result Type: INTEGER

Comments:

Description

OC_RachSlots calculates the number of RACH slots between frame number `fn1` and `fn2` excluding the slots in `fn1` and `fn2`, then return the result of the calculation.

If mode=0, 'fn1' denotes the (first) frame number for sending Paging Request, while 'fn2' denotes the (first) frame number for the received Channel Request. If mode=1 'fn1' and 'fn2' denote the (first) frame number for the received two subsequent Channel Requests. Any other values for mode is not allowed.

Since the number of RACH slots depends on the RACH being combined or not with dedicated channels, the parameter 'cmbn' is needed: TRUE --> combined, FALSE--> not combined.

It is assumed that the distance between fn1 and fn2 is not more than one group of 42432 TDMA frames (modulo 42432 is used to calculate t1' of fn1 and fn2). This is equivalent to approximately 196 seconds. The 51 TDMA frames in a multiframe are numbered 0 to 50.

fn1' := fn1 mod 51

fn2' := fn2 mod 51

CASE 1: cmbn is FALSE ('not combined')

When 'not combined', all slots are RACH slots.

1.1 mode=0 (sending paging message at fn1)

The paging uses the numbered frames 6 - 9, 12 -19, 22-29, 32-39 and 42-49.

1.1.1 If fn1 is not in the paging TDMA frame mapping, it leads testing to a test system error.

IF (fn1' < 6) OR (9 < fn1' < 12) OR (19 < fn1' < 22) OR (29 < fn1' < 32) OR (39 < fn1' < 42) OR (fn1' >49),

A test system error! The calling test case shall re-run again.

OTHERWISE

1.1.2 The number of RACH slots is equal to the number of frames between fn1 + 3 and fn2. The fn1 is added by 3 because a page message occupies 4 slots. The fn1 indicates the first slot of the paging message.

IF (fn1 + 3) < fn2, OC_RachSlots := fn2 - (fn1 + 3) -1 = fn2 - fn1 - 4,

1.1.3 If fn1+3 is equal to or greater than fn2, then this is due to fn2 being in the next group of 42432 frames. In this case 42432 frames have to be added.

IF fn2 <= (fn1 + 3), OC_RachSlots := fn2 -fn1 - 3 + 42432 -1 = fn2 -fn1 + 42428

1.2 mode=1 (receiving channel request at fn1)

The number of RACH slots is equal to the number of frames between fn1 and fn2

1.2.1 IF fn1 < fn2, OC_RachSlots := fn2 - fn1 - 1,

1.2.2 IF fn2 <= fn1, OC_RachSlots := fn2 -fn1 + 42432 -1 = fn2 -fn1 + 42431

CASE 2: cmbn is TRUE ('combined')

When combined only the slots of the numbered frames 4, 5, 14 to 36, 45 and 46 in each multiframe are RACH slots, i.e.total 27 frames per multiframe.

2.1 mode=0 (sending paging message at fn1)

The paging uses the numbered frames 6 - 9 and 12 -19.

2.1.1 If fn1 is not in the paging TDMA frame mapping, it leads testing to a test system error. If fn2 is not in the RACH TDMA frame mapping, it leads testing to fail.

IF (fn2` < 4) OR (5 < fn2` < 14) OR (36 < fn2` < 45) OR (fn2' > 46), OC_RachSlots := -9999

IF (fn1' < 6) OR (9 < fn1' < 12) OR (fn1' >19), A test system error! The calling test case shall re-run again.

OTHERWISE

2.1.2 Calculation of the number 'c' multiframes between fn1 + 3 and fn2. If fn1 + 3 is equal to or greater than fn2, then this is due to fn2 being in the next group of 42432 frames. In this case 42432 frames have to be added. '/' shall be the integer division, i.e. the result is also integer. Fractions are discarded.

2.1.2.1 IF (fn1 + 3) < fn2, c := fn2 / 51 - (fn1 + 3) / 51

2.1.2.2 IF fn2 <= (fn1 + 3), c := fn2 / 51 - (fn1 + 3) / 51 + 42432 / 51 = fn2 / 51 - (fn1 + 3) / 51 + 832

2.1.3 Calculation of the number of frames 'a' to be subtracted according to the position of fn1' within the multiframe

IF (5 < fn1` < 9) , a := 2

IF (fn1` =9), a := 3
IF (11<fn1` <20), a := fn1'-8

2.2 mode=1 (receiving channel request at fn1)

2.2.1 If fn1 or fn2 are not in the RACH TDMA frame mapping, it leads testing to fail.
IF (fn2` < 4) OR (5 < fn2` < 14) OR (36 < fn2` < 45) OR (fn2' > 46)
OR (fn1' < 6) OR (9 < fn1' < 12) OR (fn1' >19), OC_RachSlots := -9999

2.2.2 Calculation of the number 'c' multiframes between fn1 and fn2
2.2.2.1 IF fn1 < fn2, c := fn2 / 51 - fn1 / 51
2.2.2.2 IF fn2 <= fn1 c := fn2 / 51 - fn1 / 51 + 42432 / 51 = fn2 / 51 - fn1/ 51 + 832

2.2.3 Calculation of the number of frames 'a' to be subtracted according to the position of fn1' within the multiframe
IF (3 < fn1' < 6:), a:= fn1' - 3
IF (13 < fn1' < 37), a:= fn1' - 11
IF (44 < fn1' < 47), a:= fn1' - 19

2.3 Calculation of the number of slots 'b' to be added according to the position of fn2' within the multiframe
IF (3 < fn2' < 6:), b:= fn2' - 4
IF (13 < fn2' < 37), b:= fn2' - 12
IF (44 < fn2' < 47), b:= fn2' - 20

2.4 Calculation of the number of RACH slots. There are 27 RACH slots in each multiframe.
OC_RachSlots := 27 * c + b - a

Detailed Comments: TC_26_2_1_1 uses mode=0. TC_26_2_1_2 uses mode=1.

Test Suite Operation Definition

Operation Name: OC_Random(n1:INTEGER; n2:INTEGER)
Result Type: INTEGER
Comments:

Description

This operation randomly returns one number from the following candidates:
'n1', 'n1'+1, ..., 'n2'

Detailed Comments:

Test Suite Operation Definition

Operation Name: OC_RcsdPresent(msg:MODIFY_PDU)
Result Type: BOOLEAN
Comments: To check if RCSD IE is present or not in Modify PDU

Description

IF RCSD IE is present in the PDU passed on input paremeter return TRUE ELSE return FALSE.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_SaveAndProc1(val:INTEGER; mode :INTEGER; cnt:INTEGER; cmbnd:BOOLEAN)
Result Type:	BOOLEAN
Comments:	This operation is used for storing and analysing the CHANNEL REQUEST message distribution.

Description

The function of this operation is defined as an pseudo C code:

```
#define SAVE 0;
#define PROC 1;

OC_SaveAndProc(val, mode,cnt)
INTEGER       val, mode,cnt;
BOOLEAN     combined;
{
  static INTEGER   buf[200];
   int i, j, n;
   if (mode == SAVE)
       { cnt = cnt mod 200;
         if (cmbnd == C_NotCombined) AND (val<(151 + 8)) then
                 {  buf[cnt] = val;
                   return (TRUE);
                  }
         if (cmbnd == C_Combined) AND (val<(81 + 8)) then
                 {  buf[cnt] = val;
                   return (TRUE);
                  }
         else return (FALSE);
        }
   if (mode == PROC)
        {
          for (i=0,  i<200, i++)
               {  n=0;
                  for (j=0, j<200, j++)
                      if (buf[j] == buf[i])
                          n = n+1;
                  if (n > 41) return (FALSE);
               }
            return(TRUE);
         }
}
```

Detailed Comments:	This test suite operation has two operation modes : - when the mode is SAVE and the val is less than 89 for combined or 159 for non-combined, it saves the value val into internal buffer and returns TRUE, otherwise returns FALSE. 200 values will be stored in the internal buffer when the operation is invoked 200 times. - When the mode is PROC, it analyses the values stored in the internal buffer, if no more than 41 of them are equal the operation returns TRUE, otherwise returns FALSE. (i.e. for all n, CARD{ k|f(k) = n } <= 41)

Test Suite Operation Definition	
Operation Name:	OC_SaveAndProc3(val:BITSTRING; mode :INTEGER; cnt:INTEGER)
Result Type:	BOOLEAN
Comments:	This operation is used for storing and analysing the random reference.

Description

The function of this operation is defined as an pseudo C code:

```
#define SAVE 0;
#define PROC 1;

OC_SaveAndProc(val, mode,cnt)
BITSTRING[8] val;
INTEGER       mode,cnt;
{
  static BITSTRING[8]   buf[7];
   int i, j, n=7;
   if (mode == SAVE)
       { cnt = cnt mod 7;
         buf[cnt] = val;
         return(TRUE);
        }
   if (mode == PROC)
        {
          for (i=0,  i<6, i++)
             { for (j=i+1, j<7, j++)
                  if (buf[j] == buf[i])
                    { n = n-1;
                      break;
                    }
             }
          if (n>=4) then
                  return(TRUE)
          else    return (FALSE);
         }

}
```

Detailed Comments:	The function of this test suite operation is : - it saves the value val into internal buffer when the mode = SAVE. 7 values will be stored in the internal buffer when the operation is invoked 7 times. - it compares the values stored in the internal buffer when the mode = PROC, if 4 or more than 4 of them are different the operation returns TRUE, otherwise returns FALSE.

Test Suite Operation Definition

Operation Name:	OC_SaveAndRetrv(val:RQR; mode, idx, ex: INTEGER)
Result Type:	RQR
Comments:	This operation is used for storing and retrieving the random reference.

Description

The function of this test suite operation is (in prose):

- it saves the value val into internal buffer buf[idx] when the mode = SAVE and ex = 0. Maximum 9 values can be stored. The idx is ranged between 1-9.
- it returns the value stored in the internal buffer buf[idx] when the mode = RETRV and ex = 0.
- it returns a value which is different from any value stored in the buf and also different from any values which have been returned by the consecutive invocation of this operation with ex <> 0, if ex <> 0. The idx is ranged between 10-13.

The function of this operation is defined as an pseudo C code:

```
#define SAVE 0;
#define RETRV 1;

OC_SaveAndRetrv(val, mode, idx, ex)
RQR             val;
INTEGER         mode, idx, ex;
{
  int i, j;
  static RQR   buf[14];
   if ((mode == SAVE) AND (ex == 0))
       {
         buf[idx] = val;
         buf[10].ra = '00000000'B;
         buf[10].fn = '00'O;
         buf[11].ra = '00000000'B;
         buf[11].fn = '00'O;
         buf[12].ra = '00000000'B;
         buf[12].fn = '00'O;
         buf[13].ra = '00000000'B;
         buf[13].fn = '00'O;
         return (buf[10];
        }
   if ((mode == RETRV) AND (ex == 0))
         return (buf[idx]);
   if (ex != 0)
         {
           buf[0].ra = buf[1].ra + '00000001'B;
           buf[0].fn = buf[1].fn;
           for (j=1, j <13, j++)
              {
                for (i=1, i<14, i++)
                  if( buf[0] == buf[i])
                      { buf[0].ra = buf[i].ra + '00000001'B;
                        break;
                       }
                if (i == 14)
                 {
                   buf[idx].ra = buf[0].ra:
                   return (buf[0]);
                 }
              }
          buf[idx].ra = buf[i].ra+'00000001'B;
          return (buf[idx]);
         }
}
```

Detailed Comments:	0. The pseudo C code gives one of the possible implementations for the OC description in prose. 1. '00'O should be understood as equivalence of Fn_01, a TTCN structured type constraint. 2. The OC can save maximum 13 RQR values. In SAVE mode, idx is ranged between 1-9.

	3. For ex<>0, there are maximum 13 scans. In each scan a new value is assigned to buf[0].ra. If there is a 'non-match' in one of the 13 scans the buf[0] is returned. In case of 'all-match' for the 13 scans, the pair of (buf[i].ra + '00000001' at the last scan, buf[idx].fn is returned (idx is ranged between 10 - 13).

Test Suite Operation Definition	
Operation Name:	OC_StartTime(frmn:FN; t, i:INTEGER)
Result Type:	STRT
Comments:	

Description

OC_StartTime operation generates the STARTING TIME IE according to the input parameters.

```
(* frmn is the current frame number,
   t is the "delay" in applying the new frequencies,
   i is the contents of the starting time IE. *)
OC_StartTime(frmn, t, i)
INTEGER  t, i;
FN          frmn;
{
int tmp;
STRT strt;

tmp = 51 * ((frmn.t3 - frmn.t2) MOD 26) + frmn.t3 + 1326 * frmn.t1_ ;
tmp = (tmp + t) MOD 42432;
strt.fn.t1_ = (tmp DIV 1326) MOD 32;
strt.fn.t2   = tmp MOD 26;
strt.fn.t3   = tmp MOD 51;
strt.iei = OMIT;
if (i == 1) strt.iei := '01111100'B  /* if i=1 the information element identifier shall be included \
```

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_SubOctet(src:OCTETSTRING; len :INTEGER)
Result Type:	OCTETSTRING
Comments:	

Description

OC_SubOctet(src, len) is the octetstring of length `len` starting from the leftmost position of the source octetstring `src`.

For example : OC_SubOctet('123456789ABCDEF'O, 4) = '12345678'O

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_SubchOfFacchh(subch: B_1; cell:CellID; inst:INTEGER)
Result Type:	LOGICCH
Comments:	subch is of type BITSTRING[1]

Description

OC_SubchOfFacchh operation returns a logic channel identifier for FACCHH subchannel indicated by the input parameters `subch`, `cell` and `inst`, where 'subch' is the subchannel number, `cell` is the cell identifier which the channel belongs to, `inst` is the instance of the channel.

for example:

OC_SubchOfFacchh('0'B, C_CellA, 1) = C_FACCHH0_A_1;
OC_SubchOfFacchh('1'B, C_CellA, 1) = C_FACCHH1_A_1;
OC_SubchOfFacchh('0'B, C_CellB, 1) = C_FACCHH0_B_1;
OC_SubchOfFacchh('1'B, C_CellB, 1) = C_FACCHH1_B_1.

pseudo C code definition for the operation as following :

```
OC_SubchOfFacchh(subch, cell, inst)
BITSTRING   subch;
CellID     cell;
INTEGER   inst;
{
   LOGICCH    logch;
   if (subch=='0'B && cell == C_CellA)
      switch (inst)
        {
          case 1 : logch = C_FACCHH0_A_1; break;
          case 2 : logch = C_FACCHH0_A_2; break;
          case 3 : logch = C_FACCHH0_A_3; break;
        }
  if (subch=='0'B && cell == C_CellB)
      switch (inst)
        {
          case 1 : logch = C_FACCHH0_B_1; break;
          case 2 : logch = C_FACCHH0_B_2; break;
          case 3 : logch = C_FACCHH0_B_3; break;
        }
  if (subch=='0'B && cell == C_CellC)
      switch (inst)
        {
          case 1 : logch = C_FACCHH0_C_1; break;
          case 2 : logch = C_FACCHH0_C_2; break;
          case 3 : logch = C_FACCHH0_C_3; break;
        }
   if (subch=='1'B && cell == C_CellA)
      switch (inst)
        {
          case 1 : logch = C_FACCHH1_A_1; break;
          case 2 : logch = C_FACCHH1_A_2; break;
          case 3 : logch = C_FACCHH1_A_3; break;
        }
  if (subch=='1'B && cell == C_CellB)
      switch (inst)
        {
          case 1 : logch = C_FACCHH1_B_1; break;
          case 2 : logch = C_FACCHH1_B_2; break;
          case 3 : logch = C_FACCHH1_B_3; break;
        }
  if (subch=='1'B && cell == C_CellC)
      switch (inst)
        {
          case 1 : logch = C_FACCHH1_C_1; break;
          case 2 : logch = C_FACCHH1_C_2; break;
```

```
            case 3 : logch = C_FACCHH1_C_3; break;
          }
  return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OC_SubchOfSacchh(subch: B_1; cell:CellID; inst:INTEGER)
Result Type:	LOGICCH
Comments:	subch is of type BITSTRING[1]

Description

OC_SubchOfSacchh operation returns a logic channel identifier for SACCHH subchannel indicated by the input parameters `subch`, `cell` and `inst`, where 'subch' is the subchannel number, `cell` is the cell identifier which the channel belongs to, `inst` is the instance of the channel.

for example:

OC_SubchOfSacchh('0'B, C_CellA, 1) = C_SACCHH0_A_1;
OC_SubchOfSacchh('1'B, C_CellA, 1) = C_SACCHH1_A_1;
OC_SubchOfSacchh('0'B, C_CellB, 1) = C_SACCHH0_B_1;
OC_SubchOfSacchh('1'B, C_CellB, 1) = C_SACCHH1_B_1.

pseudo C code definition for the operation as following :

```
OC_SubchOfSacchh(subch, cell, inst)
BITSTRING   subch;
CellID     cell;
INTEGER   inst;
{
   LOGICCH    logch;
   if (subch=='0'B && cell == C_CellA)
      switch (inst)
        {
          case 1 : logch = C_SACCHH0_A_1; break;
          case 2 : logch = C_SACCHH0_A_2; break;
          case 3 : logch = C_SACCHH0_A_3; break;
        }
  if (subch=='0'B && cell == C_CellB)
      switch (inst)
        {
          case 1 : logch = C_SACCHH0_B_1; break;
          case 2 : logch = C_SACCHH0_B_2; break;
          case 3 : logch = C_SACCHH0_B_3; break;
        }
  if (subch=='0'B && cell == C_CellC)
      switch (inst)
        {
          case 1 : logch = C_SACCHH0_C_1; break;
          case 2 : logch = C_SACCHH0_C_2; break;
          case 3 : logch = C_SACCHH0_C_3; break;
        }
   if (subch=='1'B && cell == C_CellA)
      switch (inst)
        {
          case 1 : logch = C_SACCHH1_A_1; break;
          case 2 : logch = C_SACCHH1_A_2; break;
          case 3 : logch = C_SACCHH1_A_3; break;
        }
  if (subch=='1'B && cell == C_CellB)
      switch (inst)
        {
          case 1 : logch = C_SACCHH1_B_1; break;
          case 2 : logch = C_SACCHH1_B_2; break;
          case 3 : logch = C_SACCHH1_B_3; break;
        }
  if (subch=='1'B && cell == C_CellC)
      switch (inst)
        {
          case 1 : logch = C_SACCHH1_C_1; break;
          case 2 : logch = C_SACCHH1_C_2; break;
```

```
          case 3 : logch = C_SACCHH1_C_3; break;
        }
  return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name: OC_SubchOfSacch4(subch: B_2; cell:CellID)
Result Type: LOGICCH
Comments:

Description

OC_SubchOfSacch4 operation returns a logic channel identifier for SACCHC4 subchannel indicated by the input parameters `subch ` and `cell`, where `subch` is the TDMA offset, `cell` is the cell identifier which the channel belongs to.

for example:

OC_SubchOfSacch4('00'B, C_CellA) = C_SACCHC40_A;
OC_SubchOfSacch4('01'B, C_CellA) = C_SACCHC41_A;
OC_SubchOfSacch4('10'B, C_CellA) = C_SACCHC42_A;
OC_SubchOfSacch4('11'B, C_CellA) = C_SACCHC43_A;
OC_SubchOfSacch4('00'B, C_CellB) = C_SACCHC40_B;
OC_SubchOfSacch4('01'B, C_CellB) = C_SACCHC41_B.

pseudo C code definition for the operation as following :

```
OC_SubchOfSacch4(subch, cell)
BITSTRING   subch;
CellID     cell;
{
   LOGICCH    logch;
   if (cell == C_CellA)
      switch (subch)
        {
          case '00'B : logch = C_SACCHC40_A; break;
          case '01'B : logch = C_SACCHC41_A; break;
          case '10'B : logch = C_SACCHC42_A; break;
          case '11'B : logch = C_SACCHC43_A; break;
        }
   if (cell == C_CellB)
      switch (subch)
        {
          case '00'B : logch = C_SACCHC40_B; break;
          case '01'B : logch = C_SACCHC41_B; break;
          case '10'B : logch = C_SACCHC42_B; break;
          case '11'B : logch = C_SACCHC43_B; break;
        }
   if (cell == C_CellC)
      switch (subch)
        {
          case '00'B : logch = C_SACCHC40_C; break;
          case '01'B : logch = C_SACCHC41_C; break;
          case '10'B : logch = C_SACCHC42_C; break;
          case '11'B : logch = C_SACCHC43_C; break;
        }
  return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_SubchOfSacch8(subch: B_3; cell:CellID; inst:INTEGER)
Result Type:	LOGICCH
Comments:	offset is of type BITSTRING[3]

Description

OC_SubchOfSacch8 operation returns a logic channel identifier for SACCHC8 subchannel indicated by the input parameters `subch`, `cell` and `inst`, where `subch` is the TDMA offset, `cell` is the cell identifier which the channel belongs to, `inst` is the instance of the channel.

for example:

OC_SubchOfSacch8('000'B, C_CellA, 1) = C_SACCHC80_A_1;
OC_SubchOfSacch8('001'B, C_CellA, 1) = C_SACCHC81_A_1;
OC_SubchOfSacch8('010'B, C_CellA, 1) = C_SACCHC82_A_1;
OC_SubchOfSacch8('011'B, C_CellA, 1) = C_SACCHC83_A_1;
OC_SubchOfSacch8('100'B, C_CellA, 1) = C_SACCHC84_A_1;
OC_SubchOfSacch8('101'B, C_CellA, 1) = C_SACCHC85_A_1;
OC_SubchOfSacch8('110'B, C_CellA, 1) = C_SACCHC86_A_1;
OC_SubchOfSacch8('111'B, C_CellA, 1) = C_SACCHC87_A_1;
OC_SubchOfSacch8('000'B, C_CellB, 1) = C_SACCHC80_B_1;
OC_SubchOfSacch8('001'B, C_CellB, 1) = C_SACCHC81_B_1;
OC_SubchOfSacch8('000'B, C_CellC, 2) = C_SACCHC80_C_2;
OC_SubchOfSacch8('001'B, C_CellC, 2) = C_SACCHC81_C_2.

pseudo C code definition for the operation as following :

```
OC_SubchOfSacch8(subch, cell, inst)
BITSTRING   subch;
CellID     cell;
INTEGER   inst;
{
  LOGICCH    logch;
  if (inst==1 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SACCHC80_A_1; break;
         case '001'B : logch = C_SACCHC81_A_1; break;
         case '010'B : logch = C_SACCHC82_A_1; break;
         case '011'B : logch = C_SACCHC83_A_1; break;
         case '100'B : logch = C_SACCHC84_A_1; break;
         case '101'B : logch = C_SACCHC85_A_1; break;
         case '110'B : logch = C_SACCHC86_A_1; break;
         case '111'B : logch = C_SACCHC87_A_1; break;
       }
  if (inst==2 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SACCHC80_A_2; break;
         case '001'B : logch = C_SACCHC81_A_2; break;
         case '010'B : logch = C_SACCHC82_A_2; break;
         case '011'B : logch = C_SACCHC83_A_2; break;
         case '100'B : logch = C_SACCHC84_A_2; break;
         case '101'B : logch = C_SACCHC85_A_2; break;
         case '110'B : logch = C_SACCHC86_A_2; break;
         case '111'B : logch = C_SACCHC87_A_2; break;
       }
  if (inst==3 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SACCHC80_A_3; break;
         case '001'B : logch = C_SACCHC81_A_3; break;
         case '010'B : logch = C_SACCHC82_A_3; break;
         case '011'B : logch = C_SACCHC83_A_3; break;
         case '100'B : logch = C_SACCHC84_A_3; break;
```

```
        case '101'B : logch = C_SACCHC85_A_3; break;
        case '110'B : logch = C_SACCHC86_A_3; break;
        case '111'B : logch = C_SACCHC87_A_3; break;
       }
if (inst==1 && cell == C_CellB)
   switch (subch)
     {
      case '000'B : logch = C_SACCHC80_B_1; break;
      case '001'B : logch = C_SACCHC81_B_1; break;
      case '010'B : logch = C_SACCHC82_B_1; break;
      case '011'B : logch = C_SACCHC83_B_1; break;
      case '100'B : logch = C_SACCHC84_B_1; break;
      case '101'B : logch = C_SACCHC85_B_1; break;
      case '110'B : logch = C_SACCHC86_B_1; break;
      case '111'B : logch = C_SACCHC87_B_1; break;
     }
if (inst==2 && cell == C_CellB)
   switch (subch)
     {
      case '000'B : logch = C_SACCHC80_B_2; break;
      case '001'B : logch = C_SACCHC81_B_2; break;
      case '010'B : logch = C_SACCHC82_B_2; break;
      case '011'B : logch = C_SACCHC83_B_2; break;
      case '100'B : logch = C_SACCHC84_B_2; break;
      case '101'B : logch = C_SACCHC85_B_2; break;
      case '110'B : logch = C_SACCHC86_B_2; break;
      case '111'B : logch = C_SACCHC87_B_2; break;
     }
if (inst==3 && cell == C_CellB)
   switch (subch)
     {
      case '000'B : logch = C_SACCHC80_B_3; break;
      case '001'B : logch = C_SACCHC81_B_3; break;
      case '010'B : logch = C_SACCHC82_B_3; break;
      case '011'B : logch = C_SACCHC83_B_3; break;
      case '100'B : logch = C_SACCHC84_B_3; break;
      case '101'B : logch = C_SACCHC85_B_3; break;
      case '110'B : logch = C_SACCHC86_B_3; break;
      case '111'B : logch = C_SACCHC87_B_3; break;
     }
if (inst==1 && cell == C_CellC)
   switch (subch)
     {
      case '000'B : logch = C_SACCHC80_C_1; break;
      case '001'B : logch = C_SACCHC81_C_1; break;
      case '010'B : logch = C_SACCHC82_C_1; break;
      case '011'B : logch = C_SACCHC83_C_1; break;
      case '100'B : logch = C_SACCHC84_C_1; break;
      case '101'B : logch = C_SACCHC85_C_1; break;
      case '110'B : logch = C_SACCHC86_C_1; break;
      case '111'B : logch = C_SACCHC87_C_1; break;
     }
if (inst==2 && cell == C_CellC)
   switch (subch)
     {
      case '000'B : logch = C_SACCHC80_C_2; break;
      case '001'B : logch = C_SACCHC81_C_2; break;
      case '010'B : logch = C_SACCHC82_C_2; break;
      case '011'B : logch = C_SACCHC83_C_2; break;
      case '100'B : logch = C_SACCHC84_C_2; break;
      case '101'B : logch = C_SACCHC85_C_2; break;
      case '110'B : logch = C_SACCHC86_C_2; break;
      case '111'B : logch = C_SACCHC87_C_2; break;
     }
if (inst==3 && cell == C_CellC)
```

```
      switch (subch)
        {
          case '000'B : logch = C_SACCHC80_C_3; break;
          case '001'B : logch = C_SACCHC81_C_3; break;
          case '010'B : logch = C_SACCHC82_C_3; break;
          case '011'B : logch = C_SACCHC83_C_3; break;
          case '100'B : logch = C_SACCHC84_C_3; break;
          case '101'B : logch = C_SACCHC85_C_3; break;
          case '110'B : logch = C_SACCHC86_C_3; break;
          case '111'B : logch = C_SACCHC87_C_3; break;
        }
return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name: OC_SubchOfSdcch4(subch: B_2; cell:CellID)
Result Type: LOGICCH
Comments:

Description

OC_SubchOfSdcch4 operation returns a logic channel identifier for SDCCH4 subchannel indicated by the input parameters `subch ` and `cell`, where `subch` is the TDMA offset, `cell` is the cell identifier which the channel belongs to.

for example:

OC_SubchOfSdcch4('00'B, C_CellA) = C_SDCCH40_A;
OC_SubchOfSdcch4('01'B, C_CellA) = C_SDCCH41_A;
OC_SubchOfSdcch4('10'B, C_CellA) = C_SDCCH42_A;
OC_SubchOfSdcch4('11'B, C_CellA) = C_SDCCH43_A;
OC_SubchOfSdcch4('00'B, C_CellB) = C_SDCCH40_B;
OC_SubchOfSdcch4('01'B, C_CellB) = C_SDCCH41_B.

pseudo C code definition for the operation as following :

```
OC_SubchOfSdcch4(subch, cell)
BITSTRING   subch;
CellID     cell;
{
   LOGICCH    logch;
   if (cell == C_CellA)
      switch (subch)
         {
           case '00'B : logch = C_SDCCH40_A; break;
           case '01'B : logch = C_SDCCH41_A; break;
           case '10'B : logch = C_SDCCH42_A; break;
           case '11'B : logch = C_SDCCH43_A; break;
         }
   if (cell == C_CellB)
      switch (subch)
         {
           case '00'B : logch = C_SDCCH40_B; break;
           case '01'B : logch = C_SDCCH41_B; break;
           case '10'B : logch = C_SDCCH42_B; break;
           case '11'B : logch = C_SDCCH43_B; break;
         }
   if (cell == C_CellC)
      switch (subch)
         {
           case '00'B : logch = C_SDCCH40_C; break;
           case '01'B : logch = C_SDCCH41_C; break;
           case '10'B : logch = C_SDCCH42_C; break;
           case '11'B : logch = C_SDCCH43_C; break;
         }
  return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_SubchOfSdcch8(subch: B_3; cell:CellID; inst:INTEGER)
Result Type:	LOGICCH
Comments:	offset is of type BITSTRING[3]

Description

OC_SubchOfSdcch8 operation returns a logic channel identifier for SDCCH8 subchannel indicated by the input parameters `subch`, `cell` and `inst`, where `subch` is the TDMA offset, `cell` is the cell identifier which the channel belongs to, `inst` is the instance of the channel.

for example:

OC_SubchOfSdcch8('000'B, C_CellA, 1) = C_SDCCH80_A_1;
OC_SubchOfSdcch8('001'B, C_CellA, 1) = C_SDCCH81_A_1;
OC_SubchOfSdcch8('010'B, C_CellA, 1) = C_SDCCH82_A_1;
OC_SubchOfSdcch8('011'B, C_CellA, 1) = C_SDCCH83_A_1;
OC_SubchOfSdcch8('100'B, C_CellA, 1) = C_SDCCH84_A_1;
OC_SubchOfSdcch8('101'B, C_CellA, 1) = C_SDCCH85_A_1;
OC_SubchOfSdcch8('110'B, C_CellA, 1) = C_SDCCH86_A_1;
OC_SubchOfSdcch8('111'B, C_CellA, 1) = C_SDCCH87_A_1;
OC_SubchOfSdcch8('000'B, C_CellB, 1) = C_SDCCH80_B_1;
OC_SubchOfSdcch8('001'B, C_CellB, 1) = C_SDCCH81_B_1;
OC_SubchOfSdcch8('000'B, C_CellC, 2) = C_SDCCH80_C_2;
OC_SubchOfSdcch8('001'B, C_CellC, 2) = C_SDCCH81_C_2.

pseudo C code definition for the operation as following :

```
OC_SubchOfSdcch8(subch, cell, inst)
BITSTRING   subch;
CellID     cell;
INTEGER   inst;
{
  LOGICCH    logch;
  if (inst==1 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SDCCH80_A_1; break;
         case '001'B : logch = C_SDCCH81_A_1; break;
         case '010'B : logch = C_SDCCH82_A_1; break;
         case '011'B : logch = C_SDCCH83_A_1; break;
         case '100'B : logch = C_SDCCH84_A_1; break;
         case '101'B : logch = C_SDCCH85_A_1; break;
         case '110'B : logch = C_SDCCH86_A_1; break;
         case '111'B : logch = C_SDCCH87_A_1; break;
       }
  if (inst==2 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SDCCH80_A_2; break;
         case '001'B : logch = C_SDCCH81_A_2; break;
         case '010'B : logch = C_SDCCH82_A_2; break;
         case '011'B : logch = C_SDCCH83_A_2; break;
         case '100'B : logch = C_SDCCH84_A_2; break;
         case '101'B : logch = C_SDCCH85_A_2; break;
         case '110'B : logch = C_SDCCH86_A_2; break;
         case '111'B : logch = C_SDCCH87_A_2; break;
       }
  if (inst==3 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SDCCH80_A_3; break;
         case '001'B : logch = C_SDCCH81_A_3; break;
         case '010'B : logch = C_SDCCH82_A_3; break;
         case '011'B : logch = C_SDCCH83_A_3; break;
         case '100'B : logch = C_SDCCH84_A_3; break;
```

```
        case '101'B : logch = C_SDCCH85_A_3; break;
        case '110'B : logch = C_SDCCH86_A_3; break;
        case '111'B : logch = C_SDCCH87_A_3; break;
       }
if (inst==1 && cell == C_CellB)
   switch (subch)
     {
      case '000'B : logch = C_SDCCH80_B_1; break;
      case '001'B : logch = C_SDCCH81_B_1; break;
      case '010'B : logch = C_SDCCH82_B_1; break;
      case '011'B : logch = C_SDCCH83_B_1; break;
      case '100'B : logch = C_SDCCH84_B_1; break;
      case '101'B : logch = C_SDCCH85_B_1; break;
      case '110'B : logch = C_SDCCH86_B_1; break;
      case '111'B : logch = C_SDCCH87_B_1; break;
     }
if (inst==2 && cell == C_CellB)
   switch (subch)
     {
      case '000'B : logch = C_SDCCH80_B_2; break;
      case '001'B : logch = C_SDCCH81_B_2; break;
      case '010'B : logch = C_SDCCH82_B_2; break;
      case '011'B : logch = C_SDCCH83_B_2; break;
      case '100'B : logch = C_SDCCH84_B_2; break;
      case '101'B : logch = C_SDCCH85_B_2; break;
      case '110'B : logch = C_SDCCH86_B_2; break;
      case '111'B : logch = C_SDCCH87_B_2; break;
     }
if (inst==3 && cell == C_CellB)
   switch (subch)
     {
      case '000'B : logch = C_SDCCH80_B_3; break;
      case '001'B : logch = C_SDCCH81_B_3; break;
      case '010'B : logch = C_SDCCH82_B_3; break;
      case '011'B : logch = C_SDCCH83_B_3; break;
      case '100'B : logch = C_SDCCH84_B_3; break;
      case '101'B : logch = C_SDCCH85_B_3; break;
      case '110'B : logch = C_SDCCH86_B_3; break;
      case '111'B : logch = C_SDCCH87_B_3; break;
     }
if (inst==1 && cell == C_CellC)
   switch (subch)
     {
      case '000'B : logch = C_SDCCH80_C_1; break;
      case '001'B : logch = C_SDCCH81_C_1; break;
      case '010'B : logch = C_SDCCH82_C_1; break;
      case '011'B : logch = C_SDCCH83_C_1; break;
      case '100'B : logch = C_SDCCH84_C_1; break;
      case '101'B : logch = C_SDCCH85_C_1; break;
      case '110'B : logch = C_SDCCH86_C_1; break;
      case '111'B : logch = C_SDCCH87_C_1; break;
     }
if (inst==2 && cell == C_CellC)
   switch (subch)
     {
      case '000'B : logch = C_SDCCH80_C_2; break;
      case '001'B : logch = C_SDCCH81_C_2; break;
      case '010'B : logch = C_SDCCH82_C_2; break;
      case '011'B : logch = C_SDCCH83_C_2; break;
      case '100'B : logch = C_SDCCH84_C_2; break;
      case '101'B : logch = C_SDCCH85_C_2; break;
      case '110'B : logch = C_SDCCH86_C_2; break;
      case '111'B : logch = C_SDCCH87_C_2; break;
     }
if (inst==3 && cell == C_CellC)
```

```
    switch (subch)
      {
        case '000'B : logch = C_SDCCH80_C_3; break;
        case '001'B : logch = C_SDCCH81_C_3; break;
        case '010'B : logch = C_SDCCH82_C_3; break;
        case '011'B : logch = C_SDCCH83_C_3; break;
        case '100'B : logch = C_SDCCH84_C_3; break;
        case '101'B : logch = C_SDCCH85_C_3; break;
        case '110'B : logch = C_SDCCH86_C_3; break;
        case '111'B : logch = C_SDCCH87_C_3; break;
      }
return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OM_2Msgs(ch:LOGICCH; pgg:PGG; mode:SENDINGMODE)
Result Type:	BOOLEAN
Comments:	

Description

OM_2Msgs puts the layer 2 emulator into a special operation mode then returns. In this special mode the layer 2 emulator sends the next two consecutive messages in the following way:

- send the first message on the paging subchannel indicated by the `pgg` and 'ch';

- if the `mode` = C_NxtButOne, send the second message in the next but one paging sub block;
- if the `mode` = C_FmrAGB, send the second message in a former access grant block;
- if the `mode` = C_BfReOcc, send the second message before the MS's original paging subchannel re-occurs but later than the next paging block of that CCCH (paging block not belong to the MS);
- if the `mode` = C_NxtButOneNxt, nothing is sent in the next but one paging sub block, then send the second message in the next paging subblock of the MS's paging subchannel.

Detailed Comments:

Test Suite Operation Definition

Operation Name: OM_Assoc(lgch1:LOGICCH; sublgch1:LOGICCH; sublgch2:LOGICCH; sublgch3:LOGICCH; sublgch4:LOGICCH; sublgch5:LOGICCH; sublgch6:LOGICCH; sublgch7: LOGICCH; sublgch8:LOGICCH; lgch2:LOGICCH; sublgch9:LOGICCH; sublgch10:LOGICCH; sublgch11:LOGICCH; sublgch12:LOGICCH; sublgch13:LOGICCH; sublgch14:LOGICCH; sublgch15: LOGICCH; sublgch16:LOGICCH)

Result Type: BOOLEAN

Comments:

Description

OM_Assoc operation associates the sub logic channel identifiers `sublgch1` to `sublgch8` and `sublgch9` to `sublgch16` with the generic "parent" channel identifier ` lgch1` and `lgch2` respectively, therefore the subchannel identifiers can refer to the corresponding channels configured by OM_ChConf operation.

The `lgch1` can be an identifier for SDCCH4, SDCCH8, FACCHH and the `lgch2` can be SACCHC4, SACCHC8, and SACCHH.

"dummy" in the actual parameter list means the corresponding parameter is not used.

Naming convention for logical channel constants:

C_<channel type>[<rate>][<subchannel>][_<cell>][_<instance>]

e.g.: C_FACCH H 0 _C _3 => C_FACCHH0_C_3

channel type: FCCH, SCH, BCCH, CBCH, PCH, RACH, FACCH, SACCH, SDCCH, DCCH

rate: F(ull), H(alf), 4 (fourths, C4 for SACCHC4), 8 (eighths, C8 for SACCHC8).

subchannel: H: 0, 1
(C)4: 0, ..., 3
(C)8: 0, ..., 7

cell: A, ..., H

instance: 1, ..., 3 When several channels of the same type can exist in one cell simultaneously (in different time slots and / or on different frequencies) they are identified by their instance. The parameters of ChConf are used to specify the frequency and slot of this instance of the channel type.

Not all possible combinations of this naming syntax have been used. According to the elements that are used / omitted the name can have a special meaning.

Examples:	Comments:
C_FCCH_A	FCCH of cell A.
C_SCH_A	SCH of cell A.
C_BCCH_A_1	First instance of a BCCH in cell A.
C_CBCH_A	CBCH of cell A.
C_PCH_A_1	First instance of a PCH in cell A (there is one instance of the PCH per BCCH in the cell).
C_AGCH_A_1	First instance of a AGCH in cell A (there is one instance of the AGCH per BCCH in the cell).
C_RACH_A_1	First instance of a RACH in cell A.
C_FACCHF_A_1	First instance of a full rate FACCH in cell A.
C_FACCHH_A_1	First instance of a half rate FACCH in cell A (all subchannels).
C_FACCHH0_A_1	Subchannel 0 of the first instance of a half rate FACCH in cell A.
C_SACCHF_A_1	First instance of a full rate SACCH in cell A.
C_SACCHH_A_1	First instance of a half rate SACCH in cell A (all subchannels).
C_SACCHH1_A_1	Subchannel 1 of the first instance of a half rate SACCH in cell A.
C_SACCHC4_A	SACCH/4 of cell A (all subchannels).
C_SACCHC43_A	Subchannel 3 of the SACCH/4 of cell A.
C_SACCHC8_A_1	First instance of a SACCH/8 in cell A (all subchannels).
C_SACCHC87_A_1	Subchannel 7 of the first instance of a SACCH/8 in cell A.
C_SDCCH4_A	SDCCH/4 of cell A (all subchannels).
C_SDCCH42_A	Subchannel 2 of the SDCCH/4 of cell A.
C_SDCCH87_A_1	First instance of a SDCCH/8 in cell A (all subchannels).
C_SDCCH87_A_1	Subchannel 7 of the first instance of a SDCCH/8 in cell A.

This list covers all types of combinations used in the TTCN for the time being.

When the meaning refers to 'all' subchannels of a channel (eg. C_FACCHH_A_1), it means that all active subchannels of this channel activated previously by OM_ChConf, are associated with a logical channel identifier in the OM_Assoc.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OM_BmInfo(ch:LOGICCH; mode:B_8)
Result Type:	BOOLEAN
Comments:	mode is of type BITSTRING[8], imode is a part of the channel mode IE

Description

OM_BmInfo operation checks whether the MS starts transmiting Bm channel information according to the new channel mode `mode'. The operation returns TRUE if the result of the check is correct otherwise the TSO returns FALSE.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OM_ChangeRFOf2Cells(cellid1:CellID; bspwr1:INTEGER; cellid2:CellID; bspwr2:INTEGER)
Result Type:	BOOLEAN
Comments:	

Description

This TSO is used to change the RF level of the cell 'cellid1' to the RF level 'bspwr1' and the RF level of the cell 'cellid2' to the RF level 'bspwr2' [dBuVemf], then the TSO returns TRUE.

The main use of this TSO is to force the Mobile Station(MS) to select a new cell according to the GSM 05.08 and GSM 03.22.

For the MS to select the new cell then the path loss criterion parameter (C1) of the old will have to be less than zero (C1 < 0).
This criterion is defined as:

C1 = (A - max(B, 0))

when

A - Received Level Average - (RXLEV_ACCESS_MIN)
B - MS_TXPWR_MAX_CCH - P {non DCS 1800}
MS_TXPWR_MAX_CCH + POWEROFFSET - P {DCS 1800}

RXLEV_ACCESS_MIN - Minimum received level at the MS required for access to the system.
MS_TXPWR_MAX_CCH - Maximum TX power level an MS may use when accessing the system until otherwise commanded.
POWER OFFSET - The power offset to be used in conjunction with the MS TXPWR MAX CCH
parameter by the class 3 DCS 1 800 MS.
P - Maximum RF output power of the MS.

All values are expressed in dBm.

In most test cases MS_TXPWR_MAX_CHH and the POWER_OFFSET will be set at their minimum values thus C1 = A (B always being negative). Suitable levels of cells will be either 63 dBuV(C_E_default) ,53 dBuV(C_E_neighbourdefault) or 45 dBuV(C_E_suitable). Assuming that the MS uses a RXLEV_ACCESS_MIN between -110 and -115 dbM then to ensure that C1 < 0 the test system's output needs to be set to ensure the level input detected by the MS is less than -123 dBm or lower (-115 plus an 8dB for discrimination between C1 or C2 values and 0, GSM 11.10-1 section 20). This value of -123 dBm corresponds to -10 dBuV (C_E_notsuitable)

Detailed Comments:

Test Suite Operation Definition

Operation Name: OM_ChConf(bspwr: INTEGER; mspwr:INTEGER; acttype:BITSTRING; chmod:CHMOD; ta:TA; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; td:INTEGER; fn:INTEGER; babr, cch_con, bpm:B_3; pgfil:PG1_RQ_PDU; dtxu:BITSTRING; dtxd:BITSTRING; cell:CellID; lgch1:LOGICCH; lgch2:LOGICCH; lgch3:LOGICCH; lgch4:LOGICCH; lgch5:LOGICCH; lgch6:LOGICCH; lgch7:LOGICCH; lgch8:LOGICCH; lgch9:LOGICCH)

Result Type: BOOLEAN

Comments:

Description

The operation sets the configuration of a basic physical channel according to the input parameters and map logic channel combination onto it:

- bspwr: base station power level in dBuv;
- mspwr: mobile station power level in "power control level";
- acttype: type of activation:
  - '000'B activation for intra-cell channel change ---- immediate assignment;
  - '001'B activation for intra-cell channel change ---- assignment;
  - '010'B activation for inter-cell channel change ---- asynchronous handover;
  - '011'B activation for inter-cell channel change ---- synchronous handover;
  - '100'B activation for additional assignment;
    NOTE: the types defined above are all the same, no real difference
  - '101'B activation for receiving only;
- chmod: channel mode;
- ta: timing advance ---- receive timing ta bit periods in advance of the normal receive timing if the channel is a dedicated channel ;
- slot: time slot;
- tsc: Training sequence code for normal bursts;
- rf: ARFCN or hopping parameters;
- chcmbn: logic channel combination mapping to the physical channel;
- td: timing difference between the cell and conceptual timing reference ---- to make 2 cells with different timebase;
- fn: initial frame number offset to the timing base counter ---- to make 2 cells with different frame numbers;
- babr number of blocks reserved for access grant
- cch-con CCCH/SDCCHs configuration
- bpm multiframe period for transmission of PAGING REQUEST
- pgfil: paging filling contents;
- dtxu: mobile station discontinuous transmission:
  - '0'B mobile station discontinuous transmission is not applied;
  - '1'B mobile station discontinuous transmission is applied;
- dtxd: base station discontinuous transmission:
  - '0'B base station discontinuous transmission is not applied;
  - '1'B base station discontinuous transmission is applied;
- cell: cell identifier;
- lgch1: logic channel identifier for the channel FCCH;
- lgch2: logic channel identifier for the channel SCH;
- lgch3: logic channel identifier for the channel BCCH;
- lgch4: logic channel identifier for the channel PCH;
- lgch5: logic channel identifier for the channel AGCH;
- lgch6: logic channel identifier for the channel RACH;
- lgch7: logic channel identifier for the channel FACCH or SDCCH;
- lgch8: logic channel identifier for the channel SACCH;
- lgch9: logic channel identifier for the channel CBCH;

For parameters lgch1 to lgch9 "dummy" in the actual parameter list means that the corresponding parameters are not used.

In the test cases the logic channel identifiers lgch1 to lgch9 are used to refer the logic channels configured by the operation. There are only generic identifiers for SDCCH4 or SDCCH8 or FACCHH channel and SACCHC4 or SACCHC8 or SACCHH, identifiers for subchannels of SDCCH4 or SDCCH8 or FACCHH and SACCHC4 or SACCHC8 or SACCHH are linked to the generic identifiers by OM_Assoc operation and in turn refer to the sub logic channels configured by the operation.

Channel re-use whithiin OM_ChConf:

When an OM_ChConf calls is made then the following rules can be applied for deciding whether or not to re-use existing physical channel.

Decode the 'chcmbn' (channel combination) parameter to determine the new channel type.

If new channel = "C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4" (combined) or

if new channel = "C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4" (combined + SMS) then re-use any existing control channel configuration with same CellID.

If new channel = "C_FCCH_SCH_BCCH_CCCH" (non-combined) then use any combined, combined + SMS or non-combined control channel configuration with the same CellID. If new channel = "C_BCCH_CCCH_2", "C_BCCH_CCCH_3" or "C_BCCH_CCCH_4" (ccch-group) then re-use any existing ccch-group control channel configuration with the same CellID and the slot, or re-use any existing combined, combined + SMS with the same CellID. If new channel = "C_TCHF_ACCHF_1", "C_TCHF_ACCHF_2", "C_TCHH_ACCHH_1", "C_TCHH_ACCHH_2" (traffic) or "C_SDCCH8_SACCHC8_1", "C_SDCCH8_SACCHC8_2", "C_SDCCH8_SACCHC8_3" (dedicated) then re-use any existing traffic or dedicated channel with the same channel description.
Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OM_ChMdModi(ch:LOGICCH; chmod: CHMOD)
Result Type:	BOOLEAN
Comments:	
Description	
OM_ChMdModi operation passes channel mode IE for channel `ch` to lower layer emulator and requests the emulator to set the `ch` to the mode `chmod`. After the mode changed, it returns.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_CphMdChg(ch:LOGICCH; cphmod: CPHMS; key:BITSTRING)
Result Type:	BOOLEAN
Comments:	
Description	
OM_CphMdChg operation passes ciphering parameters for channel `ch` to lower layer and sets the lower layer emulator into a special operation mode in which lower layer starts the 3-step ciphering mode setting sequence when the next L3 message on the channel `ch` arrives.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_CphMd(ch:LOGICCH; cphmod: CPHMS; key:BITSTRING)
Result Type:	BOOLEAN
Comments:	
Description	
OM_CphMd operation passes ciphering parameters for channel `ch` to lower layer and sets the channel `ch` in the ciphering mode `cphmod`. This operation puts the channel `ch` into specified ciphering mode `cphmod` immediately without the 3-step ciphering mode setting sequence.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_ComingFn(ch:LOGICCH)
Result Type:	FN
Comments:	
Description	
OM_ComingFn operation returns the frame number (FN modulo 42432) which is about 5 seconds later than current frame number and is able to carry L3 message on the channel `ch`.	
Detailed Comments:	The delay of about 5 seconds ensures that there is still enough time left for higher layer controller to prepare next TTCN send event after the OM_ComingFn returns. the exact delay value is up to the implementor.

Test Suite Operation Definition	
Operation Name:	OM_Deactivate(ch1, ch2:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
OM_Deactivate operation deactivates the logical channels `ch1` and 'ch2', then returns TRUE. The ch1 can be a DCCH or a TCH, the ch2 is the associated SACCH. Deactivation of an active SDCCH channel can be done by stopping transmission; this means that for the bursts on the BCCH carriers, dummy bursts shall be transmitted, for other bursts nothing shall be transmitted. Alternatively the deactivation of a SDCCH can be done by transmitting bad blocks on the SACCH with the parity bits set incorrectly. This will lead to a radio link failure in the MS. Deactivation of a TCH channel can be done by stopping transmission if it is non-BCCH. Deactivation of a SACCH_T can be done by transmitting bad blocks on the SACCH_T with the parity bits set incorrectly. The wrong channel coding on the SACCH will lead to a radio link failure in the MS after several seconds depending on the system information messages..	
Detailed Comments:	The TSO is called in TC_26_7_3_2, TC_26_7_4_3_2, TC_26_7_4_3_3, TC_26_7_4_3_4 on a SDCCH/4/SACCH/C4 channel in downlink, TC_26_2_4_1, TC_26_2_4_2, TC_26_8_2_1, TC_26_8_2_2, TC_26_8_2_3 on TCH/SACCH_T channel in both directions.

Test Suite Operation Definition	
Operation Name:	OM_FHCHK(ch:LOGICCH; ca:CCHD; ma:MA; cd:CHD; t :INTEGER; fn:FN)
Result Type:	BOOLEAN
Comments:	
Description	
After the SS has been configured to start / change frequency hopping at a particular start time, this TSO instructs the SS to check whether the MS correctly uses the hopping frequencies on and after the expected start time according to the frequency parameters. If `t` = 0 or 32024 <= `t` <= 42431 the operation checks that the new frequencies starts "without delay". If 0 < `t` <= 31000 the operation checks that the new frequencies starts "on" the frame number of (X + t) mod 42432, where x = 51 * ((fn.t3 - fn.t2) MOD 26) + fn.t3 + 1326 * fn.t1. If 31000 < `t` < 32024, or `t` < 0 or `t` > 42431 then the test has been configured incorrectly and should be re-run with a permitted value for `t`. This check is performed at the uplink RF burst level by sampling for a period of 3 seconds from the start time and should therefore be called immediately after the SS has been configured to start / change hopping. The operation returns TRUE if the MS behaviour is correct, otherwise it returns FALSE. Note: before this TSO is called, a layer 3 message must be sent to the MS to command the MS to start / change hopping using the same frequency hopping parameters.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_FreeResource(ch :LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
OM_FreeResource operation instructs the lower layer emulator to release the resources which have been allocated to the logical channel "ch". After the resources are released the operation returns TRUE.	
Detailed Comments:	In some test cases (most of them are hanhover test cases) diffirent channels are needed while the test goes on. the purpose of the operation is to reuse lower layer resources (e.g. tranceivers), which have been allocated to the previous channel, for configing another channel required in the next testing step.

Test Suite Operation Definition	
Operation Name:	OM_FreqDef(st:STRT; ma:MA; ch :LOGICCH; chd: CHD; ca:CCHD; frmn: FN)
Result Type:	BOOLEAN
Comments:	
Description	
OM_FreqDef operation passes the frequency hopping parameters for channel `ch` to lower layer emulator. The emulator starts using these frequency parameters at the frame number indicated by starting time `st`. The parameter 'frmn' indicates a reference frame number at which the first burst of the message, FREQUENCY REDEFINITION, ASSIGNMENT COMMAND or HANDOVER COMMAND, containing starting time is sent in the concerned cell. If 0 <= (st - frmn) mod 42432 <= 31623, the indicated time is the next time when frame number mod 42432 is equal to st. If 32024 <= (st - frmn) mod 42432 <= 42431, the indicated time has already elapsed. After completely sending out the message containing the parameter starting time the tester starts frequency hopping.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_GetHoaccPara(ch:LOGICCH)
Result Type:	HOACC_PARA
Comments:	
Description	
OM_GetHoaccPara instructs the lower layer emulator to return the total number of the handover access bursts received on channel `ch`, the observed minimum and maximum power levels in dBm modulo 256 and the observed minimum and maximum timing advance in bits modulo 256.The lower layer emulator starts the observation of handover access bursts as soon as sending out a HANDOVER COMMAND message. The TSO returns an accumulative result of the observation when it is called.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_GetL1Hd(ch:LOGICCH)
Result Type:	L1HD
Comments:	
Description	
OM_GetL1Hd instructs the L 2 emulator to read the L 1 header of the next SACCH frame received on channel `ch` and return this value.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_LowerLayerFail(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
This operation is to force the lower layer failure, then it returns. The lower layer failure can be any one of layer 1 failures or layer 2 failures. - layer 1 failure, see GSM 05.08; - layer 2 failure, see GSM 04.06.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_NoL2Ack(i:INTEGER; ch: LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
When the parameter i>0 the layer 2 emulator is put into a special operation mode then the TSO returns. In this special mode the layer 2 emulator shall not aknowledge the 'i'th occurence of the L2 I frame which has the more data bit "M" = 0 on the channel 'ch'. The layer 2 emulator resumes normal operation when - either the TSO is called again with i = 0 on the same 'ch', - or a data link SABM message is received on the 'ch'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_NotAckSetup(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
This operation puts the layer 2 emulator into a special operation mode then returns. In this special mode the layer 2 emulator will not acknowledge the last L 2 frame which carries the SETUP message, and the layer 2 emulator resumes normal operation after this L 2 frame.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_NoUAforDISC(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
OM_NoUAforDISC and OM_ResumUAforDISC are a pair of TSOs. OM_NoUAforDISC forces the L2 emulator of the tester to enter a specific mode. OM_ResumUAforDISC resumes the L2 emulator to a normal L2 operational mode. In the specific mode on receiving a DISC on the channel 'ch', the L2 emulator - does not respond with a UA, - remains in the multiple-frame established state, - indicates that the DISC has been received (by means of subsequent DL_RELEASE_INDICATION ASPs). Then the TSO returns TRUE.	
Detailed Comments:	The TSO is called in TC_26_6_12_2 and TC_26_6_12_4.

Test Suite Operation Definition	
Operation Name:	OM_PgFill(cellid:CellID; msg:PG1_RQ_PDU)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
OM_PgFill operation sets up the contents of paging filling frame to be sent on all paging subchannels continuously. If the contents of paging filling frame in cell `cellid` have not been set up, the operation sets the contents to `msg`. If paging filling frame in cell `cellid` has been set up, the operation changes the paging filling message to `msg`. After the required action has been taken it returns.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_Reactivate(ch1, ch2:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
The TSO OM_Reactivate reactivates the logical channels `ch1` and 'ch2' which were previously deactivated by the OM_Deactivate. Then the TSO returns TRUE. The ch1 is an SDCCH, the ch2 is the associated SACCH. Reactivation of deactivated SDCCH/SACCH channels means to restart normal transmissions both on the SDCCH and the associated SACCH as specified in the ETS 300 574 (GSM 05.02).	
Detailed Comments:	The TSO is called in TC_26_7_3_2, TC_26_7_4_3_2, TC_26_7_4_3_3, TC_26_7_4_3_4 on the SDCCH/4 / SACCH/C4 channels.

Test Suite Operation Definition	
Operation Name:	OM_ReturnFn(ch:LOGICCH)
Result Type:	FN
Comments:	
Description	
OM_ReturnFn operation returns the frame number (FN modulo 42432) on which the last L3 message was sent on logic channel `ch`.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_ResumUAforDISC(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
OM_NoUAforDISC and OM_ResumUAforDISC are a pair of TSOs. OM_NoUAforDISC forces the L2 emulator to enter a specific mode. OM_ResumUAforDISC resumes the L2 emulator to a normal L2 operational mode. OM_ResumeUAforDISC cancels the effect of OM_NoUAforDISC, returning L2 on the channel 'ch' to normal operation in the multiple-frame established state. Then the TSO returns TRUE.	
Detailed Comments:	The TSO is called in TC_26_6_12_2 and TC_26_6_12_4.

Test Suite Operation Definition	
Operation Name:	OM_SendSMSCBWhilePaging(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The TSO makes the lower layers send the second SMSCB message at the same time as the MS is paged. This shall be achieved by paging the MS immediately after the first block of the CB message has been sent. The SS shall ensure that the page is transmitted on the radio interface prior to the transmission of the 4th block of the CB message.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_SendNextOn(ch:LOGICCH; fn:FN)
Result Type:	BOOLEAN
Comments:	
Description	
OM_SendNextOn operation sets the lower layer emulator into a special operation mode then returns. In the special mode the lower layer emulator sends the next L3 message on the indicated frame number `fn` on the 'ch' channel. After the next L3 message is sent, the lower layer resumes normal operation automatically.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_SendNextOnHO(ch:LOGICCH; cnt:INTEGER)
Result Type:	BOOLEAN
Comments:	Used in non-synchronised handover tests
Description	
OM_SendNextOnHO operation sets the lower layer emulator into a special operation mode then returns. In the special mode the lower layer emulator waits for the number of Handover bursts specified by the parameter 'cnt' on channel 'ch' before sending the next L3 message. After the message is sent, the lower layer resumes normal operation automatically.The TSO does not affect any TTCN receiving event.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_StartMsrReport(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The operation starts the reporting of received MEASUREMENT REPORT message to layer 3 emulator, then it returns with value TRUE. Measurement Reports shall be disable by default. By default means that the tester filters them out from the TTCN in-buffer. This operation enables the reports entering the TTCN in-buffer. The OM_StopMsrReport operation disables the reports entering the TTCN in-buffer. When starting each test case, the measurement reports are filtered out by default.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_StopAllBCCH(cell1, cell2, cell3, cell4, cell5, cell6, cell7, cell8:CellID)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The TSO stops the RF transmission on all BCCH channels including dummy bursts in the indicated cells, but keeps the uplink CCCH reception active. After all transmission have been stopped the TSO returns TRUE. If a cell in the parameter list is not in use a "dummy" value needs to be assigned to it by the calling test case. The TSO is called in TC_26_3_3.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_StopCell(cell:CellID)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The TSO stops completely all RF transmission of the specific cell `cell`, including the dummy bursts on other slots. After all RF transmission have been stopped the TSO returns TRUE. The cell shall be deleted from the lower layer tester and therefore if the cell 'cell' is required again by the calling test case it needs to be re-configured. The TSO is called in TC_26_3_4, TC_26_6_6_1, TC_26_7_4_2_2_1, TC_26_7_4_2_2_2, TC_26_7_4_6, TC_31_6_2_3.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_StopMsrReport(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The operation stops the reporting of received MEASUREMENT REPORT message to layer 3 emulator, after measurement report stops it returns with value TRUE. Measurement Reports shall be disable by default. By default means that the tester filters them out from the TTCN in-buffer. This operation disables the reports entering the TTCN in-buffer. The OM_StartMsrReport operation enables the reports entering the TTCN in-buffer. When starting each test case, the measurement reports are filtered out by default.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ACMIncCHK(para:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
the OO_ACMIncCHK operation requests (e.g. from the control console of the test system) the test operator to read the value of ACM on SIM and check the increment of the value. Then the operator informs the test system whether the increment is as description `para`, if it is as the description the operation returns TRUE, otherwise returns FALSE. The value of ACM can be read either via MMI or by removing the SIM and using SIM reader.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ACMReading
Result Type:	BOOLEAN
Comments:	
Description	
the OO_ACMReading operation requests (e.g. from the control console of the test system) the test operator to read and note the value of ACM on SIM. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE. The value of ACM can be read either via MMI or by removing the SIM and using SIM reader.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ACMSetting
Result Type:	BOOLEAN
Comments:	
Description	
the OO_ACMSetting operation requests (e.g. from the control console of the test system) the test operator to reset the ACM to zero and set the ACMmax to 2 units. After the operator finishes the action and informs the test system it returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_AddPwrAmp
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to add power amplification at the Mobile Station under test. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_AltIndCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the alerting indication is generated. If the operator informs (e.g. from the control console of the test system) the test system that the alerting indication is generated, the operation returns TRUE. If the operator informs the test system that the alerting indication is not generated, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CalledNumCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the called party number is correctly displayed on the mobile station. If the operator informs (e.g. from the control console of the test system) the test system that the display is correct, the operation returns TRUE. If the operator informs the test system that the display is wrong, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CalledPtyNumCHK(num:CDPN)
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the called party number `num` is the one that was entered into the MS. If the operator informs (e.g. from the control console of the test system) the test system that the number is correct, the operation returns TRUE. If the operator informs the test system that the number is wrong, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckAllSMPresentBut4th
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the mobile station under test has stored all short messages during the test case but not the 4th one which should have been replaced by the 5th(cf. GSM 11.10, clause 34.2.7.4). The messages are displayed. Then the operation returns. TRUE: All SM present but not the 4th FALSE: 4th SM present or 1st, 2nd, 3rd or 5th missing.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckCBSMReceived(NumbOfIA5chara: INTEGER; msg:SMSCBpack)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check whether the MS under test has correctly received the cell broadcast short message represented by 'msg' . Then the TSO returns TRUE: if the CBSM has been correctly received, FALSE: if non or an incorrect CBSM has been received. The TSO needs to convert the 'msg' into IA5 string according to the packing character rules specified in the section 6.1.2 of ETS 300 628 (GSM 03.38). The 'NumbOfIA5chara' indicates the number of IA5 characters in the 'msg'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckMCEFOnSIM
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the Memory Capacity Exceeded Flag has been set on the SIM simulator and to inform the test system of the result of the checking. If the checking succeds the operation returns TRUE, FALSE otherwise.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckMCEFOnSIMUnset
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the Memory Capacity Exceeded Flag has been unset on the SIM simulator and to inform the test system of the result of the checking. If the checking succeds the operation returns TRUE, FALSE otherwise.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckMessageDisplayed(NumbOfIA5chara:INTEGER; msg:OCTETSTRING)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the mobile station under test indicates that an SM has arrived. If the MS provides the functionality to display MT messages, it is checked that the message contents represented by 'msg' is correctly displayed. Then the operation returns TRUE: if the MS indicates SM reception (and displays correct message), FALSE: if the MS does not indicate SM reception (or displays incorrect message). The TSO needs to convert the 'msg' into IA5 string according to the packing character rules specified in the section 6.1.2 of ETS 300 628 (GSM 03.38). The 'NumbOfIA5chara' indicates the number of IA5 characters in the 'msg'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckUssdStringDisplayed(strg: IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the mobile station under test displays the Ussd String 'strg'. Then the operation returns. TRUE: MS displays the correct Ussd String. FALSE: MS does not display the correct Ussd String.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ConnectSIMSimulator
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to connect the SIM simulator to the mobile station under test. For the contents of the SIM simulator the operator refers to the initial conditions of the test case in GSM 11.10-1. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_DepressEndKey
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to depress the 'END' key of the Mobile Stations keypad, then returns.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_DialCalledNum
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to dial the called party number at the Mobile Station under test but not press the "SEND" key (i.e. not initiate the call setup). After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_DisplaySMAndSendReplySM(n, NumbOfIA5chara: INTEGER; msg:OCTETSTRING)
Result Type:	OCTETSTRING
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to display the 'n'-th short message received in the arriving order and to check whether the short message contents represented by 'msg' is correctly displayed. Then the TSO asks the operator to send a reply short message from the mobile. After the operator finishes the action, the operation returns with the message contents sent in OCTETSTRING. The TSO needs to convert the 'msg' into IA5 string and the IA5 text entered by the operator into OCTETSTRING according to the packing character rules specified in the section 6.1.2 of ETS 300 628 (GSM 03.38). The 'NumbOfIA5chara' indicates the number of IA5 characters in the 'msg'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_DTMFIndCHK(character:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the `character` is indicated by the DTMF indicator of the MS. If the operator informs (e.g. from the control console of the test system) the test system that the MS does not give the indication of `character`, the operation returns FALSE. If the operator informs the test system that the MS gives the indication, the operation returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_EmptyMessageStorage
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to empty the message storage of the mobile station under test. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_EnterPswd(pswd:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This TSO displays a message on the control console to inform the operator that he is to enter a given password (the actual password is given by 'pswd') on the mobile. The operation waits for the operator to acknowledge the request, which may be done before or after entering the password into the mobile, and then returns the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_HookOff
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to answer the mobile terminating call at the Mobile Station under test. This operation waits for the operator to acknowledge the request which may be done before or after the call has been answered and then returns the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_Telephony
Result Type:	BOOLEAN
Comments:	
Description	
This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or configuration of the MS in such a way that the MS is able to successfully receive the call for telephony service. If the MS supports both telephony without immediate connection mode and telephony with immediate connection mode, set the MS in telephony without immediate connection mode. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE. This TSO is used in MO call test cases except TC_11_2, where this TSO is call for single-numbering scheme MT testing.	
Detailed Comments:	used for TS11 and TS12

Test Suite Operation Definition

Operation Name: OO_IFsetup_BS2xorBS3x_MO(srv:SERVICES; itc, sacp:B_3; ce:B_2; rate:IA5String)

Result Type: BOOLEAN

Comments:

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS, when trigged, is able to initiate the call for the BS2x or BS3x service `srv` with the specified itc, sacp, ce values. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

This TSO is used in MO call test cases except TC_11_2, where this TSO is call for single-numbering scheme MT testing.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT paramters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:

1) srv ----- basic service

- C_Async300 Basic Service 21, Asynchronous 300 bps
- C_Async1200 Basic Service 22, Asynchronous 1.2 kbps
- C_Async120075 Basic Service 23, Asynchronous 1200/75 bps
- C_Async2400 Basic Service 24, Asynchronous 2.4 kbps
- C_Async4800 Basic Service 25, Asynchronous 4.8 kbps
- C_Async9600 Basic Service 26, Asynchronous 9.6 kbps
- C_Sync1200 Basic Service 31, Synchronous 1.2 kbps
- C_Sync2400 Basic Service 32, Synchronous 2.4 kbps
- C_Sync4800 Basic Service 33, Synchronous 4.8 kbps
- C_Sync9600 Basic Service 34, Synchronous 9.6 kbps

2) itc ------ information transfer capability

- '001'B unrestricted digital information
- '010'B 3.1 kHz audio, ex PLMN

3) sacp ---- signalling access protocol

- '001'B I.440/450
- '010'B X.21
- '101'B X.28 - non dedicated PAD
- '110'B X.32

4) ce ------ connection element

- '00'B transparent
- '01'B non transparent

5) rate ----- channel rate

- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: Used for BS21, BS22, BS23, BS24, BS25, BS26, BS31, BS32, BS33, BS34

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS4x_MO(srv:SERVICES; ce:B_2; rate:IA5String)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, orthe MS configuration in such a way that the MS, when trigged, is able to initiate the call for the BS4x service `srv` with specified ur and ce values. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:

1) srv ----- basic service

- C_PAD300 Basic Service 41, PAD access 300 bps
- C_PAD1200 Basic Service 42, PAD access 1.2 kbps
- C_PAD120075 Basic Service 43, PAD access 1200/75 bps
- C_PAD2400 Basic Service 44, PAD access 2.4 kbps
- C_PAD4800 Basic Service 45, PAD access 4.8 kbps
- C_PAD9600 Basic Service 46, PAD access 9.6 kbps

2) ce ------ connection element

- '00'B transparent
- '01'B non transparent

3) rate ----- channel rate

- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: Used for BS41, BS42, BS43, BS44, BS45, BS46

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS5x_MO(srv:SERVICES; ur:B_4; rate:IA5String)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, orthe MS configuration in such a way that the MS, when trigged, is able to initiate the call for the BS5x service `srv` with the specifyied ur value. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:

1) srv ----- basic service

- C_Packet2400 Basic Service 51, Packet access 2.4 kbps
- C_Packet4800 Basic Service 52, Packet access 4.8 kbps
- C_Packet9600 Basic Service 53, Packet access 9.6 kbps

2) ur ------ user rate

- '0011'B 2400 bit/s
- '0100'B 4800 bit/s
- '0101'B 9600 bit/s

3) rate ----- channel rate

- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: Used for BS51, BS52, BS53

Test Suite Operation Definition

Operation Name: OO_IFsetup_BS61orBS81_MO(srv:SERVICES; ur:B_4; sa:B_1; ce:B_2; rate:IA5String)

Result Type: BOOLEAN

Comments:

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS, when trigged, is able to initiate the call for BS61 or BS81 with the specified ur, ce and sa values, and the first bearer capability is speech. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

This TSO is used in MO call test cases except TC_11_2, where this TSO is call for single-numbering scheme MT testing.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:

1) srv ----- basic service

- C_AltSpchData Basic Service 61, Alternatie Speech/data
- C_SpchData Basic Service 81, Speech followed by data

2) ur ------ user rate

- '0001'B 300 bit/s
- '0010'B 1200 bit/s
- '0111'B 1200/75 bit/s
- '0011'B 2400 bit/s
- '0100'B 4800 bit/s
- '0101'B 9600 bit/s

3) sa ------ synchronous/asynchronous

- '0'B synchronous
- '1'B asynchronous

4) ce ------ connection element

- '00'B transparent
- '01'B non transparent

5) rate ----- channel rate

- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: used for BS61 and BS81

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_TS6x_MO(srv:SERVICES; ur:B_4; ce:B_2; rate:IA5String)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS, when trigged, is able to initiate the call for TS61 and TS62 with the specified ur and ce values, and the first bearer capability is speech. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

This TSO is used in MO call test cases except TC_11_2, where this TSO is call for single-numbering scheme MT testing.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:
1) srv ----- basic service
- C_AltSpchFax Teleservice 61, Alternatie Speech and Facsimile group 3
- C_AutoFax Teleservice 62, Automatic Facsimile group 3

2) ur ------ user rate
- '0011'B 2400 bit/s
- '0100'B 4800 bit/s
- '0101'B 9600 bit/s

3) ce ------ connection element
- '00'B transparent
- '01'B non transparent

4) rate ----- channel rate
- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: used for TS61 and TS62

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS2x_MT(srv:SERVICES; itc:B_3)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS is able to successfully receive the call for the BS2x service `srv` with the specified itc value. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT paramters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

Detailed Comments: Used for BS21, BS22, BS24, BS25, BS26

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS3x_MT(srv:SERVICES; itc, sacp:B_3)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, orthe MS configuration in such a way that the MS is able to successfully receive the call for the BS3x service `srv` with specified itc and sacp values. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

Detailed Comments: Used for BS31, BS32, BS33, BS34

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS61orBS81_MT(srv:SERVICES; ur:B_4; sa:B_1)
Result Type:	BOOLEAN
Comments:	
Description	
This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS is able to successfully receive the call for BS61 or BS81 with the specified ur and sa values. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE. Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.	
Detailed Comments:	used for BS61 and BS81

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_TS6x_MT(srv:SERVICES; ur:B_4)
Result Type:	BOOLEAN
Comments:	
Description	
This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS is able to successfully receive the call for TS61 and TS62 with the specified ur value. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE. Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.	
Detailed Comments:	used for TS61 and TS62

Test Suite Operation Definition	
Operation Name:	OO_InCallModi(srv:SERVICES)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to initiate a in-call modification at the Mobile Station under test, which is triggered by the calling tone identification (CNG) received by the MS, for the basic service 'srv', then returns TRUE. The calling test step can pass the 'srv' value either as TSPX or constant to the TSO. The possible service values include (see also test suite constant declarations): 1) Teleservices - C_Telephony - C_EmgCall - C_AltSpchFax - C_AutoFax 2) Bearer services - C_Async300, C_Async1200, C_Async120075, C_Async2400, C_Async4800, C_Async9600 - C_Sync1200, C_Sync2400, C_Sync4800, C_Sync9600 - C_PAD300, C_PAD1200, C_PAD120075, C_PAD2400, C_PAD4800, C_PAD9600 - C_Packet2400, C_Packet4800, C_Packet9600 - C_AltSpchData - C_SpchData	
Detailed Comments:	

Test Suite Operation Definition

Operation Name: OO_InitCall(srv:SERVICES)
Result Type: BOOLEAN
Comments:

Description

This operation requests (e.g. from the control console of the test system) the test operator to initiate an MS originating call for the basic service `srv`, then returns TRUE. The calling test step can pass the 'srv' value either as TSPX or constant to the TSO. The necessary MS configuration has been done in the corresponding TSO OO_IFsetup_xx...xx_MO before this TSO is called.

The possible service values include (see also test suite constant declarations):

1) Teleservices

- C_Telephony ----- TS11
- C_EmgCall ----- TS12
- C_AltSpchFax ----- TS61
- C_AutoFax ----- TS62

- C_Short_Message_MT/PP ---- TS21
- C_Short_Message_MO/PP ---- TS22
- C_Short_Message_CB ---- TS23

2) Bearer services

- C_Async300 ----- BS21
- C_Async1200 ----- BS22
- C_Async120075 ----- BS23
- C_Async2400 ----- BS24
- C_Async4800 ----- BS25
- C_Async9600 ----- BS26
- C_Sync1200 ----- BS31
- C_Sync2400 ----- BS32
- C_Sync4800 ----- BS33
- C_Sync9600 ----- BS34
- C_PAD300 ----- BS41
- C_PAD1200 ----- BS42
- C_PAD120075 ----- BS43
- C_PAD2400 ----- BS44
- C_PAD4800 ----- BS45
- C_PAD9600 ----- BS46
- C_Packet2400 ----- BS51
- C_Packet4800 ----- BS52
- C_Packet9600 ----- BS53
- C_AltSpchData ----- BS61
- C_SpchData ----- BS81

Detailed Comments:

Test Suite Operation Definition

Operation Name: OO_InitNonCallSS
Result Type: BOOLEAN
Comments:

Description

This operation requests (e.g. from the control console of the test system) the test operator to initiate a non call related supplementary service which is supported by the MS.

After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OO_InitSS(action: IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to initiate the required supplementary service by the MMI sequence `action`. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_InServiceCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the Mobile Station under test is in normal service state ("Idle, updated")-- listening to CCCH and BCCH and with U1 UPDATED status. If the operator informs (e.g. from the control console of the test system) the test system that the MS does not give any service indication, the operation returns FALSE. If the operator informs the test system that the MS gives service indication, the operation returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_Key
Result Type:	BOOLEAN
Comments:	
Description	
This operation returns TRUE if a key of the SS' keyboard has been pressed. Otherwise it returns FALSE. After OO_PressKeyWhenInService was called, the TSO is then continuously called until the test operator hits a key or a timer expires.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_MptyCall
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to enter MultiParty MMI command at the Mobile Station under test, then it returns with the value TRUE after the operator finishes the action and informs the test system.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_MSSetupStoreClass1SMInMEMemory
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to setup the MS under test to store class 1 SM in the ME memory (by way of MMI, as described in PICS/PIXIT statement). After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PLMNsCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the Mobile Station presents a list of available PLMNs. The list shall include the MNC and MCC of cells 1 to 7, but not cell 8 (for GSM900) or cells 1 to 6, but not cell 7 (for DCS1800). If the operator informs (e.g. from the control console of the test system) the test system that the MS correctly presents the list, the operation returns TRUE. If the operator informs the test system that the list is incorrect, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PLMNselModeAuto
Result Type:	BOOLEAN
Comments:	
Description	
This operation switches the PLMN selection mode of the MS to automatic selection. After correctly switching of the mode the operation returns TRUE otherwise FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PLMNselModeMan
Result Type:	BOOLEAN
Comments:	
Description	
This operation switches the PLMN selection mode of the MS to manual selection. After correctly switching of the mode the operation returns TRUE otherwise FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PowerUp
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to power up the Mobile Station under test and to inform after the test system, then it returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PowerDown
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to power down the Mobile Station under test and to inform after the test system, then it returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PressKeyWhenInService
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to press any key when the MS shows the service indicator, then returns. The return value is always TRUE. Working with OO_Key together, the TSO displays a prompt to the test operator and then immediately returns.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_RecallAndDisplaySM(NumbOfIA5chara: INTEGER; msg: OCTETSTRING)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to recall and display an SM stored in the ME of the MS under test e.g. by means of the MMI. Then the operator checks whether the short message contents represented by 'msg' is correctly displayed. The TSO returns TRUE: if SM can be recalled and displayed correctly in comparison with the 'msg', FALSE: if SM can not be recalled or displayed, or displayed incorrectly in comparison with the 'msg'. The TSO needs to convert the 'msg' into IA5 string according to the packing character rules specified in the section 6.1.2 of ETS 300 628 (GSM 03.38). The 'NumbOfIA5chara' indicates the number of IA5 characters in the 'msg'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_RemoveOneSM
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to remove one of the SM from the message store and then to inform the test system. It does not matter which short message has been removed. After that the operation returns with value TRUE. The returned value is not used in the ATS.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_RemvPwrAmp
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to remove the added power amplification at the Mobile Station under test and to inform after the test system, then the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_RFoutputCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the Mobile Station produces any RF output. If the operator informs (e.g. from the control console of the test system) the test system that the MS does not produce any RF output, the operation returns FALSE. If the operator informs the test system that the MS produces RF output, the operation returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SelPLMN(par_plmn:OCTETSTRING)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to select the given PLMN in par_plmn manually, and to inform after the test system then the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SendMOShortMessage
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to send a short message from the mobile station under test. The operation waits for the operator to acknowldge the request which may be done before or after sending the MO Short Message, and then returns the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SendSMSCOMMANDDel
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to send a SMS COMMAND message from the mobile station under test containing requiring to delete the previouly submited SM and then to inform the test system. Then, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SendSMSCOMMANDEnq
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to send a SMS COMMAND message from the mobile station under test containing an enquiry about the previouly submited SMand then to inform the test system. Then, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SetRefuseCall
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to enable call refusal on the MS. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ShortKeyDepr(ch:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to cause a DTMF tone to be generated, e.g. by short depression of the key `ch` on the Mobile Station under test. The operation waits for the operator to acknowldge the request which may be done before or after generating the DTMF tone, and then returns the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIMIns
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to insert the SIM card into the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIM2Ins
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to insert the SIM card 2 into the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE. The SIM Card 2 shall contain the following parameter values, which are different from dafault values(SIM Card 1): IMSI= '001011234' HPLMN_search_period=6min.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIM3Ins
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to insert the SIM card 3 into the Mobile Station under test and turn the MS power on, after the test operator finishes the action and informs the test system it returns with value TRUE. The SIM Card 3 shall contain fixed dialling number allocated and activated.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIMRmv
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to remove the SIM card from the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIMSimulAttIndOK
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check whether the SIM simulator indicates an attempt made by the ME to store the short message in the SIM. The SIM simulator returns the status response "OK" ('90 00"). Then returns. Attempt indicated: TRUE Attempt NOT indicated: FALSE	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIMSimulAttIndMemProblem
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check whether the SIM simulator indicates an attempt made by the ME to store the short message in the SIM. The SIM simulator returns the status response "Memory Problem" ('92 40"). Then returns. Attempt indicated: TRUE Attempt NOT indicated: FALSE	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SSresultCHK(svc:INTEGER)
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check the user indication of the result of the supplementary service ` svc`. If the operator informs (e.g. from the control console of the test system) the test system that the indication is correct, the operation returns TRUE. If the operator informs the test system that the indication is wrong, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SwitchOn
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to switch on (or if there is no switch then to restore the power to) the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SwitchOff
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to switch off (or if there is no switch then to remove the power from) the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_TCHThroConnCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the TCH is through connected. The SS has to generate a noise pattern so that the operator can check this. If the operator informs (e.g. from the control console of the test system) the test system that the TCH is through connected, the operation returns TRUE. If the operator informs the test system that the TCH is not through connected, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_TermCall
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to terminate the ongoing call. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, it returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_TguardTimeOut
Result Type:	BOOLEAN
Comments:	
Description	
This operation informs (e.g. from the control console of the test system) the test operator that the T_guard timer has expired during the test case leading to an inconclusive or fail verdict and returns immediately with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ToneStopCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the MS stops the tone generation If the operator informs (e.g. from the control console of the test system) the test system that the MS does not stop tone generation, the operation returns FALSE. If the operator informs the test system that the MS stops tone generation, the operation returns TRUE. Definition of "tone generation": -During MO call: MS generates alerting tone/presents inband information sent by the network. -During MT call: MS generates ringing tone.	
Detailed Comments:	

Test suite parameter declarations

Test Suite Parameter Declarations			
Parameter Name	**Type**	**PICS/PIXIT Ref**	**Comments**
TSPC_24DataF	BOOLEAN	PICS Table A.25 Item 12	2.4 k full rate data mode supported
TSPC_24DataH	BOOLEAN	PICS Table A.25 Item 13	2.4 k half rate data mode supported
TSPC_48DataF	BOOLEAN	PICS Table A.25 Item 14	4.8 k full rate data mode supported
TSPC_48DataH	BOOLEAN	PICS Table A.25 Item 15	4.8 k half rate data mode supported
TSPC_96Data	BOOLEAN	PICS Table A.25 Item 16	9.6 k full rate data mode supported
TSPC_Feat_A51	BOOLEAN	PICS Table A.2 Item 17	ciphering algorithm A5/1 supported
TSPC_Feat_A52	BOOLEAN	PICS Table A.2 Item 18	ciphering algorithm A5/2 supported
TSPC_AddCharSet	BOOLEAN	PICS Table A.25 Item 47	A, B, C, D chars supported
TSPC_AddInfo_PseudoSynch	BOOLEAN	PICS	Pseudo synchronised supported
TSPC_AlertInd	BOOLEAN	PICS Table A.25 Item 49	alerting indication to the user supported
TSPC_SvcOnTCH	BOOLEAN	PICS Table A.25 Item 22	at least one service on traffic channel supported
TSPC_SMS	BOOLEAN	PICS Table A.25 Item 31	at least one short message service supported
TSPC_SS	BOOLEAN	PICS Table A.25 Item 29	at least one supplementary service supported
TSPC_AutoAutoMode	BOOLEAN	PICS Table A.25 Item 48	automatically enter automatic selection of PLMN mode supported
TSPC_BasCharSet	BOOLEAN	PICS Table A.25 Item 46	Chars 0-9, *, # supported
TSPC_CalledNumDisp	BOOLEAN	PICS Table A.2 Item 1	called number display supported
TSPC_DCS	BOOLEAN	PICS Table A.1 Item 3	testing DCS1800
TSPC_DetachOnPwrDn	BOOLEAN	PICS Table A.25 Item 38	detach on power down supported
TSPC_DetachOnSIMRmv	BOOLEAN	PICS Table A.25 Item 39	detach on SIM remove supported
TSPC_DispRcvSMS	BOOLEAN	PICS Table A.25 Item 34	display of received SMS supported
TSPC_DualRate	BOOLEAN	PICS Table A.25 Item 23	dual rate channel types supported
TSPC_EGSM	BOOLEAN	PICS Table A.1 Item 2	both standard and extended GSM band supported
TSPC_EmgOnly	BOOLEAN	PICS Table A.25 Item 27	the only circuit switched basic service is emergency call
TSPC_followOnReq	BOOLEAN	PICS Table A.25 Item 53	follow-on request procedure supported
TSPC_HalfRateData	BOOLEAN	PICS Table A.25 Item 6	half rate data supported
TSPC_HalfRateSpeech	BOOLEAN	PICS Table A.25 Item 3	half rate speech mode supported
TSPC_InCallMod	BOOLEAN	PICS Table A.25 Item 52	In-Call modification supported
TSPC_NoimmConn	BOOLEAN	PICS Table A.25 Item 51	at least one service not support immediate connection
TSPC_Feat_FND	BOOLEAN	PICS Table A.2 Item 21	FND feature supported
TSPC_NonCallSS	BOOLEAN	PICS Table A.25 Item 30	non call related supplementary service supported
TSPC_MTsvc	BOOLEAN	PICS Table A.25 Item 19	at least one MT circuit switched basic service supported
TSPC_MOsvc	BOOLEAN	PICS Table A.25 Item 20	at least one MO circuit switched basic service supported
TSPC_PGSM	BOOLEAN	PICS Table A.1 Item 1	only standard GSM band supported
TSPC_RefusalCall	BOOLEAN	PICS Table A.25 Item 54	refusal of call supported
TSPC_ReplaceSMS	BOOLEAN	PICS Table A.25 Item 33	replace SMS supported
TSPC_ReplyProc	BOOLEAN	PICS Table A.25 Item 32	(SMS) reply procedures supported
TSPC_RFAmp	BOOLEAN	PICS Table A.25 Item 55	RF amplification supported
TSPC_SDCCHOnly	BOOLEAN	PICS Table A.25 Item 21	only SDCCH supported
TSPC_Serv_SS_AoCC	BOOLEAN	PICS Table A.5 Item 14	AOCC SS supported
TSPC_Serv_SS_BAIC	BOOLEAN	PICS Table A.5 Item 18	BAIC SS supported
TSPC_Serv_SS_BI	BOOLEAN	PICS	BI SS supported

TSPC_Serv_SS_BICRoam	BOOLEAN	PICS Table A.5 Item 19	BICRoam SS supported
TSPC_Serv_SS_BAOC	BOOLEAN	PICS Table A.5 Item 15	BAOC SS supported
TSPC_Serv_SS_BOIC	BOOLEAN	PICS Table A.5 Item 16	BOIC SS supported
TSPC_Serv_SS_BOICexHC	BOOLEAN	PICS Table A.5 Item 17	BOICExHC SS supported
TSPC_Serv_SS_CFB	BOOLEAN	PICS Table A.5 Item 6	CFB supported
TSPC_Serv_SS_CFNRy	BOOLEAN	PICS Table A.5 Item 7	CFNry supported
TSPC_Serv_SS_CFNRc	BOOLEAN	PICS Table A.5 Item 8	CFNRc supported
TSPC_Serv_SS_CFU	BOOLEAN	PICS Table A.5 Item 5	CFU supported
TSPC_Serv_SS_HOLD	BOOLEAN	PICS Table A.5 Item 10	Call Hold SS supported
TSPC_Serv_SS_MPTY	BOOLEAN	PICS Table A.5 Item 11	multiparty SS supported
TSPC_Serv_SS_unstruct	BOOLEAN	PICS	USSD supported
TSPC_Serv_TS11	BOOLEAN	PICS Table A.3 Item 1	telephony supported
TSPC_Serv_TS12	BOOLEAN	PICS Table A.3 Item 2	emergency call supported
TSPC_Serv_TS21	BOOLEAN	PICS Table A.3 Item 3	SMS MT/PP supported
TSPC_Serv_TS22	BOOLEAN	PICS Table A.3 Item 4	SMS MO/PP supported
TSPC_Serv_TS23	BOOLEAN	PICS Table A.3 Item 5	SMS cell broadcast supported
TSPC_Serv_TS61	BOOLEAN	PICS	alternate speech and G3 fax (TS61) supported
TSPC_Serv_TS62	BOOLEAN	PICS	automatic G3 fax (TS62) supported
TSPC_Serv_BS21	BOOLEAN	PICS Table A.4 Item 1	data circuit duplex async 300 bit/s supported
TSPC_Serv_BS22	BOOLEAN	PICS Table A.4 Item 2	data circuit duplex async 1200 bit/s supported
TSPC_Serv_BS23	BOOLEAN	PICS Table A.4 Item 3	data circuit duplex async 1200/75 bit/s supported
TSPC_Serv_BS24	BOOLEAN	PICS Table A.4 Item 4	data circuit duplex async 2400 bit/s supported
TSPC_Serv_BS25	BOOLEAN	PICS Table A.4 Item 5	data circuit duplex async 4800 bit/s supported
TSPC_Serv_BS26	BOOLEAN	PICS Table A.4 Item 6	data circuit duplex async 9600 bit/s supported
TSPC_Serv_BS31	BOOLEAN	PICS Table A.4 Item 7	data circuit duplex sync 1200 bit/s supported
TSPC_Serv_BS32	BOOLEAN	PICS Table A.4 Item 8	data circuit duplex sync 2400 bit/s supported
TSPC_Serv_BS33	BOOLEAN	PICS Table A.4 Item 9	data circuit duplex sync 4800 bit/s supported
TSPC_Serv_BS34	BOOLEAN	PICS Table A.4 Item 10	data circuit duplex sync 9600 bit/s supported
TSPC_Serv_BS41	BOOLEAN	PICS Table A.4 Item 11	PAD access 300 bit/s supported
TSPC_Serv_BS42	BOOLEAN	PICS Table A.4 Item 12	PAD access 1200 bit/s supported
TSPC_Serv_BS43	BOOLEAN	PICS Table A.4 Item 13	PAD access 1200/75 bit/s supported
TSPC_Serv_BS44	BOOLEAN	PICS Table A.4 Item 14	PAD access 2400 bit/s supported
TSPC_Serv_BS45	BOOLEAN	PICS Table A.4 Item 15	PAD access 4800 bit/s supported
TSPC_Serv_BS46	BOOLEAN	PICS Table A.4 Item 16	PAD access 9600 bit/s supported
TSPC_Serv_BS51	BOOLEAN	PICS Table A.4 Item 17	packet access 2400 bit/s supported
TSPC_Serv_BS52	BOOLEAN	PICS Table A.4 Item 18	packet access 4800 bit/s supported
TSPC_Serv_BS53	BOOLEAN	PICS Table A.4 Item 19	packet access 9600 bit/s supported
TSPC_Serv_BS61	BOOLEAN	PICS	alternate speech/data (BS61) supported
TSPC_Serv_BS81	BOOLEAN	PICS	speech followed data (BS81) supported
TSPC_SMSStatusRepCap	BOOLEAN	PICS Table A.25 Item 35	SMS status report capabilities supported
TSPC_StoreRcvSMSME	BOOLEAN	PICS Table A.25 Item 37	Storage of received SMS in ME supported
TSPC_StoreRcvSMSSIM	BOOLEAN	PICS Table A.25 Item 36	Storage of received SMS in SIM supported
TSPC_SIMRmv	BOOLEAN	PICS Table A.25 Item 40	SIM removable without power down supported
TSPC_SwitchOnOff	BOOLEAN	PICS Table A.2 Item 15	switch on/off supported
TSPC_TranspDataOnly	BOOLEAN	PICS Table A.25 Item 9	only transparent data service supported
TSPC_CC	BOOLEAN	PICS Table A.25 Item 26	CC protocol for at least one BC

			supported
TSPC_TeleSvc	BOOLEAN	PICS Table A.25 Item 25	at least one teleservice supported
TSPC_EFR	BOOLEAN		Set to TRUE for GSM-900 Phase 2+ (EFR) MS which supports any EFR features, and FALSE otherwise.
TSPC_EFR_Speech_v2	BOOLEAN		Set to TRUE for GSM-900 Phase 2+ (EFR) MS which supports Speech Version 2 and FALSE otherwise.
TSPC_EFR_Speech_v3	BOOLEAN		Set to TRUE for GSM-900 Phase 2+ (EFR) MS which supports Speech Version 3 and FALSE otherwise.
TSPC_EFR_EmgCallBcap	BOOLEAN		Set to TRUE if received ESETUP msg contains bearer capability IE, otherwise FALSE.
TSPC_FullRateSpeech	BOOLEAN	PICS Table A.25 Item 2	full rate speech mode supported
TSPC_FullRateOnly	BOOLEAN	PICS Table A.25 Item 24	only full rate channel type supported
TSPX_AltNb	BOOLEAN	PIXIT	use alternative neighbour cells description. default : FALSE
TSPX_TE_stopbit	B_1	PIXIT	TE conf. number stop bits. default value : '0'B, 1 bit
TSPX_TE_databit	B_1	PIXIT	TE conf. number data bits. default value : '1'B, 8 bits
TSPX_TE_FLCT	FLWCNTL	PIXIT	TE flow control. default : 0, outband flow control
TSPX_TE_parity	B_3	PIXIT	TE conf. parity. default : '001'B, no parity
TSPX_BS_21_itc1	B_3	PIXIT	itc value supported for BS21. default value : '001'B, UDI
TSPX_BS_21_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS21. default value : FALSE, only one itc
TSPX_BS_21_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS21. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_21_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_21_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent mode
TSPX_BS_21_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_21_itc2	B_3	PIXIT	other itc value supported by BS21, it shall be differenet from the previous one if the answer to TSPX_BS_21_more_itc is TRUE. default value :'001'B, UDI
TSPX_BS_22_itc1	B_3	PIXIT	itc value supported by BS22. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_22_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS22. default value : FALSE, only one itc
TSPX_BS_22_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS22. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_22_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_22_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent
TSPX_BS_22_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_22_itc2	B_3	PIXIT	other itc value supported by

			BS22, it shall be differenet from the previous one if the answer to TSPX_BS_22_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_23_itc	B_3	PIXIT	itc value supported by BS23. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_23_T_NT	BOOLEAN	PIXIT	both T and NT supported for BS23. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_23_ce	B_2	PIXIT	ce value used in MO call. default value : '00'B, transparent
TSPX_BS_23_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_24_itc1	B_3	PIXIT	itc value supported by BS24. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_24_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS24. default value : FALSE, only one itc
TSPX_BS_24_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS24. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_24_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_24_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_24_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_24_itc2	B_3	PIXIT	other itc value supported by BS24, it shall be differenet from the previous one if the answer to TSPX_BS_24_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_25_itc1	B_3	PIXIT	itc value supported by BS25. default value : '001'B, UDI
TSPX_BS_25_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS25. default value : FALSE, only one itc
TSPX_BS_25_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS25. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_25_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_25_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_25_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_25_itc2	B_3	PIXIT	other itc value supported by BS25, it shall be differenet from the previous one if the answer to TSPX_BS_25_more_itc is TRUE. default value : '001'B, UDI
TSPX_BS_26_itc1	B_3	PIXIT	itc value supported by BS26. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_26_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS26. default value : FALSE, only one itc
TSPX_BS_26_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS26. default value : FALSE, not support both transparent and non transparent modes

TSPX_BS_26_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_26_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_26_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_26_itc2	B_3	PIXIT	other itc value supported by BS26, it shall be differenet from the previous one if the answer to TSPX_BS_26_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_31_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS31. default value : FALSE, only one itc
TSPX_BS_31_more_sacp	BOOLEAN	PIXIT	more than one sacp value supported by BS31. default value : FALSE, only one sacp
TSPX_BS_31_itc1	B_3	PIXIT	itc value supported by BS31. default value : '001'B, UDI
TSPX_BS_31_sacp1	B_3	PIXIT	sacp value supported by BS31. default value : '001'B, I440/450
TSPX_BS_31_itc2	B_3	PIXIT	other itc value supported by BS31, it shall be differenet from the previous one if the answer to TSPX_BS_31_more_itc is TRUE. default value : '001'B, UDI
TSPX_BS_31_sacp2	B_3	PIXIT	other sacp value supported by BS31, it shall be differenet from the previous one if the answer to TSPX_BS_31_more_sacp is TRUE. default value : '001'B, I440/450
TSPX_BS_32_itc1	B_3	PIXIT	itc value supported by BS32. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_32_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS32. default value : FALSE, only one itc
TSPX_BS_32_more_sacp	BOOLEAN	PIXIT	more than one sacp value supported by BS32. default value : FALSE, only one sacp
TSPX_BS_32_sacp1	B_3	PIXIT	sacp value supported by BS32. default value : '110'B, X.32
TSPX_BS_32_X32_T_NT	BOOLEAN	PIXIT	BS 32 both T and NT supported for X32 3.1kHz. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_32_X32_ce	B_2	PIXIT	if the answer to TSPX_BS_32_X32_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent
TSPX_BS_32_itc2	B_3	PIXIT	other itc value supported by BS32, it shall be differenet from the previous one if the answer to TSPX_BS_32_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_32_sacp2	B_3	PIXIT	other sacp value supported by BS32, it shall be differenet from the previous one if the answer to TSPX_BS_32_more_sacp is TRUE. default value : '110'B, X.32
TSPX_BS_33_itc1	B_3	PIXIT	itc value supported by BS33. default value : '001'B, UDI
TSPX_BS_33_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS33. default value : FALSE, only one itc

TSPX_BS_33_more_sacp	BOOLEAN	PIXIT	more than one sacp value supported by BS33. default value : FALSE, only one sacp
TSPX_BS_33_sacp1	B_3	PIXIT	sacp value supported by BS33. default value : '010'B, X.21
TSPX_BS_33_X32_T_NT	BOOLEAN	PIXIT	BS 33 both T and NT supported for X32 3.1kHz. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_33_X32_ce	B_2	PIXIT	if the answer to TSPX_BS_33_X32_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_33_itc2	B_3	PIXIT	other itc value supported by BS32, it shall be differenet from the previous one if the answer to TSPX_BS_33_more_itc is TRUE. default value : '001'B, UDI
TSPX_BS_33_sacp2	B_3	PIXIT	other sacp value supported by BS33, it shall be differenet from the previous one if the answer to TSPX_BS_33_more_sacp is TRUE. default value : '010'B, X.21
TSPX_BS_34_itc1	B_3	PIXIT	itc value supported by BS34. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_34_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS34. default value : FALSE, only one itc
TSPX_BS_34_more_sacp	BOOLEAN	PIXIT	more than one sacp value supported by BS34. default value : FALSE, only one sacp
TSPX_BS_34_sacp1	B_3	PIXIT	sacp value supported by BS34. default value : '001'B, I440/450
TSPX_BS_34_X32_T_NT	BOOLEAN	PIXIT	BS 34 both T and NT supported for X32 3.1kHz. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_34_X32_ce	B_2	PIXIT	if the answer to TSPX_BS_34_X32_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_34_itc2	B_3	PIXIT	other itc value supported by BS34, it shall be differenet from the previous one if the answer to TSPX_BS_34_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_34_sacp2	B_3	PIXIT	other sacp value supported by BS34, it shall be differenet from the previous one if the answer to TSPX_BS_34_more_sacp is TRUE. default value : '001'B, I440/450
TSPX_BS_41_T_NT	BOOLEAN	PIXIT	BS41 both T and NT supported . default value : FALSE, not support both
TSPX_BS_41_ce	B_2	PIXIT	ce value used in BS41 MO call. default value : '00'B, transparent
TSPX_BS_42_T_NT	BOOLEAN	PIXIT	BS42 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_42_ce	B_2	PIXIT	ce value used in BS42 MO call. default value : '010'B, non transparent
TSPX_BS_43_T_NT	BOOLEAN	PIXIT	BS43 both T and NT supported . default value : FALSE, not

			support both transparent and non transparent modes
TSPX_BS_43_ce	B_2	PIXIT	ce value used in BS43 MO call. default value : '00'B, transparent
TSPX_BS_44_T_NT	BOOLEAN	PIXIT	BS44 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_44_ce	B_2	PIXIT	ce value used in BS44 MO call. default value : '00'B, transparent
TSPX_BS_45_T_NT	BOOLEAN	PIXIT	BS45 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_45_ce	B_2	PIXIT	ce value used in BS45 MO call. default value : '01'B, non transparent
TSPX_BS_46_T_NT	BOOLEAN	PIXIT	BS46 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_46_ce	B_2	PIXIT	ce value used in BS46 MO call. default value : '00'B, transparent
TSPX_BS_61_S	BOOLEAN	PIXIT	synchronous data supported by BS 61 . default value : TRUE, synchronous mode
TSPX_BS_61_S_more_ur	BOOLEAN	PIXIT	more than one user rate for synchronous data service supported by BS61. default value : FALSE, only ine user rate
TSPX_BS_61_S_ur1	B_4	PIXIT	usrer rate value supported for synchronous data service of BS61. default value : '0100'B, 4.8 kbit/s
TSPX_BS_61_S_ur2	B_4	PIXIT	another usrer rate value supported for synchronous data service of BS61. it shall be different from the previous one if the answer to TSPX_BS_61_S_more_ur is TRUE. default value : '0100'B, 4.8 kbit/s
TSPX_BS_61_A	BOOLEAN	PIXIT	asynchronous data service supported by BS61 . default value : TRUE, asynchronous
TSPX_BS_61_A_ur1	B_4	PIXIT	usrer rate value supported for asynchronous data service of BS61. default value : '0101'B, 9.6 kbit/s
TSPX_BS_61_A_ur1_T_NT	BOOLEAN	PIXIT	both T and NT supported for the rate TSPX_BS_61_A_ur1 of BS61. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_61_A_ur1_ce	B_2	PIXIT	if the answer to TSPX_BS_61_A_ur1_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_61_A_more_ur	BOOLEAN	PIXIT	more than one user rate for asynchronous data service supported by BS61. default value : FALSE, only one user rate
TSPX_BS_61_A_ur2	B_4	PIXIT	another usrer rate value supported for asynchronous data service of BS61. it shall be different from the previous one if the answer to TSPX_BS_61_A_more_ur is TRUE. default value : '0101'B, 9.6 kbit/s
TSPX_BS_61_A_ur2_T_NT	BOOLEAN	PIXIT	both T and NT supported for the rate TSPX_BS_61_A_ur2 of BS61. default value : FALSE, not

			support both transparent and non transparent modes
TSPX_BS_81_S	BOOLEAN	PIXIT	synchronous data supported by BS 81. default value : TRUE, synchronous mode
TSPX_BS_81_S_more_ur	BOOLEAN	PIXIT	more than one user rate for synchronous data service supported by BS81. default value : FALSE, only one user rate
TSPX_BS_81_S_ur1	B_4	PIXIT	usrer rate value supported for synchronous data service of BS81. default value : '0101'B, 9.6 kbit/s
TSPX_BS_81_S_ur2	B_4	PIXIT	another usrer rate value supported for synchronous data service of BS81. it shall be different from the previous one if the answer to TSPX_BS_81_S_more_ur is TRUE. default value : '0101'B, 9.6 kbit/s
TSPX_BS_81_A	BOOLEAN	PIXIT	asynchronous data service supported by BS81. default value : TRUE, asynchronous mode
TSPX_BS_81_A_ur1	B_4	PIXIT	usrer rate value supported for asynchronous data service of BS81. default value : '0100'B, 4.8 kbit/s
TSPX_BS_81_A_ur1_T_NT	BOOLEAN	PIXIT	both T and NT supported for the rate TSPX_BS_81_A_ur1 of BS81. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_81_A_ur1_ce	B_2	PIXIT	if the answer to TSPX_BS_81_A_ur1_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent
TSPX_BS_81_A_more_ur	BOOLEAN	PIXIT	more than one user rate for asynchronous data service supported by BS81. default value : FALSE, only one user rate
TSPX_BS_81_A_ur2	B_4	PIXIT	another usrer rate value supported for asynchronous data service of BS81. it shall be different from the previous one if the answer to TSPX_BS_81_A_more_ur is TRUE. default value : '0100'B, 4.8 kbit/s
TSPX_BS_81_A_ur2_T_NT	BOOLEAN	PIXIT	both T and NT supported for the rate TSPX_BS_81_A_ur2 of BS81. default value : FALSE, not support both transparent and non transparent modes
TSPX_TS_61_T_NT	BOOLEAN	PIXIT	both T and NT supported for TS61(fax). default value : FALSE, not support both transparent and non transparent modes
TSPX_TS_61_ce	B_2	PIXIT	if the answer to TSPX_TS_61_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent
TSPX_TS_61_ur1	B_4	PIXIT	user rate for TS61(fax). default value : '0101'B, 9.6 kbit/s
TSPX_TS_61_more_ur	BOOLEAN	PIXIT	more than one user rate supported for TS61. default value : FALSE, only one user rate
TSPX_TS_61_ur2	B_4	PIXIT	another user rate for TS61, it shall be different from the previous one if the answer to TSPX_TS_61_more_ur is TRUE.

			default value : '0101'B, 9.6 kbit/s
TSPX_TS_62_T_NT	BOOLEAN	PIXIT	both T and NT supported for TS62. default value : FALSE, not support both transparent and non transparent modes
TSPX_TS_62_ce	B_2	PIXIT	if the answer to TSPX_TS_62_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_TS_62_ur1	B_4	PIXIT	user rate for TS62. default value : '0100'B, 4.8 kbit/s
TSPX_TS_62_more_ur	BOOLEAN	PIXIT	more than one user rate supported by TS62. default value : FALSE, only one user rate
TSPX_TS_62_ur2	B_4	PIXIT	another user rate for TS62, it shall be different from the previous one if the answer to TSPX_TS_62_more_ur is TRUE. default value : '0100'B, 4.8 kbit/s
TSPX_DTMF	B_1	PIXIT	call control capabilities: value '1'B means MS supports DTMF
TSPX_modF	CHMOD_VAL	PIXIT	any non signalling full rate channel mode for TC_26_6_4_1
TSPX_modH	CHMOD_VAL	PIXIT	any non signalling half channel mode for TC_26_6_4_1
TSPX_anymod	CHMOD_VAL	PIXIT	channel mode for TC_62_10_2_4_2. Any supported value except signalling and Full rate speech. possible values are '00000011'B' 12k data, '00001011'B 6k data, '0010011'B 3.6k data
TSPX_CKSNA	CKSN	PIXIT	cipher key sequence number
TSPX_CKSNB	CKSN	PIXIT	cipher key sequence number
TSPX_CKSNC	CKSN	PIXIT	cipher key sequence number
TSPX_CKSNDef	CKSN	PIXIT	default cipher key sequence number
TSPX_RfPwrCap	B_3	PIXIT	RF power capability
TSPX_RevLevel	B_2	PIXIT	Revision level of classmark 1
TSPX_CiphAlgA5_1	B_1	PIXIT	A5/1 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_2	B_1	PIXIT	A5/2 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_3	B_1	PIXIT	A5/3 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_4	B_1	PIXIT	A5/4 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_5	B_1	PIXIT	A5/5 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_6	B_1	PIXIT	A5/6 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_7	B_1	PIXIT	A5/7 algorithm: '1'B : available, '0'B: not available
TSPX_pSyncCap	B_1	PIXIT	pseudo synchronisation capability, '0'B: not present '1'B: present
TSPX_SSscrnInd	B_2	PIXIT	SS screen indicator
TSPX_SMCap	B_1	PXIIT	short message capability, '0'B: not present '1'B: present
TSPX_frqCap	B_1	PIXIT	frequency capability. DCS: reserved, GSM: '0'B not support extension band, '1'B support extension band
TSPX_ClsMk3	B_1	PIXIT	classmark 3 indicator, '0'B no addition capability info, '1'B additional capability info in class mark 3
TSPX_RfPwrCapAmp	B_3	PIXIT	RF power capability with external RF amplifier
TSPX_CphAlgA	CPHALG	PIXIT	any supported ciphering algorithm

TSPX_CphAlgB	CPHALG	PIXIT	ciphering algorithm supported
TSPX_CphAlgC	CPHALG	PIXIT	ciphering algorithm supported
TSPX_CphAlgD	CPHALG	PIXIT	ciphering algorithm supported
TSPX_CphAlgE	CPHALG	PIXIT	ciphering algorithm supported
TSPX_CphAlgDef	CPHALG	PIXIT	default ciphering algorithm
TSPX_DTMFInd	BOOLEAN	PIXIT	DTMF indication to user supported
TSPX_NoOfHoAccA	INTEGER	PIXIT	number of handover access bursts, value range 10 - 20
TSPX_NoOfHoAccB	INTEGER	PIXIT	number of handover access bursts, value range 10 - 20
TSPX_NoOfHoAccC	INTEGER	PIXIT	number of handover access bursts, value range 10 - 20
TSPX_NoOfHoAccD	INTEGER	PIXIT	number of handover access bursts, value range 5 - 10
TSPX_NoOfHoAccE	INTEGER	PIXIT	number of handover access bursts, value range 5 - 10
TSPX_NoOfHoAccF	INTEGER	PIXIT	number of handover access bursts, value range 5 - 10
TSPX_NoOfHoAccG	INTEGER	PIXIT	number of handover access bursts, value range 2 - 5
TSPX_NoOfHoAccH	INTEGER	PIXIT	number of handover access bursts, value range 2 - 5
TSPX_NoOfHoAccI	INTEGER	PIXIT	number of handover access bursts, value range 2 - 5
TSPX_HoRefA	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefB	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefC	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefD	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefE	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefF	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefG	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefH	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefI	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefJ	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HSN	INTEGER	PIXIT	Hopping sequence number value range: 0 - 63
TSPX_IMSI	HEXSTRING	PIXIT	IMSI of the MS
TSPX_IMEI	HEXSTRING	PIXIT	IMEI of the MS.
TSPX_IMEISV	HEXSTRING	PIXIT	IMEISV of the MS. Used in: TC_26_6_8_5, TC_26_7_3_1
TSPX_Ki	BITSTRING	PIXIT	default authentication key used in testing
TSPX_MAIO	INTEGER	PIXIT	mobile allocation index offset, value range: 0 - 63
TSPX_MaxRetrans	INTEGER	PIXIT	Max-Retrans
TSPX_MSTxpwrMax	MAXTXPOW	PIXIT	maximum output power from MS
TSPX_PwrlvlA	INTEGER	PIXIT	power level, value between 2 to 15
TSPX_PwrlvlB	INTEGER	PIXIT	power level, value between 2 to 15
TSPX_PwrlvlC	INTEGER	PIXIT	power level, value between 2 to 15
TSPX_PwrlvlD	INTEGER	PIXIT	power level, value between 2 to 15
TSPX_RANDA	RAND	PIXIT	challenge RAND
TSPX_RANDB	RAND	PIXIT	challenge RAND

TSPX_RANDC	RAND	PIXIT	challenge RAND
TSPX_RANDDef	RAND	PIXIT	default challenge RAND
TSPX_SDCCH4SubA	B_2	PIXIT	TDMA offset of SDCCH/4 subchannel
TSPX_SDCCH4SubB	B_2	PIXIT	TDMA offset of SDCCH/4 subchannel
TSPX_SDCCH4SubC	B_2	PIXIT	TDMA offset of SDCCH/4 subchannel
TSPX_SDCCH4SubDef	B_2	PIXIT	TDMA offset of default SDCCH/4 subchannel
TSPX_SDCCH8SubA	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubB	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubC	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubD	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubE	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubF	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubG	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubDef	B_3	PIXIT	TDMA offset of default SDCCH/8 subchannel
TSPX_TC1M	INTEGER	PIXIT	timer value for GSM timer TC1M (for SMS)
TSPX_MaxCPDataRetx	INTEGER	PIXIT	max. number of CP data retransmissions for SMS
TSPX_Immconn	BOOLEAN	PIXIT	Immediate connect supported
TSPX_TimadvA	INTEGER	PIXIT	timing advance
TSPX_TimadvB	INTEGER	PIXIT	timing advance
TSPX_TimadvC	INTEGER	PIXIT	timing advance
TSPX_TmSltA	SN	PIXIT	time slot
TSPX_TmSltB	SN	PIXIT	time slot
TSPX_TmSltC	SN	PIXIT	time slot
TSPX_TmSltD	SN	PIXIT	time slot
TSPX_TmSltE	SN	PIXIT	time slot
TSPX_TmSltF	SN	PIXIT	time slot
TSPX_TmSltG	SN	PIXIT	time slot
TSPX_TmSltDef	SN	PIXIT	default time slot
TSPX_TmSltNotZero	SN	PIXIT	time slot, arbitrarily value, but not zero.
TSPX_TmSltNotZero1	SN	PIXIT	time slot, arbitrarily value, but not zero and not TSPX_TmSltNotZero
TSPX_Txint	INTEGER	PIXIT	Tx-Integer
TSPX_TscA	TSC	PIXIT	training sequence code
TSPX_TscB	TSC	PIXIT	training sequence code
TSPX_TscC	TSC	PIXIT	training sequence code
TSPX_TscD	TSC	PIXIT	training sequence code
TSPX_TscE	TSC	PIXIT	training sequence code
TSPX_TscF	TSC	PIXIT	training sequence code
TSPX_TscG	TSC	PIXIT	training sequence code
TSPX_TscDef	TSC	PIXIT	default TSC
TSPX_T3122	WI	PIXIT	value of timer T3122, HEXSTRING[2]
TSPX_TCHcarrierA_ho	INTEGER	PIXIT	the value can be choosen arbitrarily from cell allocation of cell A (GSM), but not BCCH carrier .possible value is : 10, 17, 26, 34, 42, 45, 46, 52, 59, 66, 73, 74, 75, 76, 108, 114
TSPX_TCHcarrierA_hod	INTEGER	PIXIT	the value can be choosen arbitrarily from cell allocation of cell A (DCS), but not BCCH carrier . possible value is : 734,741,754,759,762,766,767,773,775,779,782,791,798,829,832,8

			44
TSPX_TCHcarrierB	INTEGER	PIXIT	TCH and SDCCH channel frequency number of cell B (GSM)
TSPX_TCHcarrierBd	INTEGER	PIXIT	TCH and SDCCH channel frequency number of cell B (DCS)
TSPX_TCHcarrierB_ho	INTEGER	PIXIT	the value can be choosen arbitrarily from cell allocation of cell B (GSM), but not BCCH carrier . possibles value is : 14,18,22,24,30,31,38,60,66,73,74 ,75,76,108,114
TSPX_TCHcarrierB_hod	INTEGER	PIXIT	not BCCH carrier of cell B. the value can be choosen arbitrarily from cell allocation of cell B (DCS), which is (739,743,746,749,756,758,761,77 1,779,782,791,798,829,832,844)
TSPX_TCHcarrierB2_ho	INTEGER	PIXIT	Choosen arbitrarly from cell allocation B for GSM HO cases, but not BCCH carrier!
TSPX_TCHcarrierB2_hod	INTEGER	PIXIT	Choosen arbitrarly from cell allocation B for DCS HO cases, but not BCCH carrier!
TSPX_TCHHSubA	B_1	PIXIT	TDMA offset of half rate subchannel
TSPX_TCHHSubDef	B_1	PIXIT	TDMA offset of default half rate subchannel
TSPX_TMSI	TMSI_V	PIXIT	TMSI of the MS used in test
TSPX_TMSI1	TMSI_V	PIXIT	another TMSI used in test which shall differ from TSPX_TMSI, TSPX_TMSI + '01'O TSPX_TMSI + '02'O TSPX_TMSI + '03'O
TSPX_Uupd	B_8	PIXIT	user-user protocol discriminator
TSPX_UuInfo	OCTETSTRING	PIXIT	user-user information
TSPX_k	INTEGER	PIXIT	timing of cell A before cell B k bit periods for TC_26_6_5_1, TC_26_6_5_2_? TC_26_6_5_3_?, TC_26_6_5_4_?.
TSPX_y	INTEGER	PIXIT	timing advance for TC_26_6_5_3_?, TC_26_6_5_4_?, TC_26_6_5_5_? . The values of TSPX_k (or TSPX_k1) and TSPX_y: 0 < (2*TSPX_k + TSPX_y) MOD 256 < 60.
TSPX_k1	INTEGER	PIXIT	timing of cell A before cell B k bit periods for TC_26_6_5_5_?
TSPX_k2	INTEGER	PIXIT	timing of cell A before cell B k bit periods for TC_26_6_5_6
TSPX_y2	INTEGER	PIXIT	timing advance for TC_26_6_5_6, value range : 11 - 62.
TSPX_k3	INTEGER	PIXIT	timing of cell A before cell B k bit periods for TC_26_6_5_7
TSPX_y3	INTEGER	PIXIT	timing advance for TC_26_6_5_7
TSPX_nPara	INTEGER	PIXIT	the n'th ChReq for TC_26_6_1_2, shall be choose to [1..8], MAXRETRANS for TC_26_6_1_2 is 7 (See 11.10 for more explaination).
TSPX_kPara	INTEGER	PIXIT	the k'th ChReq for TC_26_6_1_2
TSPX_rPara	INTEGER	PIXIT	the r'th ChReq for TC_26_6_1_2
TSPX_i1Para	INTEGER	PIXIT	for TC_26_1_2
TSPX_i2Para	INTEGER	PIXIT	for TC_26_1_2
TSPX_i3Para	INTEGER	PIXIT	the r'th ChReq for TC_26_6_1_2
TSPX_n1Para	INTEGER	PIXIT	the n'th ChReq for TC_26_6_1_3
TSPX_i4Para	INTEGER	PIXIT	for TC_26_6_1_3
TSPX_xPara	INTEGER	PIXIT	t3122 for TC_26_6_1_3
TSPX_AGBLKS1	INTEGER	PIXIT	BS-AG-BLKS-RES for TC_26_6_2_3_1, TC_26_6_2_1_1

TSPX_PAMFRMS1	INTEGER	PIXIT	BS-PA-MFRMS for TC_26_6_2_3_1 (shall not be set to 9), TC_26_6_2_1_1
TSPX_CcchConf1	CCCH_CON	PIXIT	paging subchannel for TC_26_6_2_3_1, TC_26_6_2_1_1
TSPX_PgSubch	INTEGER	PIXIT	paging subchannel for TC_26_6_2_3_1
TSPX_AGBLKS2	INTEGER	PIXIT	BS-AG-BLKS-RES for TC_26_6_2_3_2, TC_26_6_2_1_2
TSPX_PAMFRMS2	INTEGER	PIXIT	BS-PA-MFRMS for TC_26_6_2_3_2, TC_26_6_2_1_2
TSPX_CcchConf2	CCCH_CON	PIXIT	paging subchannel for TC_26_6_2_3_2, TC_26_6_2_1_2
TSPX_AGBLKS3	INTEGER	PIXIT	BS-AG-BLKS-RES for TC_26_6_2_4, TC_26_6_2_2, TC_26_6_2_1_3
TSPX_PAMFRMS3	INTEGER	PIXIT	BS-PA-MFRMS for TC_26_6_2_4, TC_26_6_2_2, TC_26_6_2_1_3
TSPX_CcchConf3	CCCH_CON	PIXIT	paging subchannel for TC_26_6_2_4, TC_26_6_2_2, TC_26_6_2_1_3
TSPX_AGBLKS4	INTEGER	PIXIT	BS-AG-BLKS-RES for TC_26_6_2_5
TSPX_PAMFRMS4	INTEGER	PIXIT	BS-PA-MFRMS for TC_26_6_2_5
TSPX_CcchConf4	CCCH_CON	PIXIT	paging subchannel for TC_26_6_2_5 (shall be in the set ('010', '100', '110'))
TSPX_Chtp1	CH_TDMA	PIXIT	channel type for TC_26_6_13_1, any value supported by the MS
TSPX_ChMod1	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_1, any value for the channel type of TSPX_Chtp1
TSPX_Maio1	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma1, default : 2
TSPX_Ma1	MAC	PIXIT	mobile allocation for TC_26_6_13_1, its value shall indicate number of frequencies between 1 and 17. default : '010101'O
TSPX_Hsn1	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1. default: 1
TSPX_Maio2	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma2, default : 8
TSPX_Ma2	MAC	PIXIT	mobile allocation for TC_26_6_13_1, its value shall indicate number of frequencies between 1 and 17. default :'01362A5'O
TSPX_Hsn2	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, default : 6
TSPX_Maio3	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma3, default : 14
TSPX_Ma3	MAC	PIXIT	mobile allocation for TC_26_6_13_1, its value shall indicate number of frequencies between 1 and 17. default :'01FFFA'O
TSPX_Hsn3	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, default : 0

TSPX_Chtp2	CH_TDMA	PIXIT	channel type for TC_26_6_13_2, any value supported by the MS
TSPX_ChMod2	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_2
TSPX_Maio4	INTEGER	PIXIT	hopping parameter for TC_26_6_13_2, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma4, default : 3
TSPX_Ma4	MAC	PIXIT	mobile allocation for TC_26_6_13_2, its value shall indicate number of frequencies between 1 and 17. default : '001141'O
TSPX_Hsn4	INTEGER	PIXIT	hopping parameter for TC_26_6_13_2, default : 0
TSPX_Maio5	INTEGER	PIXIT	hopping parameter for TC_26_6_13_2, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma5, default : 9
TSPX_Ma5	MAC	PIXIT	mobile allocation for TC_26_6_13_2, its value shall indicate number of frequencies between 1 and 17. default : '01ABCB'O
TSPX_Hsn5	INTEGER	PIXIT	hopping parameter for TC_26_6_13_2, default : 16
TSPX_Chtp3	CH_TDMA	PIXIT	channel type for TC_26_6_13_3, any value supported by the MS
TSPX_Maio6	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma6, default : 4
TSPX_Ma6	MAC	PIXIT	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 1 and 17. default :'006248'O
TSPX_Hsn6	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, default : 4
TSPX_Maio7	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma7, default : 10
TSPX_Ma7	MAC	PIXIT	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 2 and 17. default : '009AFB'O
TSPX_Chtp4	CH_TDMA	PIXIT	channel type for TC_26_6_13_3, any value supported by the MS
TSPX_Maio8	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma8, default : 1
TSPX_Ma8	MAC	PIXIT	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 2 and 17. default : '002800'O
TSPX_Hsn8	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, default : 40
TSPX_Maio9	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma9, default : 15
TSPX_Ma9	MAC	PIXIT	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 2 and 17. default :

			'01FFFB'O
TSPX_Hsn9	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, default : 8
TSPX_Chtp5	CH_TDMA	PIXIT	SDCCH8 subchannel immediately assigned, TC_26_6_13_4
TSPX_Maio10	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma10, default : 5
TSPX_Ma10	MAC	PIXIT	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 1 and 17. default : '01D082'O
TSPX_Hsn10	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, default: 0
TSPX_Maio11	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma11, default : 11
TSPX_Ma11	MAC	PIXIT	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 2 and 17. default : '00CFF3'O
TSPX_Chtp6	CH_TDMA	PIXIT	channel type of a non- existing channel in Assignment Cmd, TC_26_6_13_4
TSPX_Maio12	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma12, default : 1
TSPX_Ma12	MAC	PIXIT	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 1 and 17. default : '010100'O
TSPX_Hsn12	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, default : 39
TSPX_Maio13	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma13, default : 6
TSPX_Ma13	MAC	PIXIT	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 1 and 17. default : '00E690'O
TSPX_Hsn13	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, default: 42
TSPX_Chtp7	CH_TDMA	PIXIT	channel type for TC_26_6_13_5
TSPX_ChMod4	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_5
TSPX_Maio14	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma14, default : 5
TSPX_Ma14	MAC	PIXIT	mobile allocation for TC_26_6_13_5, its value shall indicate number of frequencies between 1 and 17. default : '014A28'O
TSPX_Hsn14	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, default : 50
TSPX_Maio15	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma15, default : 12
TSPX_Ma15	MAC	PIXIT	mobile allocation for

			TC_26_6_13_5, its value shall indicate number of frequencies between 1 and 16. default : '00FF79'O
TSPX_Hsn15	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, default : 33
TSPX_Maio16	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma16, default : 2
TSPX_Ma16	MAC	PIXIT	mobile allocation for TC_26_6_13_5, its value shall indicate number of frequencies between 1 and 16. default : '001110'O
TSPX_Hsn16	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, default : 21
TSPX_Chtp8	CH_TDMA	PIXIT	channel type for TC_26_6_13_6
TSPX_ChMod5	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_6
TSPX_Maio17	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma17, default : 7
TSPX_Ma17	MAC	PIXIT	mobile allocation for TC_26_6_13_6, its value shall indicate number of frequencies between 1 and 17. default : '00E6A1'O
TSPX_Hsn17	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, default : 22
TSPX_Maio18	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma18, default : 13
TSPX_Ma18	MAC	PIXIT	mobile allocation for TC_26_6_13_6, its value shall indicate number of frequencies between 1 and 16. default : '00FFF9'O
TSPX_Hsn18	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, default : 9
TSPX_Maio19	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma19, default : 3
TSPX_Ma19	MAC	PIXIT	mobile allocation for TC_26_6_13_6, its value shall indicate number of frequencies between 1 and 16. default : '001111'O
TSPX_Hsn19	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, default : 44
TSPX_Chtp9	CH_TDMA	PIXIT	channel type for TC_26_6_13_7
TSPX_ChMod6	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_7
TSPX_Maio20	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma20, default : 8
TSPX_Ma20	MAC	PIXIT	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 17. default : '016699'O
TSPX_Hsn20	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, default : 30
TSPX_Maio21	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in

			TSPX_Ma21, default : 14
TSPX_Ma21	MAC	PIXIT	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 17. default : '01FEFB'O
TSPX_Chtp10	CH_TDMA	PIXIT	channel type for TC_26_6_13_7
TSPX_Maio22	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma22, default : 4
TSPX_Ma22	MAC	PIXIT	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 16. default : '006241'O
TSPX_Hsn22	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, default : 11
TSPX_Maio23	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma23, default : 11
TSPX_Ma23	MAC	PIXIT	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 16. default : '00FF1B'O
TSPX_Hsn23	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, default : 60
TSPX_Chtp11	CH_TDMA	PIXIT	SDCCH8 subchannel immediately assigned, TC_26_6_13_8
TSPX_ChMod7	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_8
TSPX_Maio24	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma24, default : 9
TSPX_Ma24	MAC	PIXIT	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 17. default : '00E6E9'O
TSPX_Hsn24	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, default : 7
TSPX_Maio25	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma25, default : 15
TSPX_Ma25	MAC	PIXIT	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 17. default : '01FFFB'O
TSPX_Chtp12	CH_TDMA	PIXIT	channel type of a non- existing channel in Assignment Cmd, TC_26_6_13_8
TSPX_Maio26	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma26, default : 5
TSPX_Ma26	MAC	PIXIT	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 16. default : '009168'O
TSPX_Hsn26	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, default : 9
TSPX_Maio27	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies)

			-1, which is defined in TSPX_Ma27, default : 1
TSPX_Ma27	MAC	PIXIT	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 16. default : '004080'O
TSPX_Hsn27	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, default : 38
TSPX_Chtp13	CH_TDMA	PIXIT	channel type for TC_26_6_13_9
TSPX_Tm3	INTEGER	PIXIT	channel mode for TC_26_6_13_9, value between 60 -100
TSPX_Maio28	INTEGER	PIXIT	hopping parameter for TC_26_6_13_9, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma28, default : 10
TSPX_Ma28	MAC	PIXIT	mobile allocation for TC_26_6_13_9, its value shall indicate number of frequencies between 1 and 17. default : '00EE7A'O
TSPX_Hsn28	INTEGER	PIXIT	hopping parameter for TC_26_6_13_9, default : 11
TSPX_Maio29	INTEGER	PIXIT	hopping parameter for TC_26_6_13_9, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma29, default : 6
TSPX_Ma29	MAC	PIXIT	mobile allocation for TC_26_6_13_9, its value shall indicate number of frequencies between 1 and 17. default : '00F070'O
TSPX_Chtp14	CH_TDMA	PIXIT	channel type for TC_26_6_13_10
TSPX_Maio30	INTEGER	PIXIT	hopping parameter for TC_26_6_13_10, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma30, default : 11
TSPX_Ma30	MAC	PIXIT	mobile allocation for TC_26_6_13_10, its value shall indicate number of frequencies between 1 and 17. default : '017F69'O
TSPX_Hsn30	INTEGER	PIXIT	hopping parameter for TC_26_6_13_10, default : 62
TSPX_Maio31	INTEGER	PIXIT	hopping parameter for TC_26_6_13_10, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma31, default : 1
TSPX_Ma31	MAC	PIXIT	mobile allocation for TC_26_6_13_10, its value shall indicate number of frequencies between 1 and 17. default : '000101'O
TSPX_MOBscSvcA	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcB	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcC	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcD	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcE	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcF	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcG	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcH	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcI	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcJ	SERVICES	PIXIT	any supported MO basic service
TSPX_MT_DualModSvc	SERVICES	PIXIT	any supported MT dual mode service
TSPX_MT_NonSptSvc	SERVICES	PIXIT	any not supported MT basic service
TSPX_MTBscSvcA	SERVICES	PIXIT	any supported MT basic service

TSPX_MTBscSvcB	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcC	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcD	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcE	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcF	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcG	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcH	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcI	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcJ	SERVICES	PIXIT	any supported MT basic service
TSPX_MTNIC_BscSvcA	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcB	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcC	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcD	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcE	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcF	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcG	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcH	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcI	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcJ	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_Telephony_Rate	RATE	PIXIT	channel rate for TS11, default value : C_Full
TSPX_MOChRateA	RATE	PIXIT	channel rate for TSPX_MOBscSvcA, default value : C_Full
TSPX_MOChRateB	RATE	PIXIT	channel rate for TSPX_MOBscSvcB, default value : C_Full
TSPX_MOChRateC	RATE	PIXIT	channel rate for TSPX_MOBscSvcC, default value : C_Full
TSPX_MOChRateD	RATE	PIXIT	channel rate for TSPX_MOBscSvcD, default value : C_Full
TSPX_MOChRateE	RATE	PIXIT	channel rate for TSPX_MOBscSvcE, default value : C_Full
TSPX_MOChRateF	RATE	PIXIT	channel rate for TSPX_MOBscSvcF, default value : C_Full
TSPX_MOChRateG	RATE	PIXIT	channel rate for TSPX_MOBscSvcG, default value : C_Full
TSPX_MOChRateH	RATE	PIXIT	channel rate for TSPX_MOBscSvcH, default value : C_Full
TSPX_MOChRateI	RATE	PIXIT	channel rate for TSPX_MOBscSvcI, default value : C_Full
TSPX_MOChRateJ	RATE	PIXIT	channel rate for TSPX_MOBscSvcJ, default value : C_Full
TSPX_MTChRateA	RATE	PIXIT	channel rate for TSPX_MTBscSvcA, default value : C_Full
TSPX_MTChRateB	RATE	PIXIT	channel rate for TSPX_MTBscSvcB, default value : C_Full
TSPX_MTChRateC	RATE	PIXIT	channel rate for TSPX_MTBscSvcC, default value : C_Full
TSPX_MTChRateD	RATE	PIXIT	channel rate for

			TSPX_MTBscSvcD, default value : C_Full
TSPX_MTChRateE	RATE	PIXIT	channel rate for TSPX_MTBscSvcE, default value : C_Full
TSPX_MTChRateF	RATE	PIXIT	channel rate for TSPX_MTBscSvcF, default value : C_Full
TSPX_MTChRateG	RATE	PIXIT	channel rate for TSPX_MTBscSvcG, default value : C_Full
TSPX_MTChRateH	RATE	PIXIT	channel rate for TSPX_MTBscSvcH, default value : C_Full
TSPX_MTChRateI	RATE	PIXIT	channel rate for TSPX_MTBscSvcI, default value : C_Full
TSPX_MTChRateJ	RATE	PIXIT	channel rate for TSPX_MTBscSvcJ, default value : C_Full
TSPX_MTNIC_ChRateA	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcA, default value : C_Full
TSPX_MTNIC_ChRateB	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcB, default value : C_Full
TSPX_MTNIC_ChRateC	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcC, default value : C_Full
TSPX_MTNIC_ChRateD	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcD, default value : C_Full
TSPX_MTNIC_ChRateE	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcE, default value : C_Full
TSPX_MTNIC_ChRateF	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcF, default value : C_Full
TSPX_MTNIC_ChRateG	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcG, default value : C_Full
TSPX_MTNIC_ChRateH	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcH, default value : C_Full
TSPX_MTNIC_ChRateI	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcI, default value : C_Full
TSPX_MTNIC_ChRateJ	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcJ, default value : C_Full
TSPX_EmgCallRate	RATE	PIXIT	Rate for the basic service supported for MO emergency calls
TSPX_MO_NonCallSS	SERVICES	PIXIT	any supported MO non-call releated supplementary service
TSPX_MO_BscSvc_SMS	SERVICES	PIXIT	any supported MO SMS calls
TSPX_MO_DualModSvc	SERVICES	PIXIT	any supported MO dual mode call
TSPX_MO_DualModRate	RATE	PIXIT	Rate for the supported MO dual mode call in TSPX_MO_DualModSvc
Detailed Comments:			

Test case selection expression definitions

Test Case Selection Expression Definitions		
Expression Name	**Selection Expression**	**Comments**
SelExpr_0000	TRUE	General test group always selected
SelExpr_0001	TRUE	always selected
SelExpr_0002	TSPC_MTsvc	MT circuit switched basic service supported
SelExpr_0003	NOT TSPC_Serv_SS_AoCC	AOCC not supported
SelExpr_0004	TSPC_Serv_SS_AoCC AND(NOT TSPC_Serv_SS_HOLD)	AOCC supported but Call Hold not supported
SelExpr_0005	TSPC_Serv_SS_AoCC AND TSPC_Serv_SS_HOLD AND(NOT TSPC_Serv_SS_MPTY)	AOCC and Call Hold supported but multiparty not supported
SelExpr_0006	NOT TSPC_Feat_FND	FND feature not supported
SelExpr_0007	TSPC_MOsvc	MO circuit switched basic service supported
SelExpr_0100	TRUE	initial test group always selected
SelExpr_0101	TRUE	always selected
SelExpr_0102	TSPC_SvcOnTCH	at least one service on traffic channel supported
SelExpr_0103	TSPC_DualRate	half rate channel supported
SelExpr_0104	TSPC_Serv_BS21 OR TSPC_Serv_BS22 OR TSPC_Serv_BS23 OR TSPC_Serv_BS24 OR TSPC_Serv_BS25 OR TSPC_Serv_BS26 OR TSPC_Serv_BS31 OR TSPC_Serv_BS32 OR TSPC_Serv_BS33 OR TSPC_Serv_BS34 OR TSPC_Serv_BS41 OR TSPC_Serv_BS42 OR TSPC_Serv_BS43 OR TSPC_Serv_BS44 OR TSPC_Serv_BS45 OR TSPC_Serv_BS46 OR TSPC_Serv_BS51 OR TSPC_Serv_BS52 OR TSPC_Serv_BS53 OR TSPC_Serv_BS61 OR TSPC_Serv_BS81 OR TSPC_Serv_TS61 OR TSPC_Serv_TS62	at least one data service supported
SelExpr_0105	TSPC_NonCallSS	non call related supplementary service supported
SelExpr_0106	TSPC_Serv_TS22	MO short message service supported
SelExpr_0107	TSPC_Serv_TS11 OR TSPC_Serv_TS12	speech supported
SelExpr_0200	TRUE	idleMode test group always selected
SelExpr_0201	TRUE	always selected
SelExpr_0300	TRUE	BiBo test group always selected
SelExpr_0301	TSPC_Serv_TS11 OR TSPC_Serv_TS12 OR SelExpr_0104	CC protocol for at least one BC supported
SelExpr_0302	TRUE	always selected
SelExpr_0400	TRUE	RR test group always selected
SelExpr_0401	TRUE	always selected
SelExpr_0402	SelExpr_0301	CC protocol for at least one BC supported
SelExpr_0403	SelExpr_0301	Full rate traffic channel supported
SelExpr_0404	TSPC_DualRate	Dual rate traffic channel supported
SelExpr_0405	SelExpr_0301 AND(TSPC_Feat_A51 OR TSPC_Feat_A52)	call control protocol and (A5/1 or A5/2) algorithm supported
SelExpr_0406	SelExpr_0301 AND TSPC_RFAmp	call control protocol and RF amplification supported
SelExpr_0407	SelExpr_0301	call control protocol and full rate traffic channel supported
SelExpr_0408	SelExpr_0301 AND TSPC_DualRate	call control protocol and dual rate traffic channel supported
SelExpr_0409	SelExpr_0301 AND TSPC_AddInfo_PseudoSynch	CC protocol for at least one BC supported and Pseudo synchronised supported
SelExpr_0410	TSPC_MOsvc	at least one MO circuit switched basic service supported
SelExpr_0500	TRUE	MM test group always selected
SelExpr_0501	TRUE	always selected
SelExpr_0502	TSPC_SIMRmv	SIM removable without power down supported
SelExpr_0503	TSPC_Serv_TS11 OR TSPC_Serv_TS12	speech supported

SelExpr_0504	TSPC_NonCallSS	The MS supports a non call related supplementary service operation during an active call
SelExpr_0600	TSPC_MOsvc OR TSPC_MTsvc	CC test group
SelExpr_0601	TSPC_MOsvc AND(NOT TSPC_EmgOnly)	at least one mobile originating circuit switched basic service supported and not only emergency call supported
SelExpr_0602	(TSPC_NoimmConn) AND TSPC_MTsvc	immediate connect not supported and at least one mobile terminated circuit switched basic service supported
SelExpr_0603	TSPC_RefusalCall AND TSPC_MTsvc	refusal of call supported and at least one mobile terminated circuit switched basic service supported
SelExpr_0604	TSPC_Serv_TS11 OR TSPC_Serv_TS12 OR TSPC_Serv_TS61 OR TSPC_Serv_BS61 OR TSPC_Serv_BS81	speech supported
SelExpr_0605	SelExpr_0604 AND TSPC_AlertInd	speech & alerting supported
SelExpr_0606	TSPC_MTsvc	at least one mobile terminated circuit switched basic service supported
SelExpr_0607	TSPC_Serv_TS61 OR TSPC_Serv_BS61 OR TSPC_Serv_BS81	dual mode services supported
SelExpr_0700	SelExpr_0301	StrcutureProc test group
SelExpr_0701	TSPC_Serv_TS11 OR TSPC_Serv_TS62	at least one teleservice (except emergency call and dual service) supported
SelExpr_0702	TSPC_Serv_TS12	emergency call supported
SelExpr_0703	TSPC_Serv_TS12 AND TSPC_DualRate	dual rate emergency call supported
SelExpr_0800	TSPC_SS	SS test group
SelExpr_0801	TSPC_Serv_SS_BOIC OR TSPC_Serv_SS_BAIC OR TSPC_Serv_SS_BOICexHC OR TSPC_Serv_SS_BICRoam OR TSPC_Serv_SS_BAOC	at least one of call restrictions supported
SelExpr_0802	TSPC_Serv_SS_AoCC	AOCC supported
SelExpr_0803	TSPC_Serv_SS_BOIC	BOIC supported
SelExpr_0804	TSPC_Serv_SS_BAIC	BAIC supported
SelExpr_0805	TSPC_Serv_SS_BOICexHC	BOICExHC supported
SelExpr_0806	TSPC_Serv_SS_BOICexHC OR TSPC_Serv_SS_BAIC	BOICExHC OR BAIC supported
SelExpr_0807	TSPC_Serv_SS_BOIC OR TSPC_Serv_SS_BICRoam	BOIC OR BICRoam supported
SelExpr_0808	TSPC_Serv_SS_BI	BI supported
SelExpr_0809	TSPC_Serv_SS_CFNRy OR TSPC_Serv_SS_CFU	CFNNRy or CFU supported
SelExpr_0810	TSPC_Serv_SS_CFB OR TSPC_Serv_SS_CFU OR TSPC_Serv_SS_CFNRc OR TSPC_Serv_SS_CFNRy	CFB or CFU or CFNRy or CFNRc supported
SelExpr_0811	TSPC_Serv_SS_CFB OR TSPC_Serv_SS_CFNRc OR TSPC_Serv_SS_CFNRy	CFNRy or CFNRc or CFB supported
SelExpr_0812	TSPC_Serv_SS_CFNRc OR TSPC_Serv_SS_CFB	CFNNRc or CFB supported
SelExpr_0813	TSPC_Serv_SS_CFB	CFB supported
SelExpr_0814	TSPC_Serv_SS_BICRoam OR TSPC_Serv_SS_BAOC	BAOC or BICRoam supported
SelExpr_0815	TRUE	always selected
SelExpr_0816	TSPC_Serv_SS_unstruct	USSD supported
SelExpr_0817	TSPC_Serv_SS_unstruct AND SelExpr_0301	USSD and CC protocol for at least one BC supported
SelExpr_0818	TSPC_Serv_SS_AoCC AND TSPC_SIMRmv	AOCC and SIM removable without power down supported
SelExpr_0819	TSPC_Serv_SS_AoCC AND TSPC_Serv_SS_HOLD	AOCC and Call Hold supported
SelExpr_0820	TSPC_Serv_SS_AoCC AND TSPC_Serv_SS_MPTY	AOCC and Multi Party service supported
SelExpr_0821	TSPC_Serv_SS_AoCC AND TSPC_SwitchOnOff	AOCC and switch on/off supported
SelExpr_0900	TSPC_SMS	SMS test group
SelExpr_0901	TSPC_Serv_TS21 AND SelExpr_0301	MT/PP supported and CC protocol for at least one BC supported

SelExpr_0902	TSPC_Serv_TS22 AND TSPC_Serv_TS21 AND SelExpr_0301	MO/PP supported and MT/PP supported and CC protocol for at least one BC supported
SelExpr_0903	TSPC_Serv_TS22 AND TSPC_StoreRcvSMSME AND TSPC_StoreRcvSMSSIM	MO/PP supported AND storage of SMS in the ME supported AND storage of SMS in the SIM supported
SelExpr_0904	TSPC_Serv_TS22 AND TSPC_Serv_TS21 AND TSPC_SMSStatusRepCap	MO/PP supported AND MT/PP supported AND SMS Status report capabilities supported
SelExpr_0905	TSPC_Serv_TS21 AND TSPC_DispRcvSMS	MT/PP supported AND display of received short message supported
SelExpr_0906	TSPC_Serv_TS21 AND TSPC_DispRcvSMS AND(TSPC_StoreRcvSMSME OR TSPC_StoreRcvSMSSIM)	MT/PP supported AND display of received short message supported AND (storage of SMS in the ME supported OR storage of SMS in the SIM supported)
SelExpr_0907	TSPC_Serv_TS21 AND TSPC_StoreRcvSMSSIM	MT/PP supported AND display of received short message supported AND storage of SMS in the SIM supported
SelExpr_0908	TSPC_Serv_TS21 AND TSPC_ReplaceSMS AND TSPC_DispRcvSMS	MT/PP supported AND "replace short message" and "display of received short message" supported
SelExpr_0909	TSPC_Serv_TS22 AND TSPC_Serv_TS21 AND TSPC_ReplyProc AND TSPC_DispRcvSMS	MO/PP supported AND MT/PP supported AND "reply procedure" and "display of received short message" supported
SelExpr_1000	TSPC_EGSM	E- band supported
SelExpr_1001	SelExpr_0301	CC protocol for at least one BC supported
SelExpr_1002	TSPC_Serv_TS11 OR TSPC_Serv_TS61 OR TSPC_Serv_TS62	telephony supported or alternate speech/data supported or Teleservice automatic G3 fax supported
SelExpr_1003	TSPC_Serv_TS12	emergency call supported
Detailed Comments:		

Test suite constant declarations

Test Suite Constant Declarations			
Constant Name	**Type**	**Value**	**Comments**
C_arfcnA	INTEGER	20	ARFCN 20
C_arfcn_1	INTEGER	1	ARFCN 1
C_arfcn_2	INTEGER	2	ARFCN 2
C_arfcn_1015	INTEGER	1015	ARFCN 1015
C_arfcn_10	INTEGER	10	ARFCN 10
C_arfcn_34	INTEGER	34	ARFCN 34
C_arfcnB	INTEGER	10	ARFCN of cell B
C_arfcnC	INTEGER	30	ARFCN 30
C_arfcnH	INTEGER	44	ARFCN 44
C_arfcnAd	INTEGER	590	ARFCN 590
C_arfcnAd_1	INTEGER	520	ARFCN 520
C_arfcnAd_2	INTEGER	514	ARFCN 514
C_arfcnAd_3	INTEGER	715	ARFCN 715
C_arfcnAd_4	INTEGER	747	ARFCN 747
C_arfcnAd_5	INTEGER	734	ARFCN 734
C_arfcnAd_6	INTEGER	759	ARFCN 759
C_arfcnBd	INTEGER	520	ARFCN 520
C_arfcnCd	INTEGER	700	ARFCN 700
C_arfcnHd	INTEGER	810	ARFCN 810
C_arfcn_tchA	INTEGER	30	ARFCN f30
C_arfcn_tchAd	INTEGER	650	ARFCN 650
C_arfcn_tchB	INTEGER	114	ARFCN 114
C_arfcn_tchBd	INTEGER	844	ARFCN 844
C_arfcnEgsm	INTEGER	990	ARFCN 990
C_arfcnEgsm_iacmd	INTEGER	20	ARFCN 20
C_arfcnEgsm_asscmd	INTEGER	20	ARFCN 20
C_BCCHcarrierB_ho	INTEGER	40	BCCH frequency number of cell B for Ho cases (GSM: 40)
C_BCCHcarrierB_hod	INTEGER	764	BCCH frequency number of cell B for Ho cases (DCS:764)
C_Noarfcn	INTEGER	65534	No arfcn defined (hopping channel)
C_ATT_0	INTEGER	0	Attach value 0
C_ATT_1	INTEGER	1	Attach value 1
C_NCC	NCC	'001'B	Network color code
C_NCC_0	NCC	'000'B	Network color code value '000'B
C_NCC_1	NCC	'001'B	Network color code value '001'B
C_NCC_2	NCC	'010'B	Network color code value '010'B
C_NCC_3	NCC	'011'B	Network color code value '011'B
C_NCC_4	NCC	'100'B	Network color code value '100'B
C_NCC_5	NCC	'101'B	Network color code value '101'B
C_NCC_6	NCC	'110'B	Network color code value '110'B
C_NCC_7	NCC	'111'B	Network color code value '111'B
C_NCCP_2	NCCP	'02'O	Network color code permitted value '02'O
C_NCCP_10	NCCP	'0A'O	Network color code permitted value '0A'O
C_BCC	BCC	'101'B	Base Station color code
C_BCC_0	BCC	'000'B	Base Station color code value 0
C_BCC_1	BCC	'001'B	Base Station color code value 1
C_BCC_3	BCC	'011'B	Base Station color code value 3
C_BCC_5	BCC	'101'B	Base Station color code value 5
C_BCC_7	BCC	'111'B	Base Station color code value 7
C_cch_1nonComb	B_3	'000'B	cch_con = 1 basic ch non combined
C_cch_1Comb	B_3	'001'B	cch_con = 1 basic ch combined
C_cch_2Comb	B_3	'011'B	cch_con = 2 basic ch combined
C_cch_3Comb	B_3	'101'B	cch_con = 3 basic ch combined

C_cch_4Comb	B_3	'111'B	cch_con = 4 basic ch combined
C_cch_2nonComb	B_3	'010'B	cch_con = 2 basic chs non combined
C_cch_3nonComb	B_3	'100'B	cch_con = 3 basic chs non combined
C_cch_4nonComb	B_3	'110'B	cch_con = 4 basic chs non combined
C_cchd_e_407	OCTETSTRING	'8DEA0DF4CC6C4AFAFB10000000000000'O	f-list for cellchdescr using of 128 format. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}. Length = 16 TC_26_10_4, k=2, c=1 TC_26.10.6, c=1
C_cchd_e_408	OCTETSTRING	'8A0A1CFD3EF4610E2FFFFA0000000000'O	f-list for cellchdescr using of 256 format. The coded set is {20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114,115} Length = 16 TC_26_10_4, k=2, c=2 TC_26.10.6, c=2
C_cchd_e_409	OCTETSTRING	'89EA037F433C7B042BFEBFEC10000000'O	f-list for cellchdescr using of 512 format. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}. Length = 16 TC_26_10_4, k=2, c=3 TC: 26.10.6 c=3
C_cchd_e_410	OCTETSTRING	'841EEA893EF98143B160000000000000'O	f-list for cellchdescr using of 1024 format. Length = 16
C_cchd_e_411	OCTETSTRING	'8FEA703E084210000000000000000000'O	f-list for cellchdescr using of variable bit format. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}. Length = 16 TC_26_10_4, k=2, c=5 TC: 26.10.6 c=5
C_cchd_e_412	OCTETSTRING	'00020800000007F020000008000080000'O	f-list for cellchdescr using of bit map 0 format. The coded set is {20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114,115} Length = 16 TC_26_10_4, k=2, c=6
C_cchd_e_414	OCTETSTRING	'00000000000000002000020000020000000'O	f-list for cellchdescr using of bit map 0 format. Length = 16 f_list: 30, 50 ,70 TC: 26.10.6 c=6
C_cchd_e_415	OCTETSTRING	'841EEA893EF981438000000000000000'O	f-list for cellchdescr using range 1024 format. The coded set is {0, 30, 40, 66, 80, 1005, 1010, 1015}. Length = 16 TC_26_10_4, k=2, c=4 TC: 26.10.6 c=4
C_ChMod_s	CHMOD_VAL	'00000000'B	Mode : signalling only.
C_ChMod_r	CHMOD_VAL	'00000001'B	Mode : speech full or half rate.
C_ChMod_12k	CHMOD_VAL	'00000011'B	Mode : 12k radio rate.
C_ChMod_6k	CHMOD_VAL	'00001011'B	Mode : 6k radio rate.
C_ChMod_3k	CHMOD_VAL	'00010011'B	Mode : 3.6k radio rate.
C_Sap0	SAPID	'00'O	service access point 0
C_Sap3	SAPID	'03'O	service access point 3
C_CellA	CellID	"C_CellA"	cell A -- cell 1
C_CellB	CellID	"C_CellB"	cell B -- cell 2
C_CellC	CellID	"C_CellC"	cell C -- cell 3
C_CellD	CellID	"C_CellD"	cell D -- cell 4
C_CellE	CellID	"C_CellE"	cell E -- cell 5

C_CellF	CellID	"C_CellF"	cell F -- cell 6
C_CellG	CellID	"C_CellG"	cell G -- cell 7
C_CellH	CellID	"C_CellH"	cell H -- cell 8
C_ci_cellA	CI	'0001'O	Cell Id for cell A
C_ci_cellB	CI	'0002'O	Cell Id for cell B
C_ci_cellC	CI	'0003'O	Cell Id for cell C
C_ci_cellD	CI	'0004'O	Cell Id for cell D
C_ci_cellE	CI	'0005'O	Cell Id for cell E
C_ci_cellF	CI	'0006'O	Cell Id for cell F
C_ci_cellG	CI	'0007'O	Cell Id for cell G
C_ci_cellH	CI	'0008'O	Cell Id for cell H
C_cksn_nokey	BITSTRING	'111'B	No key available
C_E_default	INTEGER	63	Default field strength
C_E_neighbourdefault	INTEGER	53	Default field strength of neighbouriong cells.
C_E_suitable	INTEGER	45	Field strength of a suitable cell
C_E_notsuitable	INTEGER	-10	Field strength of a not suitable cell for reselection (C1 < 0)
C_E_notsuitable_fSel	INTEGER	24	Field strength of a not suitable cell for selection
C_flist_e_401	OCTETSTRING	'8DF68AEC00'O	It includes the list of f's in EGSM test cases. The coded set is {1005, 1010, 1015}. range 128 format Length = 5 TC_26_10_4, k=1, c=1 TC: 26.10.5.1 k=1/2,c=1
C_flist_e_402	OCTETSTRING	'8A2481FF03F8'O	It includes the list of f's in EGSM test cases. The coded set is {73, 74, 75, 76, 77}, range 256 format. Length = 6 TC_26_10_4, k=1, c=2 TC: 26.10.5.1 k=1/2,c=2
C_flist_e_403	OCTETSTRING	'89EA00BFC040'O	It includes the list of f's in EGSM test cases. The coded set is {980, 981, 982, 983}, range 512 format. Length = 6 TC_26_10_4, k=1, c=3 TC: 26.10.5.1 k=1/2,c=3
C_flist_e_404	OCTETSTRING	'801EED02BEC0'O	It includes the list of f's in EGSM test cases. The coded set is {30, 40, 1010, 1015}, range 1024 format. Length = 6 TC_26_10_4, k=1, c=4 TC: 26.10.5.1 k=1/2,c=4
C_flist_e_405	OCTETSTRING	'8FEA001E000010'O	It includes the list of f's in EGSM test cases. The coded set is {980, 991, 992, 993, 994, 1015}. Length = 7 TC_26_10_4, k=1, c=5 TC: 26.10.5.1 k=1/2,c=5
C_flist_e_406	OCTETSTRING	'00000000000000020000008000080000'O	It includes the list of f's in EGSM test cases. The coded set is {20, 40, 66}, bitmap 0 format. Length = 16 TC_26_10_4, k=1, c=6 TC: 26.10.5.1 k=2,c=6
C_flist_e_407	OCTETSTRING	'8DEA0DF4CC6C4AFAFB10000000000000'O	It includes the list of f's in EGSM test cases. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}, range 128 format. Length = 16 TC_26_10_4, k=2, c=1 TC: 26.10.5.1 k=3,c=1 TC: 26.10.6 c=1
C_flist_e_408	OCTETSTRING	'8A0A1CFD3EF4610E2FFFFA0000'O	It includes the list of f's in EGSM test cases. The coded set is {20,

			40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114, 115}, range 256 format. Length = 13 TC_26_10_4, k=2, c=2 TC: 26.10.5.1 k=3,c=2 TC: 26.10.6 c=2
C_flist_e_409	OCTETSTRING	'89EA037F433C7B042BFEBFEC10000000'O	It includes the list of f's in EGSM test cases. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}, range 512 format. Length = 16 TC_26_10_4, k=2, c=3 TC: 26.10.5.2 k=3,c=3 TC: 26.10.6 c=3
C_flist_e_410	OCTETSTRING	'841EEA893EF98143B161'O	Length = 10 TC: 26.10.5.2 k=3,c=4
C_flist_e_411	OCTETSTRING	'8FEA703E084210'O	It includes the list of f's in EGSM test cases. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}, variable bitmap format. Length = 7 TC_26_10_4, k=2, c=5 TC: 26.10.5.1 k=3,c=5 TC: 26.10.6 c=5
C_flist_e_412	OCTETSTRING	'00020800000007F020000008000080000'O	It includes the list of f's in EGSM test cases. The coded set is {20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114, 115}, bitmap 0 format. Length = 16 TC_26_10_4, k=2, c=6 TC: 26.10.5.1 k=3,c=6
C_flist_e_415	OCTETSTRING	'841EEA893EF9814380'O	It includes the list of f's in EGSM test cases. The coded set is {0, 30, 40, 66, 80, 1005, 1010, 1015}, range 1024 format. Length = 9 Format: 1024 TC_26_10_4, k=2, c=4 TC: 26.10.6 c=4
C_flist_e_416	OCTETSTRING	'00020800000007F020000008000080000'O	bit map 0 format. The coded set is {20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114} Length = 16
C_BABR_0	B_3	'000'B	BABR value 0
C_BABR_1	B_3	'001'B	BABR value 1
C_BABR_2	B_3	'010'B	BABR value 2
C_BPM_0	B_3	'000'B	BPM value 0, 2 multiframes
C_BPM_2	B_3	'010'B	BPM value 2, 4 multiframes
C_BPM_3	B_3	'011'B	BPM value 3, 5 multiframes
C_BPM_7	B_3	'111'B	BPM value 7, 9 multiframes
C_T3212_0	OCTETSTRING	'00'O	T3212 infinite
C_T3212_1	OCTETSTRING	'01'O	T3212 6 mn / 1 deci hours
C_T3212_2	OCTETSTRING	'02'O	T3212 12 mn / 2 deci hours
C_T3212_5	OCTETSTRING	'05'O	T3212 30 mn / 5 deci hours
C_IMSI	INTEGER	0	MI type: IMSI
C_TMSI	INTEGER	1	MI type: TMSI
C_IMEI	INTEGER	2	MI type: IMEI
C_IMEISV	INTEGER	3	MI type: IMEISV
C_shortIMSI	HEXSTRING	'001011234'H	A short IMSI
C_LAC_1	OCTETSTRING	'0001'O	lac 1.
C_LAC_2	OCTETSTRING	'0002'O	lac 2.
C_LAC_3	OCTETSTRING	'0003'O	lac 3.
C_LAC_4	OCTETSTRING	'0004'O	lac 4.
C_LAC_5	OCTETSTRING	'0005'O	lac 5.
C_LAC_6	OCTETSTRING	'0006'O	lac 6.
C_LAC_7	OCTETSTRING	'0007'O	lac 7.
C_LAC_8	OCTETSTRING	'0008'O	lac 8.

C_LAC_spe	OCTETSTRING	'5344'O	lac '5344'H.
C_lacdeleted	OCTETSTRING	'FFFE'O	lac value deleted.
C_MCC_1	OCTETSTRING	'00F1'O	MCC 1
C_MCC_2	OCTETSTRING	'00F2'O	MCC 2
C_MCC_3	OCTETSTRING	'00F3'O	MCC 3
C_MCC_4	OCTETSTRING	'00F4'O	MCC 4
C_MCC_5	OCTETSTRING	'00F5'O	MCC 5
C_MCC_6	OCTETSTRING	'00F6'O	MCC 6
C_MCC_7	OCTETSTRING	'00F7'O	MCC 7
C_MCC_8	OCTETSTRING	'00F8'O	MCC 8
C_MCC_9	OCTETSTRING	'013B'O	MCC 9
C_Max_1	INTEGER	1	Max Retrans value 1
C_Max_2	INTEGER	2	Max Retrans value 2
C_Max_7	INTEGER	7	Max Retrans value 7
C_NECI_0	INTEGER	0	NECI value 0, not supported
C_NECI_1	INTEGER	1	NECI value 1, supported
C_PLMN_1	OCTETSTRING	'10'O	PLMN 1 (mnc=01)
C_PLMN_2	OCTETSTRING	'20'O	PLMN 2 (mnc=02)
C_PLMN_3	OCTETSTRING	'30'O	PLMN 3 (mnc=03)
C_PLMN_4	OCTETSTRING	'F0'O	PLMN 4
C_PLMN_5	OCTETSTRING	'F2'O	PLMN 5
C_PLMN_6	OCTETSTRING	'F3'O	PLMN 6
C_PLMN_7	OCTETSTRING	'F4'O	PLMN 7
C_PLMN_8	OCTETSTRING	'F5'O	PLMN 8
C_PLMN_9	OCTETSTRING	'F6'O	PLMN 9
C_PLMN_10	OCTETSTRING	'F7'O	PLMN 10
C_PLMN_Home	OCTETSTRING	'10'O	PLMN 1 (mnc=01)
C_TxInt_5	INTEGER	5	Tx Integer value 5
C_TxInt_7	INTEGER	7	Tx Integer value 7
C_TxInt_10	INTEGER	10	Tx Integer value 10
C_NotCombined	BOOLEAN	FALSE	CCCH not combined with SDCCH
C_Combined	BOOLEAN	TRUE	CCCH combined with SDCCH
C_FCCH_A	LOGICCH	"C_FCCH_A"	frequency correction channel of cell A(1)
C_FCCH_B	LOGICCH	"C_FCCH_B"	frequency correction channel of cell B(2)
C_FCCH_C	LOGICCH	"C_FCCH_C"	frequency correction channel of cell C(3)
C_FCCH_D	LOGICCH	"C_FCCH_D"	frequency correction channel of cell D(4)
C_FCCH_E	LOGICCH	"C_FCCH_E"	frequency correction channel of cell E(5)
C_FCCH_F	LOGICCH	"C_FCCH_F"	frequency correction channel of cell F(6)
C_FCCH_G	LOGICCH	"C_FCCH_G"	frequency correction channel of cell G(7)
C_FCCH_H	LOGICCH	"C_FCCH_H"	frequency correction channel of cell H(8)
C_SCH_A	LOGICCH	"C_SCH_A"	sync channel of cell A(1)
C_SCH_B	LOGICCH	"C_SCH_B"	sync channel of cell B(2)
C_SCH_C	LOGICCH	"C_SCH_C"	sync channel of cell C(3)
C_SCH_D	LOGICCH	"C_SCH_D"	sync channel of cell D(4)
C_SCH_E	LOGICCH	"C_SCH_E"	sync channel of cell E(5)
C_SCH_F	LOGICCH	"C_SCH_F"	sync channel of cell F(6)
C_SCH_G	LOGICCH	"C_SCH_G"	sync channel of cell G(7)
C_SCH_H	LOGICCH	"C_SCH_H"	sync channel of cell H(8)
C_BCCH_A_1	LOGICCH	"C_BCCH_A_1"	1st broadcast channel of cell A
C_BCCH_A_2	LOGICCH	"C_BCCH_A_2"	2nd broadcast channel of cell A
C_BCCH_A_3	LOGICCH	"C_BCCH_A_3"	3rd broadcast channel of cell A
C_BCCH_A_4	LOGICCH	"C_BCCH_A_4"	4th broadcast channel of cell A
C_BCCH_B_1	LOGICCH	"C_BCCH_B_1"	1st broadcast channel of cell B
C_BCCH_C_1	LOGICCH	"C_BCCH_C_1"	1st broadcast channel of cell C
C_BCCH_D_1	LOGICCH	"C_BCCH_D_1"	1st broadcast channel of cell D
C_BCCH_E_1	LOGICCH	"C_BCCH_E_1"	1st broadcast channel of cell E
C_BCCH_F_1	LOGICCH	"C_BCCH_F_1"	1st broadcast channel of cell F
C_BCCH_G_1	LOGICCH	"C_BCCH_G_1"	1st broadcast channel of cell G

C_BCCH_H_1	LOGICCH	"C_BCCH_H_1"	1st broadcast channel of cell H
C_CBCH_A	LOGICCH	"C_CBCH_A"	Cell broadcast channel in cell A
C_AGCH_A_1	LOGICCH	"C_AGCH_A_1"	1st down link CCCH (AGCH) of cell A
C_AGCH_A_2	LOGICCH	"C_AGCH_A_2"	2nd down link CCCH (AGCH) of cell A
C_AGCH_A_3	LOGICCH	"C_AGCH_A_3"	3rd down link CCCH (AGCH) of cell A
C_AGCH_A_4	LOGICCH	"C_AGCH_A_4"	4th down link CCCH (AGCH) of cell A
C_AGCH_B_1	LOGICCH	"C_AGCH_B_1"	1st down link CCCH (AGCH) of cell B
C_AGCH_B_2	LOGICCH	"C_AGCH_B_2"	2nd down link CCCH (AGCH) of cell B
C_AGCH_B_3	LOGICCH	"C_AGCH_B_3"	3rd down link CCCH (AGCH) of cell B
C_AGCH_B_4	LOGICCH	"C_AGCH_B_4"	4th down link CCCH (AGCH) of cell B
C_AGCH_C_1	LOGICCH	"C_AGCH_C_1"	1st down link CCCH (AGCH) of cell C
C_AGCH_C_2	LOGICCH	"C_AGCH_C_2"	2nd down link CCCH (AGCH) of cell C
C_AGCH_C_3	LOGICCH	"C_AGCH_C_3"	3rd down link CCCH (AGCH) of cell C
C_AGCH_C_4	LOGICCH	"C_AGCH_C_4"	4th down link CCCH (AGCH) of cell C
C_AGCH_D_1	LOGICCH	"C_AGCH_D_1"	1st down link CCCH (AGCH) of cell D
C_AGCH_E_1	LOGICCH	"C_AGCH_E_1"	1st down link CCCH (AGCH) of cell E
C_AGCH_F_1	LOGICCH	"C_AGCH_F_1"	1st down link CCCH (AGCH) of cell F
C_AGCH_G_1	LOGICCH	"C_AGCH_G_1"	1st down link CCCH (AGCH) of cell G
C_AGCH_H_1	LOGICCH	"C_AGCH_H_1"	1st down link CCCH (AGCH) of cell H
C_PCH_A_1	LOGICCH	"C_PCH_A_1"	1st down link CCCH (PCH) of cell A
C_PCH_A_2	LOGICCH	"C_PCH_A_2"	2nd down link CCCH (PCH) of cell A
C_PCH_A_3	LOGICCH	"C_PCH_A_3"	3rd down link CCCH (PCH) of cell A
C_PCH_A_4	LOGICCH	"C_PCH_A_4"	4th down link CCCH (PCH) of cell A
C_PCH_B_1	LOGICCH	"C_PCH_B_1"	1st down link CCCH (PCH) of cell B
C_PCH_B_2	LOGICCH	"C_PCH_B_2"	2nd down link CCCH (PCH) of cell B
C_PCH_B_3	LOGICCH	"C_PCH_B_3"	3rd down link CCCH (PCH) of cell B
C_PCH_B_4	LOGICCH	"C_PCH_B_4"	4th down link CCCH (PCH) of cell B
C_PCH_C_1	LOGICCH	"C_PCH_C_1"	1st down link CCCH (PCH) of cell C
C_PCH_C_2	LOGICCH	"C_PCH_C_2"	2nd down link CCCH (PCH) of cell C
C_PCH_C_3	LOGICCH	"C_PCH_C_3"	3rd down link CCCH (PCH) of cell C
C_PCH_C_4	LOGICCH	"C_PCH_C_4"	4th down link CCCH (PCH) of cell C
C_PCH_D_1	LOGICCH	"C_PCH_D_1"	1st down link CCCH (PCH and AGCH) of cell D
C_PCH_E_1	LOGICCH	"C_PCH_E_1"	1st down link CCCH (PCH) of cell E
C_PCH_F_1	LOGICCH	"C_PCH_F_1"	1st down link CCCH (PCH) of cell F
C_PCH_G_1	LOGICCH	"C_PCH_G_1"	1st down link CCCH (PCH) of cell G
C_PCH_H_1	LOGICCH	"C_PCH_H_1"	1st down link CCCH (PCH) of cell H
C_RACH_A_1	LOGICCH	"C_RACH_A_1"	1st uplink CCCH (RACH) of cell A

C_RACH_A_2	LOGICCH	"C_RACH_A_2"	2nd uplink CCCH (RACH) of cell A
C_RACH_A_3	LOGICCH	"C_RACH_A_3"	3rd uplink CCCH (RACH) of cell A
C_RACH_A_4	LOGICCH	"C_RACH_A_4"	4th uplink CCCH (RACH) of cell A
C_RACH_B_1	LOGICCH	"C_RACH_B_1"	1st uplink CCCH (RACH) of cell B
C_RACH_B_2	LOGICCH	"C_RACH_B_2"	2nd uplink CCCH (RACH) of cell B
C_RACH_B_3	LOGICCH	"C_RACH_B_3"	3rd uplink CCCH (RACH) of cell B
C_RACH_B_4	LOGICCH	"C_RACH_B_4"	4th uplink CCCH (RACH) of cell B
C_RACH_C_1	LOGICCH	"C_RACH_C_1"	1st uplink CCCH (RACH) of cell C
C_RACH_C_2	LOGICCH	"C_RACH_C_2"	2nd uplink CCCH (RACH) of cell C
C_RACH_C_3	LOGICCH	"C_RACH_C_3"	3rd uplink CCCH (RACH) of cell C
C_RACH_C_4	LOGICCH	"C_RACH_C_4"	4th uplink CCCH (RACH) of cell C
C_RACH_D_1	LOGICCH	"C_RACH_D_1"	1st uplink CCCH (RACH) of cell D
C_RACH_E_1	LOGICCH	"C_RACH_E_1"	1st uplink CCCH (RACH) of cell E
C_RACH_F_1	LOGICCH	"C_RACH_F_1"	1st uplink CCCH (RACH) of cell F
C_RACH_G_1	LOGICCH	"C_RACH_G_1"	1st uplink CCCH (RACH) of cell G
C_RACH_H_1	LOGICCH	"C_RACH_H_1"	1st uplink CCCH (RACH) of cell H
C_FACCHF_A_1	LOGICCH	"C_FACCHF_A_1"	FACCH associated with 1st TCH/F of cell A
C_FACCHF_A_2	LOGICCH	"C_FACCHF_A_2"	FACCH associated with 2nd TCH/F of cell A
C_FACCHF_A_3	LOGICCH	"C_FACCHF_A_3"	FACCH associated with 3rd TCH/F of cell A
C_FACCHF_B_1	LOGICCH	"C_FACCHF_B_1"	FACCH associated with 1st TCH/F of cell B
C_FACCHF_B_2	LOGICCH	"C_FACCHF_B_2"	FACCH associated with 2nd TCH/F of cell B
C_FACCHF_B_3	LOGICCH	"C_FACCHF_B_3"	FACCH associated with 3rd TCH/F of cell B
C_FACCHF_C_1	LOGICCH	"C_FACCHF_C_1"	FACCH associated with 1sd TCH/F of cell C
C_FACCHF_C_2	LOGICCH	"C_FACCHF_C_2"	FACCH associated with 2nd TCH/F of cell C
C_FACCHF_C_3	LOGICCH	"C_FACCHF_C_3"	FACCH associated with 3rd TCH/F of cell C
C_FACCHF_H_1	LOGICCH	"C_FACCHF_H_1"	FACCH associated with 1st TCH/F of cell H
C_FACCHH_A_1	LOGICCH	"C_FACCHH_A_1"	FACCH associated with 1st TCH/H of cell A
C_FACCHH_A_2	LOGICCH	"C_FACCHH_A_2"	FACCH associated with 2nd TCH/H of cell A
C_FACCHH_A_3	LOGICCH	"C_FACCHH_A_3"	FACCH associated with 3rd TCH/H of cell A
C_FACCHH_B_1	LOGICCH	"C_FACCHH_B_1"	FACCH associated with 1st TCH/H of cell B
C_FACCHH_B_2	LOGICCH	"C_FACCHH_B_2"	FACCH associated with 2nd TCH/H of cell B
C_FACCHH_B_3	LOGICCH	"C_FACCHH_B_3"	FACCH associated with 3rd TCH/H of cell B
C_FACCHH_C_1	LOGICCH	"C_FACCHH_C_1"	FACCH associated with 1st TCH/H of cell C
C_FACCHH_C_2	LOGICCH	"C_FACCHH_C_2"	FACCH associated with 2nd TCH/H of cell C
C_FACCHH_C_3	LOGICCH	"C_FACCHH_C_3"	FACCH associated with 3rd TCH/H of cell C
C_FACCHH0_A_1	LOGICCH	"C_FACCHH0_A_1"	FACCH associated with 1st TCH/H0 of cell A
C_FACCHH1_A_1	LOGICCH	"C_FACCHH1_A_1"	FACCH associated with 1st TCH/H1 of cell A
C_FACCHH0_A_2	LOGICCH	"C_FACCHH0_A_2"	FACCH associated with 2nd TCH/H0 of cell A
C_FACCHH1_A_2	LOGICCH	"C_FACCHH1_A_2"	FACCH associated with 2nd TCH/H1 of cell A
C_FACCHH0_A_3	LOGICCH	"C_FACCHH0_A_3"	FACCH associated with 3rd TCH/H0 of cell A
C_FACCHH1_A_3	LOGICCH	"C_FACCHH1_A_3"	FACCH associated with 3rd TCH/H1 of cell A
C_FACCHH0_B_1	LOGICCH	"C_FACCHH0_B_1"	FACCH associated with 1st

			TCH/H0 of cell B
C_FACCHH1_B_1	LOGICCH	"C_FACCHH1_B_1"	FACCH associated with 1st TCH/H1 of cell B
C_FACCHH0_B_2	LOGICCH	"C_FACCHH0_B_2"	FACCH associated with 2nd TCH/H0 of cell B
C_FACCHH1_B_2	LOGICCH	"C_FACCHH1_B_2"	FACCH associated with 2nd TCH/H1 of cell B
C_FACCHH0_B_3	LOGICCH	"C_FACCHH0_B_3"	FACCH associated with 3rd TCH/H0 of cell B
C_FACCHH1_B_3	LOGICCH	"C_FACCHH1_B_3"	FACCH associated with 3rd TCH/H1 of cell B
C_FACCHH0_C_1	LOGICCH	"C_FACCHH0_C_1"	FACCH associated with 1st TCH/H0 of cell C
C_FACCHH1_C_1	LOGICCH	"C_FACCHH1_C_1"	FACCH associated with 1st TCH/H1 of cell C
C_FACCHH0_C_2	LOGICCH	"C_FACCHH0_C_2"	FACCH associated with 2nd TCH/H0 of cell C
C_FACCHH1_C_2	LOGICCH	"C_FACCHH1_C_2"	FACCH associated with 2nd TCH/H1 of cell C
C_FACCHH0_C_3	LOGICCH	"C_FACCHH0_C_3"	FACCH associated with 3rd TCH/H0 of cell C
C_FACCHH1_C_3	LOGICCH	"C_FACCHH1_C_3"	FACCH associated with 3rd TCH/H1 of cell C
C_SACCH_A	LOGICCH	"C_SACCH_A"	all SACCHs of cell A
C_SACCH_B	LOGICCH	"C_SACCH_B"	all SACCHs of cell B
C_SACCH_C	LOGICCH	"C_SACCH_C"	all SACCHs of cell C
C_SACCH_D	LOGICCH	"C_SACCH_D"	all SACCHs of cell D
C_SACCH_E	LOGICCH	"C_SACCH_E"	all SACCHs of cell E
C_SACCH_F	LOGICCH	"C_SACCH_F"	all SACCHs of cell F
C_SACCH_G	LOGICCH	"C_SACCH_G"	all SACCHs of cell G
C_SACCH_H	LOGICCH	"C_SACCH_H"	all SACCHs of cell H
C_SACCHF_A_1	LOGICCH	"C_SACCHF_A_1"	SACCH associated with 1st TCH/F of cell A
C_SACCHF_A_2	LOGICCH	"C_SACCHF_A_2"	SACCH associated with 2nd TCH/F of cell A
C_SACCHF_B_1	LOGICCH	"C_SACCHF_B_1"	SACCH associated with 1st TCH/F of cell B
C_SACCHF_B_2	LOGICCH	"C_SACCHF_B_2"	SACCH associated with 2nd TCH/F of cell B
C_SACCHF_H_1	LOGICCH	"C_SACCHF_H_1"	SACCH associated with 1st TCH/F of cell H
C_SACCHH_A_1	LOGICCH	"C_SACCHH_A_1"	SACCH associated with 1st TCH/H of cell A
C_SACCHH_A_2	LOGICCH	"C_SACCHH_A_2"	SACCH associated with 2nd TCH/H of cell A
C_SACCHH_A_3	LOGICCH	"C_SACCHH_A_3"	SACCH associated with 3rd TCH/H of cell A
C_SACCHH_B_1	LOGICCH	"C_SACCHH_B_1"	SACCH associated with 1st TCH/H of cell B
C_SACCHH_B_2	LOGICCH	"C_SACCHH_B_2"	SACCH associated with 2nd TCH/H of cell B
C_SACCHH_B_3	LOGICCH	"C_SACCHH_B_3"	SACCH associated with 3rd TCH/H of cell B
C_SACCHH_C_1	LOGICCH	"C_SACCHH_C_1"	SACCH associated with 1st TCH/H of cell C
C_SACCHH_C_2	LOGICCH	"C_SACCHH_C_2"	SACCH associated with 2nd TCH/H of cell C
C_SACCHH_C_3	LOGICCH	"C_SACCHH_C_3"	SACCH associated with 3rd TCH/H of cell C
C_SACCHH0_A_1	LOGICCH	"C_SACCHH0_A_1"	SACCH associated with 1st TCH/H_0 of cell A
C_SACCHH1_A_1	LOGICCH	"C_SACCHH1_A_1"	SACCH associated with 1st TCH/H_1 of cell A
C_SACCHH0_A_2	LOGICCH	"C_SACCHH0_A_2"	SACCH associated with 2nd TCH/H_0 of cell A
C_SACCHH1_A_2	LOGICCH	"C_SACCHH1_A_2"	SACCH associated with 2nd TCH/H_1 of cell A
C_SACCHH0_A_3	LOGICCH	"C_SACCHH0_A_3"	SACCH associated with 3rd TCH/H_0 of cell A
C_SACCHH1_A_3	LOGICCH	"C_SACCHH1_A_3"	SACCH associated with 3rd

			TCH/H_1 of cell A
C_SACCHH0_B_1	LOGICCH	"C_SACCHH0_B_1"	SACCH associated with 1st TCH/H_0 of cell B
C_SACCHH1_B_1	LOGICCH	"C_SACCHH1_B_1"	SACCH associated with 1st TCH/H_1 of cell B
C_SACCHH0_B_2	LOGICCH	"C_SACCHH0_B_2"	SACCH associated with 2nd TCH/H_0 of cell B
C_SACCHH1_B_2	LOGICCH	"C_SACCHH1_B_2"	SACCH associated with 2nd TCH/H_1 of cell B
C_SACCHH0_B_3	LOGICCH	"C_SACCHH0_B_3"	SACCH associated with 3rd TCH/H_0 of cell B
C_SACCHH1_B_3	LOGICCH	"C_SACCHH1_B_3"	SACCH associated with 3rd TCH/H_1 of cell B
C_SACCHH0_C_1	LOGICCH	"C_SACCHH0_C_1"	SACCH associated with 1st TCH/H_0 of cell C
C_SACCHH1_C_1	LOGICCH	"C_SACCHH1_C_1"	SACCH associated with 1st TCH/H_1 of cell C
C_SACCHH0_C_2	LOGICCH	"C_SACCHH0_C_2"	SACCH associated with 2nd TCH/H_0 of cell C
C_SACCHH1_C_2	LOGICCH	"C_SACCHH1_C_2"	SACCH associated with 2nd TCH/H_1 of cell C
C_SACCHH0_C_3	LOGICCH	"C_SACCHH0_C_3"	SACCH associated with 3rd TCH/H_0 of cell C
C_SACCHH1_C_3	LOGICCH	"C_SACCHH1_C_3"	SACCH associated with 3rd TCH/H_1 of cell C
C_SACCHC4_A	LOGICCH	"C_SACCHC4_A"	SACCHC4 channel of cell A
C_SACCHC4_B	LOGICCH	"C_SACCHC4_B"	SACCHC4 channel of cell B
C_SACCHC4_C	LOGICCH	"C_SACCHC4_C"	SACCHC4 channel of cell C
C_SACCHC4_D	LOGICCH	"C_SACCHC4_D"	SACCHC4 channel of cell D
C_SACCHC4_E	LOGICCH	"C_SACCHC4_E"	SACCHC4 channel of cell E
C_SACCHC4_F	LOGICCH	"C_SACCHC4_F"	SACCHC4 channel of cell F
C_SACCHC4_G	LOGICCH	"C_SACCHC4_G"	SACCHC4 channel of cell G
C_SACCHC4_H	LOGICCH	"C_SACCHC4_H"	SACCHC4 channel of cell H
C_SACCHC40_A	LOGICCH	"C_SACCHC40_A"	SACCH/C4(0) associated with SDCCH/4(0) of cell A
C_SACCHC41_A	LOGICCH	"C_SACCHC41_A"	SACCH/C4(1) associated with SDCCH/4(1) of cell A
C_SACCHC42_A	LOGICCH	"C_SACCHC42_A"	SACCH/C4(2) associated with SDCCH/4(2) of cell A
C_SACCHC43_A	LOGICCH	"C_SACCCH43_A"	SACCH/C4(3) associated with SDCCH/4(3) of cell A
C_SACCHC40_B	LOGICCH	"C_SACCHC40_B"	SACCH/C4(0) associated with SDCCH/4(0) of cell B
C_SACCHC41_B	LOGICCH	"C_SACCHC41_B"	SACCH/C4(1) associated with SDCCH/4(1) of cell B
C_SACCHC42_B	LOGICCH	"C_SACCHC42_B"	SACCH/C4(2) associated with SDCCH/4(2) of cell B
C_SACCHC43_B	LOGICCH	"C_SACCHC43_B"	SACCH/C4(3) associated with SDCCH/4(3) of cell B
C_SACCHC40_C	LOGICCH	"C_SACCHC40_C"	SACCH/C4(0) associated with SDCCH/4(0) of cell C
C_SACCHC41_C	LOGICCH	"C_SACCHC41_C"	SACCH/C4(1) associated with SDCCH/4(1) of cell C
C_SACCHC42_C	LOGICCH	"C_SACCHC42_C"	SACCH/C4(2) associated with SDCCH/4(2) of cell C
C_SACCHC43_C	LOGICCH	"C_SACCHC43_C"	SACCH/C4(3) associated with SDCCH/4(3) of cell C
C_SACCHC40_D	LOGICCH	"C_SACCHC40_D"	SACCH/C4(0) associated with SDCCH/4(0) of cell D
C_SACCHC41_D	LOGICCH	"C_SACCHC41_D"	SACCH/C4(1) associated with SDCCH/4(1) of cell D
C_SACCHC42_D	LOGICCH	"C_SACCHC42_D"	SACCH/C4(2) associated with SDCCH/4(2) of cell D
C_SACCHC43_D	LOGICCH	"C_SACCCH43_D"	SACCH/C4(3) associated with SDCCH/4(3) of cell D
C_SACCHC40_E	LOGICCH	"C_SACCHC40_E"	SACCH/C4(0) associated with SDCCH/4(0) of cell E
C_SACCHC41_E	LOGICCH	"C_SACCHC41_E"	SACCH/C4(1) associated with SDCCH/4(1) of cell E
C_SACCHC42_E	LOGICCH	"C_SACCHC42_E"	SACCH/C4(2) associated with

			SDCCH/4(2) of cell E
C_SACCHC43_E	LOGICCH	"C_SACCHC43_E"	SACCH/C4(3) associated with SDCCH/4(3) of cell E
C_SACCHC40_F	LOGICCH	"C_SACCHC40_F"	SACCH/C4(0) associated with SDCCH/4(0) of cell F
C_SACCHC41_F	LOGICCH	"C_SACCHC41_F"	SACCH/C4(1) associated with SDCCH/4(1) of cell F
C_SACCHC42_F	LOGICCH	"C_SACCHC42_F"	SACCH/C4(2) associated with SDCCH/4(2) of cell F
C_SACCHC43_F	LOGICCH	"C_SACCHC43_F"	SACCH/C4(3) associated with SDCCH/4(3) of cell F
C_SACCHC40_G	LOGICCH	"C_SACCHC40_G"	SACCH/C4(0) associated with SDCCH/4(0) of cell G
C_SACCHC41_G	LOGICCH	"C_SACCHC41_G"	SACCH/C4(1) associated with SDCCH/4(1) of cell G
C_SACCHC42_G	LOGICCH	"C_SACCHC42_G"	SACCH/C4(2) associated with SDCCH/4(2) of cell G
C_SACCHC43_G	LOGICCH	"C_SACCHC43_G"	SACCH/C4(3) associated with SDCCH/4(3) of cell G
C_SACCHC40_H	LOGICCH	"C_SACCHC40_C"	SACCH/C4(0) associated with SDCCH/4(0) of cell H
C_SACCHC41_H	LOGICCH	"C_SACCHC41_C"	SACCH/C4(1) associated with SDCCH/4(1) of cell H
C_SACCHC42_H	LOGICCH	"C_SACCHC42_C"	SACCH/C4(2) associated with SDCCH/4(2) of cell H
C_SACCHC43_H	LOGICCH	"C_SACCHC43_C"	SACCH/C4(3) associated with SDCCH/4(3) of cell H
C_SACCHC8_A_1	LOGICCH	"C_SACCHC8_A_1"	1st SACCH/8 channel of cell A
C_SACCHC8_A_2	LOGICCH	"C_SACCHC8_A_2"	2nd SACCH/8 channel of cell A
C_SACCHC8_A_3	LOGICCH	"C_SACCHC8_A_3"	3rd SACCH/8 channel of cell A
C_SACCHC8_B_1	LOGICCH	"C_SACCHC8_B_1"	1st SACCH/8 channel of cell B
C_SACCHC8_B_2	LOGICCH	"C_SACCHC8_B_2"	2nd SACCH/8 channel of cell B
C_SACCHC8_B_3	LOGICCH	"C_SACCHC8_B_3"	3rd SACCH/8 channel of cell B
C_SACCHC8_C_1	LOGICCH	"C_SACCHC8_C_1"	1st SACCH/8 channel of cell C
C_SACCHC8_C_2	LOGICCH	"C_SACCHC8_C_2"	2nd SACCH/8 channel of cell C
C_SACCHC8_C_3	LOGICCH	"C_SACCHC8_C_3"	3rd SACCH/8 channel of cell C
C_SACCHC80_A_1	LOGICCH	"C_SACCHC80_A_1"	SACCH/C8(0) associated with 1st SDCCH/8(0) of cell A
C_SACCHC81_A_1	LOGICCH	"C_SACCHC81_A_1"	SACCH/C8(1) associated with 1st SDCCH/8(1) of cell A
C_SACCHC82_A_1	LOGICCH	"C_SACCHC82_A_1"	SACCH/C8(2) associated with 1st SDCCH/8(2) of cell A
C_SACCHC83_A_1	LOGICCH	"C_SACCHC83_A_1"	SACCH/C8(3) associated with 1st SDCCH/8(3) of cell A
C_SACCHC84_A_1	LOGICCH	"C_SACCHC84_A_1"	SACCH/C8(4) associated with 1st SDCCH/8(4) of cell A
C_SACCHC85_A_1	LOGICCH	"C_SACCHC85_A_1"	SACCH/C8(5) associated with 1st SDCCH/8(5) of cell A
C_SACCHC86_A_1	LOGICCH	"C_SACCHC86_A_1"	SACCH/C8(6) associated with 1st SDCCH/8(6) of cell A
C_SACCHC87_A_1	LOGICCH	"C_SACCHC87_A_1"	SACCH/C8(7) associated with 1st SDCCH/8(7) of cell A
C_SACCHC80_A_2	LOGICCH	"C_SACCHC80_A_2"	SACCH/C8(0) associated with 2nd SDCCH/8(0) of cell A
C_SACCHC81_A_2	LOGICCH	"C_SACCHC81_A_2"	SACCH/C8(1) associated with 2nd SDCCH/8(1) of cell A
C_SACCHC82_A_2	LOGICCH	"C_SACCHC82_A_2"	SACCH/C8(2) associated with 2nd SDCCH/8(2) of cell A
C_SACCHC83_A_2	LOGICCH	"C_SACCHC83_A_2"	SACCH/C8(3) associated with 2nd SDCCH/8(3) of cell A
C_SACCHC84_A_2	LOGICCH	"C_SACCHC84_A_2"	SACCH/C8(4) associated with 2nd SDCCH/8(4) of cell A
C_SACCHC85_A_2	LOGICCH	"C_SACCHC85_A_2"	SACCH/C8(5) associated with 2nd SDCCH/8(5) of cell A
C_SACCHC86_A_2	LOGICCH	"C_SACCHC86_A_2"	SACCH/C8(6) associated with 2nd SDCCH/8(6) of cell A
C_SACCHC87_A_2	LOGICCH	"C_SACCHC87_A_2"	SACCH/C8(7) associated with 2nd SDCCH/8(7) of cell A
C_SACCHC80_A_3	LOGICCH	"C_SACCHC80_A_3"	SACCH/C8(0) associated with 3rd SDCCH/8(0) of cell A

C_SACCHC81_A_3	LOGICCH	"C_SACCHC81_A_3"	SACCH/C8(1) associated with 3rd SDCCH/8(1) of cell A
C_SACCHC82_A_3	LOGICCH	"C_SACCHC82_A_3"	SACCH/C8(2) associated with 3rd SDCCH/8(2) of cell A
C_SACCHC83_A_3	LOGICCH	"C_SACCHC83_A_3"	SACCH/C8(3) associated with 3rd SDCCH/8(3) of cell A
C_SACCHC84_A_3	LOGICCH	"C_SACCHC84_A_3"	SACCH/C8(4) associated with 3rd SDCCH/8(4) of cell A
C_SACCHC85_A_3	LOGICCH	"C_SACCHC85_A_3"	SACCH/C8(5) associated with 3rd SDCCH/8(5) of cell A
C_SACCHC86_A_3	LOGICCH	"C_SACCHC86_A_3"	SACCH/C8(6) associated with 3rd SDCCH/8(6) of cell A
C_SACCHC87_A_3	LOGICCH	"C_SACCHC87_A_3"	SACCH/C8(7) associated with 3rd SDCCH/8(7) of cell A
C_SACCHC80_B_1	LOGICCH	"C_SACCHC80_B_1"	SACCH/C8(0) associated with 1st SDCCH/8(0) of cell B
C_SACCHC81_B_1	LOGICCH	"C_SACCHC81_B_1"	SACCH/C8(1) associated with 1st SDCCH/8(1) of cell B
C_SACCHC82_B_1	LOGICCH	"C_SACCHC82_B_1"	SACCH/C8(2) associated with 1st SDCCH/8(2) of cell B
C_SACCHC83_B_1	LOGICCH	"C_SACCHC83_B_1"	SACCH/C8(3) associated with 1st SDCCH/8(3) of cell B
C_SACCHC84_B_1	LOGICCH	"C_SACCHC84_B_1"	SACCH/C8(4) associated with 1st SDCCH/8(4) of cell B
C_SACCHC85_B_1	LOGICCH	"C_SACCHC85_B_1"	SACCH/C8(5) associated with 1st SDCCH/8(5) of cell B
C_SACCHC86_B_1	LOGICCH	"C_SACCHC86_B_1"	SACCH/C8(6) associated with 1st SDCCH/8(6) of cell B
C_SACCHC87_B_1	LOGICCH	"C_SACCHC87_B_1"	SACCH/C8(7) associated with 1st SDCCH/8(7) of cell B
C_SACCHC80_B_2	LOGICCH	"C_SACCHC80_B_2"	SACCH/C8(0) associated with 2nd SDCCH/8(0) of cell B
C_SACCHC81_B_2	LOGICCH	"C_SACCHC81_B_2"	SACCH/C8(1) associated with 2nd SDCCH/8(1) of cell B
C_SACCHC82_B_2	LOGICCH	"C_SACCHC82_B_2"	SACCH/C8(2) associated with 2nd SDCCH/8(2) of cell B
C_SACCHC83_B_2	LOGICCH	"C_SACCHC83_B_2"	SACCH/C8(3) associated with 2nd SDCCH/8(3) of cell B
C_SACCHC84_B_2	LOGICCH	"C_SACCHC84_B_2"	SACCH/C8(4) associated with 2nd SDCCH/8(4) of cell B
C_SACCHC85_B_2	LOGICCH	"C_SACCHC85_B_2"	SACCH/C8(5) associated with 2nd SDCCH/8(5) of cell B
C_SACCHC86_B_2	LOGICCH	"C_SACCHC86_B_2"	SACCH/C8(6) associated with 2nd SDCCH/8(6) of cell B
C_SACCHC87_B_2	LOGICCH	"C_SACCHC87_B_2"	SACCH/C8(7) associated with 2nd SDCCH/8(7) of cell B
C_SACCHC80_B_3	LOGICCH	"C_SACCHC80_B_3"	SACCH/C8(0) associated with 3rd SDCCH/8(0) of cell B
C_SACCHC81_B_3	LOGICCH	"C_SACCHC81_B_3"	SACCH/C8(1) associated with 3rd SDCCH/8(1) of cell B
C_SACCHC82_B_3	LOGICCH	"C_SACCHC82_B_3"	SACCH/C8(2) associated with 3rd SDCCH/8(2) of cell B
C_SACCHC83_B_3	LOGICCH	"C_SACCHC83_B_3"	SACCH/C8(3) associated with 3rd SDCCH/8(3) of cell B
C_SACCHC84_B_3	LOGICCH	"C_SACCHC84_B_3"	SACCH/C8(4) associated with 3rd SDCCH/8(4) of cell B
C_SACCHC85_B_3	LOGICCH	"C_SACCHC85_B_3"	SACCH/C8(5) associated with 3rd SDCCH/8(5) of cell B
C_SACCHC86_B_3	LOGICCH	"C_SACCHC86_B_3"	SACCH/C8(6) associated with 3rd SDCCH/8(6) of cell B
C_SACCHC87_B_3	LOGICCH	"C_SACCHC87_B_3"	SACCH/C8(7) associated with 3rd SDCCH/8(7) of cell B
C_SACCHC80_C_1	LOGICCH	"C_SACCHC80_C_1"	SACCH/C8(0) associated with 1st SDCCH/8(0) of cell C
C_SACCHC81_C_1	LOGICCH	"C_SACCHC81_C_1"	SACCH/C8(1) associated with 1st SDCCH/8(1) of cell C
C_SACCHC82_C_1	LOGICCH	"C_SACCHC82_C_1"	SACCH/C8(2) associated with 1st SDCCH/8(2) of cell C
C_SACCHC83_C_1	LOGICCH	"C_SACCHC83_C_1"	SACCH/C8(3) associated with 1st SDCCH/8(3) of cell C
C_SACCHC84_C_1	LOGICCH	"C_SACCHC84_C_1"	SACCH/C8(4) associated with 1st

			SDCCH/8(4) of cell C
C_SACCHC85_C_1	LOGICCH	"C_SACCHC85_C_1"	SACCH/C8(5) associated with 1st SDCCH/8(5) of cell C
C_SACCHC86_C_1	LOGICCH	"C_SACCHC86_C_1"	SACCH/C8(6) associated with 1st SDCCH/8(6) of cell C
C_SACCHC87_C_1	LOGICCH	"C_SACCHC87_C_1"	SACCH/C8(7) associated with 1st SDCCH/8(7) of cell C
C_SACCHC80_C_2	LOGICCH	"C_SACCHC80_C_2"	SACCH/C8(0) associated with 2nd SDCCH/8(0) of cell C
C_SACCHC81_C_2	LOGICCH	"C_SACCHC81_C_2"	SACCH/C8(1) associated with 2nd SDCCH/8(1) of cell C
C_SACCHC82_C_2	LOGICCH	"C_SACCHC82_C_2"	SACCH/C8(2) associated with 2nd SDCCH/8(2) of cell C
C_SACCHC83_C_2	LOGICCH	"C_SACCHC83_C_2"	SACCH/C8(3) associated with 2nd SDCCH/8(3) of cell C
C_SACCHC84_C_2	LOGICCH	"C_SACCHC84_C_2"	SACCH/C8(4) associated with 2nd SDCCH/8(4) of cell C
C_SACCHC85_C_2	LOGICCH	"C_SACCHC85_C_2"	SACCH/C8(5) associated with 2nd SDCCH/8(5) of cell C
C_SACCHC86_C_2	LOGICCH	"C_SACCHC86_C_2"	SACCH/C8(6) associated with 2nd SDCCH/8(6) of cell C
C_SACCHC87_C_2	LOGICCH	"C_SACCHC87_C_2"	SACCH/C8(7) associated with 2nd SDCCH/8(7) of cell C
C_SACCHC80_C_3	LOGICCH	"C_SACCHC80_C_3"	SACCH/C8(0) associated with 3rd SDCCH/8(0) of cell C
C_SACCHC81_C_3	LOGICCH	"C_SACCHC81_C_3"	SACCH/C8(1) associated with 3rd SDCCH/8(1) of cell C
C_SACCHC82_C_3	LOGICCH	"C_SACCHC82_C_3"	SACCH/C8(2) associated with 3rd SDCCH/8(2) of cell C
C_SACCHC83_C_3	LOGICCH	"C_SACCHC83_C_3"	SACCH/C8(3) associated with 3rd SDCCH/8(3) of cell C
C_SACCHC84_C_3	LOGICCH	"C_SACCHC84_C_3"	SACCH/C8(4) associated with 3rd SDCCH/8(4) of cell C
C_SACCHC85_C_3	LOGICCH	"C_SACCHC85_C_3"	SACCH/C8(5) associated with 3rd SDCCH/8(5) of cell C
C_SACCHC86_C_3	LOGICCH	"C_SACCHC86_C_3"	SACCH/C8(6) associated with 3rd SDCCH/8(6) of cell C
C_SACCHC87_C_3	LOGICCH	"C_SACCHC87_C_3"	SACCH/C8(7) associated with 3rd SDCCH/8(7) of cell C
C_SDCCH4_A	LOGICCH	"C_SDCCH4_A"	SDCCH/4 channel of cell A
C_SDCCH4_B	LOGICCH	"C_SDCCH4_B"	SDCCH/4 channel of cell B
C_SDCCH4_C	LOGICCH	"C_SDCCH4_C"	SDCCH/4 channel of cell C
C_SDCCH4_D	LOGICCH	"C_SDCCH4_D"	SDCCH/4 channel of cell D
C_SDCCH4_E	LOGICCH	"C_SDCCH4_E"	SDCCH/4 channel of cell E
C_SDCCH4_F	LOGICCH	"C_SDCCH4_F"	SDCCH/4 channel of cell F
C_SDCCH4_G	LOGICCH	"C_SDCCH4_G"	SDCCH/4 channel of cell G
C_SDCCH4_H	LOGICCH	"C_SDCCH4_H"	SDCCH/4 channel of cell H
C_SDCCH40_A	LOGICCH	"C_SDCCH40_A"	SDCCH/4(0) of cell A
C_SDCCH41_A	LOGICCH	"C_SDCCH41_A"	SDCCH/4(1) of cell A
C_SDCCH42_A	LOGICCH	"C_SDCCH42_A"	SDCCH/4(2) of cell A
C_SDCCH43_A	LOGICCH	"C_SDCCH43_A"	SDCCH/4(3) of cell A
C_SDCCH40_B	LOGICCH	"C_SDCCH40_B"	SDCCH/4(0) of cell B
C_SDCCH41_B	LOGICCH	"C_SDCCH41_B"	SDCCH/4(1) of cell B
C_SDCCH42_B	LOGICCH	"C_SDCCH42_B"	SDCCH/4(2) of cell B
C_SDCCH43_B	LOGICCH	"C_SDCCH43_B"	SDCCH/4(3) of cell B
C_SDCCH40_C	LOGICCH	"C_SDCCH40_C"	SDCCH/4(0) of cell C
C_SDCCH41_C	LOGICCH	"C_SDCCH41_C"	SDCCH/4(1) of cell C
C_SDCCH42_C	LOGICCH	"C_SDCCH42_C"	SDCCH/4(2) of cell C
C_SDCCH43_C	LOGICCH	"C_SDCCH43_C"	SDCCH/4(3) of cell C
C_SDCCH40_D	LOGICCH	"C_SDCCH40_D"	SDCCH/4(0) of cell D
C_SDCCH41_D	LOGICCH	"C_SDCCH41_D"	SDCCH/4(1) of cell D
C_SDCCH42_D	LOGICCH	"C_SDCCH42_D"	SDCCH/4(2) of cell D
C_SDCCH43_D	LOGICCH	"C_SDCCH43_D"	SDCCH/4(3) of cell D
C_SDCCH40_E	LOGICCH	"C_SDCCH40_E"	SDCCH/4(0) of cell E
C_SDCCH41_E	LOGICCH	"C_SDCCH41_E"	SDCCH/4(1) of cell E
C_SDCCH42_E	LOGICCH	"C_SDCCH42_E"	SDCCH/4(2) of cell E
C_SDCCH43_E	LOGICCH	"C_SDCCH43_E"	SDCCH/4(3) of cell E
C_SDCCH40_F	LOGICCH	"C_SDCCH40_F"	SDCCH/4(0) of cell F

C_SDCCH41_F	LOGICCH	"C_SDCCH41_F"	SDCCH/4(1) of cell F
C_SDCCH42_F	LOGICCH	"C_SDCCH42_F"	SDCCH/4(2) of cell F
C_SDCCH43_F	LOGICCH	"C_SDCCH43_F"	SDCCH/4(3) of cell F
C_SDCCH40_G	LOGICCH	"C_SDCCH40_G"	SDCCH/4(0) of cell G
C_SDCCH41_G	LOGICCH	"C_SDCCH41_G"	SDCCH/4(1) of cell G
C_SDCCH42_G	LOGICCH	"C_SDCCH42_G"	SDCCH/4(2) of cell G
C_SDCCH43_G	LOGICCH	"C_SDCCH43_G"	SDCCH/4(3) of cell G
C_SDCCH40_H	LOGICCH	"C_SDCCH40_H"	SDCCH/4(0) of cell H
C_SDCCH41_H	LOGICCH	"C_SDCCH41_H"	SDCCH/4(1) of cell H
C_SDCCH42_H	LOGICCH	"C_SDCCH42_H"	SDCCH/4(2) of cell H
C_SDCCH43_H	LOGICCH	"C_SDCCH43_H"	SDCCH/4(3) of cell H
C_SDCCH8_A_1	LOGICCH	"C_SDCCH8_A_1"	1st SDCCH/8 channel of cell A
C_SDCCH8_A_2	LOGICCH	"C_SDCCH8_A_2"	2nd SDCCH/8 channel of cell A
C_SDCCH8_A_3	LOGICCH	"C_SDCCH8_A_3"	3rd SDCCH/8 channel of cell A
C_SDCCH8_B_1	LOGICCH	"C_SDCCH8_B_1"	1st SDCCH/8 channel of cell B
C_SDCCH8_B_2	LOGICCH	"C_SDCCH8_B_2"	2nd SDCCH/8 channel of cell B
C_SDCCH80_A_1	LOGICCH	"C_SDCCH80_A_1"	1st SDCCH/8(0) of cell A
C_SDCCH81_A_1	LOGICCH	"C_SDCCH81_A_1"	1st SDCCH/8(1) of cell A
C_SDCCH82_A_1	LOGICCH	"C_SDCCH82_A_1"	1st SDCCH/8(2) of cell A
C_SDCCH83_A_1	LOGICCH	"C_SDCCH83_A_1"	1st SDCCH/8(3) of cell A
C_SDCCH84_A_1	LOGICCH	"C_SDCCH84_A_1"	1st SDCCH/8(4) of cell A
C_SDCCH85_A_1	LOGICCH	"C_SDCCH85_A_1"	1st SDCCH/8(5) of cell A
C_SDCCH86_A_1	LOGICCH	"C_SDCCH86_A_1"	1st SDCCH/8(6) of cell A
C_SDCCH87_A_1	LOGICCH	"C_SDCCH87_A_1"	1st SDCCH/8(7) of cell A
C_SDCCH80_A_2	LOGICCH	"C_SDCCH80_A_2"	2nd SDCCH/8(0) of cell A
C_SDCCH81_A_2	LOGICCH	"C_SDCCH81_A_2"	2nd SDCCH/8(1) of cell A
C_SDCCH82_A_2	LOGICCH	"C_SDCCH82_A_2"	2nd SDCCH/8(2) of cell A
C_SDCCH83_A_2	LOGICCH	"C_SDCCH83_A_2"	2nd SDCCH/8(3) of cell A
C_SDCCH84_A_2	LOGICCH	"C_SDCCH84_A_2"	2nd SDCCH/8(4) of cell A
C_SDCCH85_A_2	LOGICCH	"C_SDCCH85_A_2"	2nd SDCCH/8(5) of cell A
C_SDCCH86_A_2	LOGICCH	"C_SDCCH86_A_2"	2nd SDCCH/8(6) of cell A
C_SDCCH87_A_2	LOGICCH	"C_SDCCH87_A_2"	2nd SDCCH/8(7) of cell A
C_SDCCH80_A_3	LOGICCH	"C_SDCCH80_A_3"	3rd SDCCH/8(0) of cell A
C_SDCCH81_A_3	LOGICCH	"C_SDCCH81_A_3"	3rd SDCCH/8(1) of cell A
C_SDCCH82_A_3	LOGICCH	"C_SDCCH82_A_3"	3rd SDCCH/8(2) of cell A
C_SDCCH83_A_3	LOGICCH	"C_SDCCH83_A_3"	3rd SDCCH/8(3) of cell A
C_SDCCH84_A_3	LOGICCH	"C_SDCCH84_A_3"	3rd SDCCH/8(4) of cell A
C_SDCCH85_A_3	LOGICCH	"C_SDCCH85_A_3"	3rd SDCCH/8(5) of cell A
C_SDCCH86_A_3	LOGICCH	"C_SDCCH86_A_3"	3rd SDCCH/8(6) of cell A
C_SDCCH87_A_3	LOGICCH	"C_SDCCH87_A_3"	3rd SDCCH/8(7) of cell A
C_SDCCH80_B_1	LOGICCH	"C_SDCCH80_B_1"	1st SDCCH/8(0) of cell B
C_SDCCH81_B_1	LOGICCH	"C_SDCCH81_B_1"	1st SDCCH/8(1) of cell B
C_SDCCH82_B_1	LOGICCH	"C_SDCCH82_B_1"	1st SDCCH/8(2) of cell B
C_SDCCH83_B_1	LOGICCH	"C_SDCCH83_B_1"	1st SDCCH/8(3) of cell B
C_SDCCH84_B_1	LOGICCH	"C_SDCCH84_B_1"	1st SDCCH/8(4) of cell B
C_SDCCH85_B_1	LOGICCH	"C_SDCCH85_B_1"	1st SDCCH/8(5) of cell B
C_SDCCH86_B_1	LOGICCH	"C_SDCCH86_B_1"	1st SDCCH/8(6) of cell B
C_SDCCH87_B_1	LOGICCH	"C_SDCCH87_B_1"	1st SDCCH/8(7) of cell B
C_SDCCH80_B_2	LOGICCH	"C_SDCCH80_B_2"	2nd SDCCH/8(0) of cell B
C_SDCCH81_B_2	LOGICCH	"C_SDCCH81_B_2"	2nd SDCCH/8(1) of cell B
C_SDCCH82_B_2	LOGICCH	"C_SDCCH82_B_2"	2nd SDCCH/8(2) of cell B
C_SDCCH83_B_2	LOGICCH	"C_SDCCH83_B_2"	2nd SDCCH/8(3) of cell B
C_SDCCH84_B_2	LOGICCH	"C_SDCCH84_B_2"	2nd SDCCH/8(4) of cell B
C_SDCCH85_B_2	LOGICCH	"C_SDCCH85_B_2"	2nd SDCCH/8(5) of cell B
C_SDCCH86_B_2	LOGICCH	"C_SDCCH86_B_2"	2nd SDCCH/8(6) of cell B
C_SDCCH87_B_2	LOGICCH	"C_SDCCH87_B_2"	2nd SDCCH/8(7) of cell B
C_SDCCH80_B_3	LOGICCH	"C_SDCCH80_B_3"	3rd SDCCH/8(0) of cell B
C_SDCCH81_B_3	LOGICCH	"C_SDCCH81_B_3"	3rd SDCCH/8(1) of cell B
C_SDCCH82_B_3	LOGICCH	"C_SDCCH82_B_3"	3rd SDCCH/8(2) of cell B
C_SDCCH83_B_3	LOGICCH	"C_SDCCH83_B_3"	3rd SDCCH/8(3) of cell B
C_SDCCH84_B_3	LOGICCH	"C_SDCCH84_B_3"	3rd SDCCH/8(4) of cell B
C_SDCCH85_B_3	LOGICCH	"C_SDCCH85_B_3"	3rd SDCCH/8(5) of cell B
C_SDCCH86_B_3	LOGICCH	"C_SDCCH86_B_3"	3rd SDCCH/8(6) of cell B
C_SDCCH87_B_3	LOGICCH	"C_SDCCH87_B_3"	3rd SDCCH/8(7) of cell B
C_SDCCH80_C_1	LOGICCH	"C_SDCCH80_C_1"	1st SDCCH/8(0) of cell C

C_SDCCH81_C_1	LOGICCH	"C_SDCCH81_C_1"	1st SDCCH/8(1) of cell C
C_SDCCH82_C_1	LOGICCH	"C_SDCCH82_C_1"	1st SDCCH/8(2) of cell C
C_SDCCH83_C_1	LOGICCH	"C_SDCCH83_C_1"	1st SDCCH/8(3) of cell C
C_SDCCH84_C_1	LOGICCH	"C_SDCCH84_C_1"	1st SDCCH/8(4) of cell C
C_SDCCH85_C_1	LOGICCH	"C_SDCCH85_C_1"	1st SDCCH/8(5) of cell C
C_SDCCH86_C_1	LOGICCH	"C_SDCCH86_C_1"	1st SDCCH/8(6) of cell C
C_SDCCH87_C_1	LOGICCH	"C_SDCCH87_C_1"	1st SDCCH/8(7) of cell C
C_SDCCH80_C_2	LOGICCH	"C_SDCCH80_C_2"	2nd SDCCH/8(0) of cell C
C_SDCCH81_C_2	LOGICCH	"C_SDCCH81_C_2"	2nd SDCCH/8(1) of cell C
C_SDCCH82_C_2	LOGICCH	"C_SDCCH82_C_2"	2nd SDCCH/8(2) of cell C
C_SDCCH83_C_2	LOGICCH	"C_SDCCH83_C_2"	2nd SDCCH/8(3) of cell C
C_SDCCH84_C_2	LOGICCH	"C_SDCCH84_C_2"	2nd SDCCH/8(4) of cell C
C_SDCCH85_C_2	LOGICCH	"C_SDCCH85_C_2"	2nd SDCCH/8(5) of cell C
C_SDCCH86_C_2	LOGICCH	"C_SDCCH86_C_2"	2nd SDCCH/8(6) of cell C
C_SDCCH87_C_2	LOGICCH	"C_SDCCH87_C_2"	2nd SDCCH/8(7) of cell C
C_SDCCH80_C_3	LOGICCH	"C_SDCCH80_C_3"	3rd SDCCH/8(0) of cell C
C_SDCCH81_C_3	LOGICCH	"C_SDCCH81_C_3"	3rd SDCCH/8(1) of cell C
C_SDCCH82_C_3	LOGICCH	"C_SDCCH82_C_3"	3rd SDCCH/8(2) of cell C
C_SDCCH83_C_3	LOGICCH	"C_SDCCH83_C_3"	3rd SDCCH/8(3) of cell C
C_SDCCH84_C_3	LOGICCH	"C_SDCCH84_C_3"	3rd SDCCH/8(4) of cell C
C_SDCCH85_C_3	LOGICCH	"C_SDCCH85_C_3"	3rd SDCCH/8(5) of cell C
C_SDCCH86_C_3	LOGICCH	"C_SDCCH86_C_3"	3rd SDCCH/8(6) of cell C
C_SDCCH87_C_3	LOGICCH	"C_SDCCH87_C_3"	3rd SDCCH/8(7) of cell C
C_CHSDCCH8_FH	OCTETSTRING	'00'O	ch.type SDCCH8 with FH
C_CHSDCCH8_NonFH	OCTETSTRING	'01'O	ch.type SDCCH8 No FH
C_CHSDCCH4_NonFH	OCTETSTRING	'03'O	ch.type SDCCH4 NoFH
C_CHTCHF_FH	OCTETSTRING	'04'O	ch.type TCHF FH
C_CHTCHF_NonFH	OCTETSTRING	'05'O	ch.type TCHF NonFH
C_CHTCHH_FH	OCTETSTRING	'06'O	ch.type TCHH FH
C_CHTCHH_NonFH	OCTETSTRING	'07'O	ch.type TCHH NonFH
C_TCHF_ACCHF_1	LOGCH	"C_TCHF_ACCHF_1"	first TCH/F + ACCHs type channel
C_TCHF_ACCHF_2	LOGCH	"C_TCHF_ACCHF_2"	second TCH/F + ACCHs type channel
C_TCHH_ACCHH_1	LOGCH	"C_TCHH_ACCHH_1"	first TCH/H + ACCHs type channel
C_TCHH_ACCHH_2	LOGCH	"C_TCHH_ACCHH_2"	second TCH/H + ACCHs type channel
C_FCCH_SCH_BCCH_CCCH	LOGCH	"C_FCCH_SCH_BCCH_CCCH"	FCCH_SCH_BCCH_CCCH type channel
C_BCCH_CCCH_2	LOGCH	"C_BCCH_CCCH_2"	second BCCH_CCCH type channel
C_BCCH_CCCH_3	LOGCH	"C_BCCH_CCCH_3"	third BCCH_CCCH type channel
C_BCCH_CCCH_4	LOGCH	"C_BCCH_CCCH_4"	fourth BCCH_CCCH type channel
C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4	LOGCH	"C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4"	combined CCCH type channel
C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4	LOGCH	"C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4"	combined CCCH type channel for cell broadcast SM
C_SDCCH8_SACCHC8_1	LOGCH	"C_SDCCH8_SACCHC8_1"	first SDCCH/8 type channel
C_SDCCH8_SACCHC8_2	LOGCH	"C_SDCCH8_SACCHC8_2"	second SDCCH/8 type channel
C_SDCCH8_SACCHC8_3	LOGCH	"C_SDCCH8_SACCHC8_3"	3rd SDCCH/8 type channel
C_S0	SN	'000'B	time slot 0
C_S2	SN	'010'B	time slot 2
C_S3	SN	'011'B	time slot 3
C_S4	SN	'100'B	time slot 4
C_S6	SN	'110'B	time slot 6
C_SAVE	INTEGER	0	the OC_SaveAndProc saves input values
C_PROC	INTEGER	1	the OC_SvaAndProc processes saved values
C_RETRV	INTEGER	1	
C_U1	CCSTATE	1	CC state U1
C_U3	CCSTATE	3	CC state U3
C_U4	CCSTATE	4	CC state U4
C_U6	CCSTATE	6	CC state U6
C_U7	CCSTATE	7	CC state U7

C_U8	CCSTATE	8	CC state U8
C_U9	CCSTATE	9	CC state U9
C_U10	CCSTATE	10	CC state U10
C_U11	CCSTATE	11	CC state U11
C_U12	CCSTATE	12	CC state U12
C_U19	CCSTATE	19	CC state U19
C_U26	CCSTATE	26	CC state U26
C_NxtButOne	SENDINGMODE	1	send the second message on the next but one paging subblock
C_FmrAGB	SENDINGMODE	2	send the second message on the former access grant block
C_BfReOcc	SENDINGMODE	3	send the second message before the MS's original paging subchannel re-occurs, but later than the next paging block of that CCCH
C_NxtButOneNxt	SENDINGMODE	4	nothing is sent in the next but one paging sub block, then send the second message in the next paging subblock of the MS's paging subchannel
C_CMServiceTypeE	CMSVTYPE	'0010'B	CM Service Type for emergency call.
C_Telephony	IA5String	"C_Telephony"	telephony service (TS11)
C_EmgCall	IA5String	"C_EmgCall"	emergency call service (TS12)
C_AltSpchFax	IA5String	"C_AltSpchFax"	alternate speech and G3 fax service (TS61)
C_AutoFax	IA5String	"C_AutoFax"	automatic G3 fax service (TS62)
C_Async300	IA5String	"C_Async300"	data circuit duplex async. 300 bit/s service (BS21)
C_Async1200	IA5String	"C_Async1200"	data circuit duplex async. 1200 bit/s service (BS22)
C_Async120075	IA5String	"C_Async120075"	data circuit duplex async. 1200/75 bit/s service (BS23)
C_Async2400	IA5String	"C_Async2400"	data circuit duplex async. 2400 bit/s service (BS24)
C_Async4800	IA5String	"C_Async4800"	data circuit duplex async. 4800 bit/s service (BS25)
C_Async9600	IA5String	"C_Async9600"	data circuit duplex async 9600 bit/s service (BS26)
C_Sync1200	IA5String	"C_Sync1200"	data circuit duplex sync. 1200 bit/s service (BS31)
C_Sync2400	IA5String	"C_Sync2400"	data circuit duplex sync. 2400 bit/s service (BS32)
C_Sync4800	IA5String	"C_Sync4800"	data circuit duplex sync. 4800 bit/s service (BS33)
C_Sync9600	IA5String	"C_Sync9600"	data circuit duplex sync 9600 bit/s service (BS34)
C_PAD300	IA5String	"C_PAD300"	PAD access 300 bit/s service (BS41)
C_PAD1200	IA5String	"C_PAD1200"	PAD access 1200 bit/s service (BS42)
C_PAD120075	IA5String	"C_PAD120075"	PAD access 1200/75 bit/s service (BS43)
C_PAD2400	IA5String	"C_PAD2400"	PAD access 2400 bit/s service (BS44)
C_PAD4800	IA5String	"C_PAD4800"	PAD access 4800 bit/s service (BS45)
C_PAD9600	IA5String	"C_PAD9600"	PAD access 9600 bit/s service (BS46)
C_Packet2400	IA5String	"C_Packet2400"	packt access 2400 bit/s service (BS51)
C_Packet4800	IA5String	"C_Packet4800"	packt access 4800 bit/s service (BS52)
C_Packet9600	IA5String	"C_Packet9600"	packt access 9600 bit/s service (BS53)
C_AltSpchData	IA5String	"C_AltSpchData"	alternate speech/data service (BS61)
C_SpchData	IA5String	"C_SpchData"	speech followed data service (BS81)

C_I	INTEGER	1	I command of L 2
C_one	INTEGER	1	one frequency, no hopping
C_Immass	BITSTRING	'000'B	activation for immediate assignment
C_Ass	BITSTRING	'001'B	activation for assignment
C_Asynho	BITSTRING	'010'B	activation for non synchronous handover
C_Synho	BITSTRING	'011'B	activation for synchronous handover
C_Rcv	BITSTRING	'101'B	activation for receiving only
C_LocEndRel	RelMode	'01'O	local end release
C_MaxPwrLvlD	INTEGER	15	Max power level for MS DCS
C_MaxPwrLvlG	INTEGER	19	Max power level for MS GSM
C_rc_conditIEerror	REJCAU	'64'O	reject cause: Conditional IE error
C_rc_congestion	REJCAU	'16'O	reject cause: congestion
C_rc_illegal_ms	REJCAU	'03'O	reject cause: illegal MS
C_rc_illegal_me	REJCAU	'06'O	reject cause: illegal ME
C_rc_imsiunknownhlr	REJCAU	'02'O	reject cause: IMSI unknown in HLR
C_rc_imsiunknownvlr	REJCAU	'04'O	reject cause: IMSI unknown in VLR
C_rc_invalidmaninfo	REJCAU	'60'O	reject cause: invalid mandatory information
C_rc_LAnotallowed	REJCAU	'0C'O	reject cause: LocationArea is not allowed
C_rc_networkfailure	REJCAU	'11'O	reject cause: network failure
C_rc_notidentified	REJCAU	'26'O	reject cause: can not be identified
C_rc_plmn_not	REJCAU	'0B'O	reject cause: PLMN not allowed
C_rc_protocolerror	REJCAU	'6F'O	reject cause: Protocol error unspecified
C_rc_reqservoptnotsub	REJCAU	'21'O	reject cause: requested service option not subscribed
C_rc_roamingnot	REJCAU	'0D'O	reject cause: Roaming not allowed
C_Restablishment	B_1	'0'B	Restablishment allowed
C_noRestablishment	B_1	'1'B	No restablishment allowed
C_SlotNU	SN	'111'B	Slot Number not used
C_CellReselectHys12	INTEGER	6	Cell Reselection Hysteresis value 12db
C_CellReselectHys4	INTEGER	2	Cell Reselection Hysteresis value 4db
C_RI_alternate	B_8	'11010001'B	Bcap repeat indicator for alternate (circular selection)
C_RI_follow	B_8	'11010011'B	Bcap repeat indicator for successive (sequential selection)
C_SIMIn	BOOLEAN	FALSE	SIM is always inserted in a test case
C_SIMneedRmv	BOOLEAN	TRUE	Removal of SIM is needed in a test case (SIM is not always inserted.)
C_Yes	BOOLEAN	TRUE	
C_No	BOOLEAN	FALSE	
C_Start_Imsi	INTEGER	1	Test case body starts without TMSI
C_Start_Tmsi	INTEGER	2	Test case body starts with Tmsi_01
C_Start_TmsiOff	INTEGER	5	Test case body starts with Tmsi_01. MS has been deactivated.

C_Start_Rej	INTEGER	3	Test case body starts with LAI deleted. MS has been deactivated.
C_Start_PLMNnot	INTEGER	4	Test case body starts with PLMN not allowed
C_Test_fh	INTEGER	1	Test case with possibly frequency hopping
C_Test_nfh	INTEGER	2	Test case with no frequency hopping
C_Test_egsm	INTEGER	3	Test case for egsm
C_Test_cbms	INTEGER	4	Test case for CBMS
C_normal_updating	B_2	'00'B	normal location updating
C_periodic_updating	B_2	'01'B	periodic location updating
C_normalOrperiodic	B_2	'0?'B	periodic or normal location updating
C_imsi_attach	B_2	'10'B	location updating IMSI attach
C_norm_period_attach	B_2	'??'B	imsi attach or periodic location updating or normal location updating
C_normal_paging	B_2	'00'B	normal page mode
C_extended_paging	B_2	'01'B	extended paging
C_reorg_paging	B_2	'10'B	paging reorganisation
C_sab_paging	B_2	'11'B	same as before paging
C_non_synchronised	B_2	'00'B	non-synchronised
C_synchronised	B_2	'01'B	synchronised
C_pre_synchronised	B_2	'10'B	pre-synchronised
C_pseudo_synchronised	B_2	'11'B	pseudo-synchronised
C_not_report_otd	B_1	'0'B	not report observed time difference
C_report_otd	B_1	'1'B	report observed time difference
C_Tzone0	TZONES	0	Time zone 0 (used in TC_34_2_1, TC_34_2_5_3)
C_Tzone1	TZONES	4	Time zone 4 (used in TC_34_2_2, TC_34_2_7)
C_Tzone2	TZONES	10	Time zone 10 (used in TC_34_2_3)
C_Tzone3	TZONES	15	Time zone 15 (used in TC_34_2_4)
C_Tzone4	TZONES	25	Time zone 25 (used in TC_34_2_5_1)
C_Tzone5	TZONES	35	Time zone 35 (used in TC_34_2_5_2)
C_Tzone6	TZONES	45	Time zone 45 (used in TC_34_2_8)
C_T_AssCmd	INTEGER	900	600ms + 300ms(tolerance)
C_T_Wait1stChReq	INTEGER	35000	max time for full BCCH data decoding
C_T_mrsrp	INTEGER	1500	3 times time btw 2 msr rpt.
C_T_T3124	INTEGER	675	T3124(675ms)
C_T_T3211plus	INTEGER	20000	20s
C_T_T3212	INTEGER	360000	T3212(6min)
C_T_T3212min	INTEGER	345000	T3212(6min) - 15sec
C_T_T3212dif	INTEGER	15000	Difference between T3212 and T3212min
C_T_T3210min	INTEGER	18000	90 % of timer T3210 in ms
C_T_T3210	INTEGER	20000	timer T3210 in ms
C_T_T3211min	INTEGER	13500	90% T3211 in ms
C_T_T3230min	INTEGER	13500	90% T3230 in ms
C_T_T3240min	INTEGER	9000	90% T3240 in ms
C_T_T3240tol	INTEGER	2000	20% T3240 in ms
C_T_Wait	INTEGER	5000	5s. T3126max
C_T_Wait1	INTEGER	1000	1s
C_T_Wait_FAC	INTEGER	1500	1.5s Facility response time (1s) + tolerance (0.5s)
C_release_time	INTEGER	11	release timer value
C_RegCFNRy	INTEGER	1	registration of call forwarding service for CFNRy (speech)
C_RegCFU	INTEGER	2	registration of call forwarding

			service for CFU (all facsimile)
C_RegCF	INTEGER	3	registration of call forwarding service for CF (all facsimile)
C_ErsCFC	INTEGER	4	erasure of call forwarding service for CFC (all facsimile)
C_ErsCFNRc	INTEGER	5	erasure of call forwarding service for CFNRc (all basic services)
C_ErsCFNRy	INTEGER	6	erasure of call forwarding service for CFNRy (all facsimile)
C_ActCF	INTEGER	7	activation of call forwarding service for CF (all synchronous services)
C_ActCFU	INTEGER	8	activation of call forwarding service for CFU (all basic services)
C_DeactCFC	INTEGER	9	deactivation of call forwarding service for CFC (speech)
C_DeactCFNRc	INTEGER	10	deactivation of call forwarding service for CFNRc (all facsimile)
C_InterrogCFC	INTEGER	11	interrogation of call forwarding service for CFC (Speech)
C_InterrogCFB	INTEGER	12	interrogation of call forwarding service for CFB (all basic services)
C_NotifyCFB	INTEGER	13	notification of call forwarding service for CFB (incoming call is forwarded)
C_NotifyCFU	INTEGER	14	notification of call forwarding service for CFU (provisioned, registered, active)
C_NotifyCFC	INTEGER	15	notification of call forwarding service for CFC (provisioned, registered, active)
C_NotifyCFNRc	INTEGER	16	notification of call forwarding service for CFNRc (Ms not reachable)
C_RegPswd	INTEGER	17	Registration of password for all call barring services
C_ActBOAC	INTEGER	18	Activation of BOAC
C_ActBICRoam	INTEGER	19	Activation of BICRoam
C_ActBOIC	INTEGER	20	Activation of BOIC
C_ActBAIC	INTEGER	21	Activation of BAIC
C_DeactBO	INTEGER	22	Deactivation of BO
C_DeactBI	INTEGER	23	Deactivation of BI
C_DeactBOICExHC	INTEGER	24	Deactivation of BOICExHC
C_NotifyBI	INTEGER	25	Notify of BI
C_InterrogBOIC	INTEGER	26	Interrogation of BOIC
C_InterrogBOICExHC	INTEGER	27	Interrogation of BOICExHC
C_Full	IA5String	"F"	full rate channel
C_Half	IA5String	"H"	half rate channel
C_nirr_dontcare	B_1	'?'B	NIRR no meaning or requested
C_Rchr_dontcare	B_2	'??'B	radio channel requirement not checked
C_Rchr_Full	B_2	'01'B	radio channel requirement : full rate
C_Rchr_DualForDualH	B_2	'1?'B	radio channel requirement : dual rate support/half rate preferred or dual rate support/full rate preferred
C_Rchr_DualForDualH0	B_2	'10'B	radio channel requirement : dual rate support/half rate preferred or dual rate support/full rate preferred
C_StartingTimeHO	INTEGER	238	Starting time 238 (1.1s), in TC_26_6_5_1, TC_26_6_5_4_2
C_StrT_1000	INTEGER	1000	Starting time 1000, in TC_26_6_13_1
C_Norm	OCTETSTRING	'00'O	Establish mode: normal
C_Speech	B_3	'000'B	Info. Transfer Cap. speech
C_3100Hz	B_3	'010'B	Info. Transfer Cap. 3.1 kHz

C_UDI	B_3	'001'B	Info. Transfer Cap. UDI
C_FAX3	B_3	'011'B	Info. Transfer Cap. FAX3
C_SDUintegrity	B_2	'00'B	Structure SDU integrity
C_Unstructured	B_2	'11'B	Structure Unstructured
C_struc_dontchk	B_2	'??'B	Structure not checked
C_nirr_nomeaning	B_1	'0'B	NIRR no meaning
C_no_rate_adaption	B_2	'00'B	Rate adaption no
C_rate_adaption_V110	B_2	'01'B	Rate adaption V110
C_rate_adaption_X31	B_2	'10'B	Rate adaption X31 flag
C_sacp_NA	B_3	'001'B	Signaling access NA (sec.11.8.1.5.1)
C_I440_450	B_3	'001'B	Signaling access I440/450
C_X21	B_3	'010'B	Signaling access X21
C_X28_unui	B_3	'100'B	Signaling access X28 U NUI
C_X28_nond	B_3	'101'B	Signaling access X28 non dedicated
C_X32	B_3	'110'B	Signaling access X32
C_Synchronous	B_1	'0'B	Synchronous
C_Asynchronous	B_1	'1'B	Asynchronous
C_stopbit_NA	B_1	'0'B	Number of Stop bits NA
C_stopbit_dontcare	B_1	'?'B	Number of Stop bits not care
C_databit_NA	B_1	'1'B	Number of Data bits not care
C_databit_dontcare	B_1	'?'B	Number of Data bits NA
C_300bs	B_4	'0001'B	User rate 300 bs
C_1200bs	B_4	'0010'B	User rate 1200 bs
C_120075bs	B_4	'0111'B	User rate 1200/75 bs
C_2400bs	B_4	'0011'B	User rate 2400 bs
C_4800bs	B_4	'0100'B	User rate 4800 bs
C_9600bs	B_4	'0101'B	User rate 9600 bs
C_parity_NA	B_3	'011'B	Parity NA
C_parity_dontcare	B_3	'???'B	Parity not care
C_ir_8kbs	B_2	'10'B	Intermediate rate 8 Kbs
C_ir_16kbs	B_2	'11'B	Intermediate rate 16 Kbs
C_ir_8or16kbs	B_2	'1?'B	Intermediate rate 8 or 16 Kbs
C_transparent	B_2	'00'B	Connection element T
C_notransparent	B_2	'01'B	Connection element NT
C_BothT	B_2	'10'B	Connection element both T
C_BothNT	B_2	'11'B	Connection element both NT
C_modemt_none	B_5	'00000'B	no modem
C_modemt_V21	B_5	'00001'B	modem V21
C_modemt_V22	B_5	'00010'B	modem V22
C_modemt_V22bis	B_5	'00011'B	modem V22bis
C_modemt_V23	B_5	'00100'B	modem V23
C_modemt_V32	B_5	'00110'B	modem V32
C_ISO6429	B_5	'01000'B	Inband flow control
C_COPnoFLCT	B_5	'01100'B	No flow control
C_X25_flct	B_5	'00110'B	X25 flow control
C_Outband	FLWCNTL	0	outband flow control
C_Inband	FLWCNTL	1	outband flow control
C_Nocontrol	FLWCNTL	2	outband flow control
C_uil2p_dontcare	B_5	'?????'B	not checked
C_HSN_0	INTEGER	0	cyclic hopping
C_modemt_ce	INTEGER	1	flag value for OC_ModifyBcap
C_modemt_Tlist	INTEGER	2	flag value for OC_ModifyBcap
C_modemt_NTlist	INTEGER	3	flag value for OC_ModifyBcap
C_modemList_ce	INTEGER	4	flag value for OC_ModifyBcap
C_modemList_Tlist	INTEGER	5	flag value for OC_ModifyBcap
C_modemList_NTlist	INTEGER	6	flag value for OC_ModifyBcap
maxAddressLength	Asn1Integer	20	
maxISDN_AddressLength	Asn1Integer	9	
maxISDN_SubaddressLength	Asn1Integer	21	
maxNumOfBasicServiceGroups	Asn1Integer	13	
maxNumOfCUG	Asn1Integer	10	
maxNumberOfSegmentsPerD	Asn1Integer	8191	

ataInterval			
maxSignalInfoLength	Asn1Integer	200	
maxUSSD_StringLength	Asn1Integer	160	
max10TimesIncrement	Asn1Integer	8191	
max10TimesIncrementPerDataInterval	Asn1Integer	8191	
max10TimesInitialTime	Asn1Integer	8191	
max100TimesScalingFactor	Asn1Integer	8191	
max10TimesTimeInterval	Asn1Integer	8191	
max10TimesUnitsPerTime	Asn1Integer	8191	
Detailed Comments:			

Test case variable declarations

Test Case Variable Declarations			
Variable Name	**Type**	**Value**	**Comments**
TCV_CC	BOOLEAN	TSPC_Serv_TS11 OR TSPC_Serv_TS12 OR TSPC_Serv_TS61 OR TSPC_Serv_TS62 OR TSPC_Serv_BS21 OR TSPC_Serv_BS22 OR TSPC_Serv_BS23 OR TSPC_Serv_BS24 OR TSPC_Serv_BS25 OR TSPC_Serv_BS26 OR TSPC_Serv_BS31 OR TSPC_Serv_BS32 OR TSPC_Serv_BS33 OR TSPC_Serv_BS34 OR TSPC_Serv_BS41 OR TSPC_Serv_BS42 OR TSPC_Serv_BS43 OR TSPC_Serv_BS44 OR TSPC_Serv_BS45 OR TSPC_Serv_BS46 OR TSPC_Serv_BS51 OR TSPC_Serv_BS52 OR TSPC_Serv_BS53 OR TSPC_Serv_BS61 OR TSPC_Serv_BS81	
TCV_AssCmd	ASS_CMD_PDU		to hold ASSIGNMENT COMMAND PDU
TCV_B	BCAP		to hold bearer capability
TCV_Bcap1	BCAP		to hold bearer capability
TCV_BcapMT1	BCAP		to hold BC for MT call
TCV_BcapMO1	BCAP		to hold BC for MO call
TCV_Bcap2	BCAP		to hold bearer capability
TCV_BcapMT2	BCAP		to hold BC for MT call
TCV_bs_ag_blks_res	INTEGER		
TCV_bs_cc_chans	INTEGER		
TCV_bs_pa_mfrms	INTEGER		
TCV_CalledNum	CDPN		to hold called party number
TCV_CallProc	CALL_PROC_PDU		to hold CALL PROCEEDING PDU
TCV_CallCfm	CALL_CO_PDU		to hold CALL COMFIRM PDU
TCV_Cau	CAU		to hold cause
TCV_Cau0	CAU		to hold cause
TCV_CAgsm	CCHD		To compute Cell Channel Descriptor for gsm
TCV_CAdcs	CCHD		To compute Cell Channel Descriptor for dcs
TCV_CB1	OCTETSTRING		store SMSCB block data
TCV_CB2	OCTETSTRING		store SMSCB block data
TCV_CB3	OCTETSTRING		store SMSCB block data
TCV_CB4	OCTETSTRING		store SMSCB block data
TCV_CE	B_2		
TCV_Ccchg	INTEGER		CCCH_GROUP
TCV_Ccd0A	CCD		a copy of control channel description in use for cell A
TCV_Ccd0B	CCD		a copy of control channel description in use for cell B
TCV_Ccd0C	CCD		a copy of control channel description in use for cell C
TCV_Ccd0D	CCD		a copy of control channel description in use for cell D
TCV_Ccd0E	CCD		a copy of control channel description in use for cell E
TCV_Ccd0F	CCD		a copy of control channel description in use for cell F
TCV_Ccd0G	CCD		a copy of control channel description in use for cell G
TCV_Ccd0H	CCD		a copy of control channel

			description in use for cell H
TCV_cchdescr	OCTETSTRING		Frequency list of Cell Channel Descr, used in EGSM
TCV_CCst	CST		to hold call status
TCV_ch	LOGICCH	"dummy"	to hold logic channel
TCV_ch1	LOGICCH	"dummy"	to hold logic channel
TCV_sacch	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_B	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_C	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_D	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_E	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_F	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_G	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_H	LOGICCH	"dummy"	to hold logic channel
TCV_sacch8	LOGICCH	"dummy"	to hold logic channel
TCV_sacchTch	LOGICCH	"dummy"	to hold logic channel
TCV_sacchTch1	LOGICCH	"dummy"	to hold logic channel
TCV_sacchTch_H	LOGICCH	"dummy"	to hold logic channel
TCV_chTch	LOGICCH	"dummy"	to hold logic channel
TCV_chTch1	LOGICCH	"dummy"	to hold logic channel
TCV_chSms	LOGICCH	"dummy"	to hold logic channel
TCV_agch	LOGICCH	"dummy"	downlink access grant channel
TCV_WorkingCh	LOGICCH	"dummy"	to hold logic channel
TCV_cellid	CellID		to hold the cell identifier
TCV_Char	IA5String		to hold the IA5 char. corresponding to a DTMF tone
TCV_chd1	CHD		to hold channel description IE
TCV_cchd1	CCHD		to hold cell channel description IE
TCV_ChMod	CHMOD		to hold channel mode IE
TCV_ChModb	CHMOD		to hold channel mode IE for second mode in dual mode services
TCV_Tchtype	CH_TDMA	'00001'B	Traffic channel type and TDMA offset in channel description IE
TCV_chtype	CH_TDMA		to channel type and TDMA offset in channel description IE
TCV_cksn	BITSTRING		to hold the CKSN.
TCV_Cnt	INTEGER		general loop counter
TCV_Cnt1	INTEGER		general loop counter
TCV_Cnt2	INTEGER		general loop counter
TCV_Cntstart	BOOLEAN	FALSE	general loop counter start decision
TCV_Cntref	INTEGER		Reference for general loop counter
TCV_CphAlg	BITSTRING		ciphering algorithm
TCV_CphKey	BITSTRING		ciphering key
TCV_CphMd	CPHMS		ciphering mode setting
TCV_CPDataRetx	INTEGER		Number of CP Data retransmissions for SMS
TCV_ChRate	IA5String	C_Full	to hold channel rate
TCV_ChRate1	IA5String	C_Full	to hold channel rate
TCV_counter_c	INTEGER		loop counter c
TCV_counter_k	INTEGER		loop counter k
TCV_Fk	INTEGER		to hold an interval between events
TCV_flist	OCTETSTRING		Frequency List, used in EGSM for IE "frequency list" or "frequency short list"
TCV_flistl	LENGTH		Length of Frequency List, used in EGSM for IE "frequency list" or "frequency short list"
TCV_Fn	FN		to hold the frame number (T1' , T2, T3)
TCV_Fn1	FN		to hold the frame number (T1' , T2, T3)
TCV_FnAss	FN		to hold the frame number in AssCmd statement
TCV_FnBurst	FN		to hold the frame number of first burst of handover access

TCV_FollowingOctets	OCTETSTRING		Contains the octets following the invoke id
TCV_HoaccPara	HOACC_PARA		Handover access timing and power
TCV_Horf	HORF		Variable for HO reference in HO-Messages of HO cases.
TCV_HoCMD	HO_CMD_PDU		to hold handover PDU
TCV_Hrf	HORF		to hold handover reference
TCV_iel	LENGTH		Length of IE
TCV_kcnt	INTEGER		loop counter
TCV_K	INTEGER		
TCV_L1Head0	L1HD		to hold L 1 header
TCV_L1Head	L1HD		to hold L 1 header
TCV_lac	OCTETSTRING		to hold the lac. 2 octets
TCV_lacunknown	OCTETSTRING	'????'O	to hold a wild card lac when the lac is unknown
TCV_mccunknown	OCTETSTRING	'????'O	to hold a wild card mcc value
TCV_plmnunknown	OCTETSTRING	'??'O	to hold a wild card plmn value
TCV_n	INTEGER		
TCV_Neci	BITSTRING		
TCV_M	INTEGER		counter
TCV_maio	MAIO		to hold MAIO
TCV_mae1	BITSTRING		Mobile Allocation in EGSM. It includes the 1.octet of MA
TCV_mae2	BITSTRING		Mobile Allocation in EGSM. It includes the 2.octet of MA
TCV_Max	BITSTRING		to hold Max-retrans bitstring
TCV_MemCapExcd	BOOLEAN		RP error Memory Capacity Exceeded was sent
TCV_MsrRes	MSRR		to hold measurement results
TCV_Mt	BITSTRING		to hold message type
TCV_Mt1	BITSTRING		to hold message type
TCV_Modify	MODIFY_PDU		To hold Modify message
TCV_Null	BOOLEAN		to collect useless result from some test suite operations
TCV_PgCh	LOGICCH	"dummy"	logical channel for paging
TCV_RaCh	LOGICCH	"dummy"	random access channel parameter.
TCV_Pgg	PGG		paging group
TCV_PreviousOctets	OCTETSTRING		Contains the octets previous to the invoke id
TCV_ProtErrorUnspec	BOOLEAN		RP error Protocol Error Unspecified was sent
TCV_Pwrlvl	BITSTRING		to hold the power level
TCV_Pwrlvl_ho	BITSTRING		power level for HO_CMD, used in HO cases.
TCV_Rchr	B_2		to hold Radio channel requirement
TCV_Res	BOOLEAN		to hold the result of a test suite operation
TCV_Rpmr	MR		RP message reference for SMS
TCV_Rqr	RQR		to hold the request reference
TCV_Rqr9	RQR		to hold the request reference differing from TCV_Rqr
TCV_Rqr10	RQR		to hold the request reference differing from TCV_Rqr, TCV_Rqr9
TCV_Rqr11	RQR		to hold the request reference differing from TCV_Rqr, TCV_Rqr9, TCV_Rqr10
TCV_Rr	BITSTRING		to hold the random request reference
TCV_Rr1	BITSTRING		to hold the random reference
TCV_Service	SERVICES		to hold basicservice selected
TCV_Service1	SERVICES		to hold basicservice selected
TCV_Setup_mo	SETUP_MO_PDU		to hold the SETUP PDU (MO)
TCV_Setup_mo1	SETUP_MO_PDU		to hold the SETUP PDU (MO)
TCV_Setup_mt	SETUP_MT_PDU		to hold the SETUP PDU (MT)

TCV_Setup_mt1	SETUP_MT_PDU		to hold the 2nd SETUP PDU (MT)
TCV_2ndtest	BOOLEAN		2nd test required in TC_11_1_1
TCV_Esetup	ESETUP_PDU		to hold the emergency SETUP PDU
TCV_Esetup1	ESETUP_PDU		to hold the emergency SETUP PDU
TCV_Ecall	BOOLEAN	FALSE	it is set to TRUE, if the service under test is emergency call
TCV_slot	SN		To hold the default slot used during the entire TC.
TCV_slt2	SN		To hold a second working slot number.
TCV_Sres	OCTETSTRING		to hold the SRES returned
TCV_sysinf5	SYSINFO5_PDU		to hold the SysInfo5 PDU
TCV_sysinfo5	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell A
TCV_sysinfo5bis	SYSINFO5bis_PDU		to hold the SysInfo5bis PDU for cell A
TCV_sysinfo5_B	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell B
TCV_sysinfo5_C	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell c
TCV_sysinfo5_D	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell D
TCV_sysinfo5_E	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell E
TCV_sysinfo5_F	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell F
TCV_sysinfo5_G	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell G
TCV_sysinfo5_H	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell H
TCV_sysinfo5bis_A	SYSINFO5bis_PDU		to hold the SysInfo5bis PDU for cell A(TC_26_10_2_4_1)
TCV_sysinfo5bis_B	SYSINFO5bis_PDU		to hold the SysInfo5bis PDU for cell B
TCV_sysinf6	SYSINFO6_PDU		to hold the SysInfo6 PDU
TCV_sysinfo6	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell A
TCV_sysinfo6_B	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell B
TCV_sysinfo6_C	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell C
TCV_sysinfo6_D	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell D
TCV_sysinfo6_E	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell E
TCV_sysinfo6_F	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell F
TCV_sysinfo6_G	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell G
TCV_sysinfo6_H	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell H
TCV_temp	INTEGER		to hold results of expressions that are to be passed as parameters
TCV_temp1	INTEGER		to hold temporary integer values
TCV_temp2	INTEGER		to hold temporary integer values
TCV_ti_v	TI_V		to hold the transaction id. value
TCV_ti_v_2	TI_V		to hold the transaction id. value for another transaction
TCV_T	INTEGER		to hold Tx-integer chosen randomly
TCV_T_TwiceTC1M	INTEGER	TSPX_TC1M * 2000	Twice TC1M in ms
TCV_chdescr_arfcn	INTEGER		ARFCN carrier of actual cell
TCV_tch_arfcn	INTEGER		ARFCN carrier for traffic channel
TCV_asscmd_ts	BITSTRING		Time slot for channel description in assignment cmd.
TCV_ts	BITSTRING		Time slot for channel description in handover cmd.
TCV_ia_ts	BITSTRING		Time slot for channel description

			in immediate assignment
TCV_tsc	TSC		To hold the default training sequence used during the entire TC.
TCV_Td	OCTETSTRING		to hold timing difference
TCV_Time	INTEGER		to hold the measured value of T310 or T305 or T308
TCV_TI	TI		to hold the transaction ID for MO calls in sending evets
TCV_TI0	TI		to hold the transaction ID for MO calls in receiving evets
TCV_TI1	TI		to hold the transaction ID
TCV_TI2	TI		to hold the transaction ID
TCV_TI3	TI		to hold the transaction ID
TCV_tmp	INTEGER		temporary integer variable
TCV_Tx	BITSTRING		to hold Tx bitstring
TCV_S	INTEGER		to hold the S parameter
TCV_Upd	BOOLEAN		to hold the information whether the MS is updated or not,.
TCV_UssdString	IA5String		String for USS Data
TCV_UssdString1	IA5String		String for USS Data
TCV_UssdString2	IA5String		String for USS Data
TCV_InvkId	OCTETSTRING		to hold SS transaction ID
TCV_InvkId0	OCTETSTRING		to hold SS transaction ID
TCV_InvkId1	OCTETSTRING		to hold SS transaction ID
TCV_Comp	Components		to hold SS Components
TCV_Strt	STRT		to hold starting time
TCV_CBch	LOGICCH	"dummy"	Cell Broadcast channel
TCV_Tpmr	MR		TP message reference
TCV_TPOA1	BCDN		TP originating address digits
TCV_TPOA2	BCDN		TP originating address digits
TCV_TPDA	BCDN		TP destination address digits
TCV_RPOA_MT	BCDN		RP originating address digits for MT short messages
TCV_RPOA_MO	BCDN		RP originating address digits for MO short messages
TCV_RPOA1	BCDN		RP originating address digits for MT short messages
TCV_RPOA2	BCDN		RP originating address digits for MT short messages
TCV_RPDA_MT	BCDN		RP destination address digits for MT short messages
TCV_RPDA_MO	BCDN		RP destination address digits for MO short messages
TCV_SMTypeM	INTEGER		Replace short message type
TCV_SMTypeN	INTEGER		Replace short message type
TCV_SMcntns	OCTETSTRING		short message contents
TCV_SMSCBpack	SMSCBpack		CB message contents
TCV_freq	FRQPARA		to hold Freq type constraints
TCV_srv	SERVICES		to hold MT service
TCV_ur	B_4		to hold user rate
TCV_itc	B_3		to hold information transfer capability
TCV_sacp	B_3		to hold service access protocol
TCV_ce	B_2		to hold connection element
TCV_ce1	B_2		to hold connection element
TCV_ce3	B_2		to hold connection element
TCV_sa	B_1		to hold synchronous/asynchronous
TCV_ir	B_2		to hold intermediate rate
TCV_uil1p	B_2		to hold user information layer 1 protocol
TCV_modemt	B_5		to hold modem type
TCV_MODEM	B_5		
TCV_ur2	B_4		to hold user rate for second configuration
TCV_itc2	B_3		to hold information transfer capability for second configuration

TCV_sacp2	B_3		to hold service access protocol for second configuration
TCV_strc	B_2		to hold structure
TCV_sa2	B_1		to hold synchronous/asynchronous for second configuration
TCV_bcapListIndicator	INTEGER		flag for OC_ModifyBcap
TCV_supported	BOOLEAN	FALSE	used for indicating a selected service is declared supported or not
TCV_SpecialCase	BOOLEAN	FALSE	used for indicating whether a test case tests non-supported service
TCV_expta	INTEGER		Expected timing advance of access bursts in bits
TCV_expdbm	INTEGER		Expected power level of access bursts in dBm
TCV_mindbm	INTEGER		Minimum received power level of access bursts in dBm
TCV_maxdbm	INTEGER		Maximum received power level of access bursts in dBm
Detailed Comments:			

PCO declarations

PCO Declarations			
PCO Name	**Type**	**Role**	**Comments**
L	SAP0_3	LT	
Detailed Comments:	1. The PCO consists of multiple SAPs: the SAP0 and SAP3. 2. The lower tester (LT) is the user of the data link layer service. 3. The SAP0 at the lower tester controlling and observing the exchange of CC, MM, RR and SS PDUs (messages) on the DCCH, SACCH and/or RACH, BCCH, CCCH channels. 4. The SAP3 at the lower tester controlling and observing the exchange of SMS PDUs (messages) on the DCCH or SACCH channels.		

Timer declarations

Timer Declarations			
Timer Name	**Duration**	**Unit**	**Comments**
T_dly		ms	general purpose delay timer
T_dly1		ms	general purpose delay timer
T_dly2		ms	general purpose delay timer
T_dlyAss	5000	ms	AssCmd timer, 1)
T_guard	300	s	guard timer
T_release	C_release_time	s	release timer
Detailed Comments:	1) In order to give to the lower tester enough time to dequeue and to send out the call SETUP and the other call establishment related messages in the sending buffer before starting sending of ASSINGMENT COMMAND message, the default value 5s is used in the test step AssCh_complete. In the test step AssCh_failure the timer value is set to 4s. For the TC_26_6_4_1, 600ms is used.		

ASP type definitions

TTCN ASP Type definitions

ASP Type Definition		
ASP Name:	DL_EstRq (DL_ESTABLISH_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the establishment of multiple frame operation (L3 -> L2). The normal establishment procedure is initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstIn (DL_ESTABLISH_INDICATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation (L2 -> L3). The normal establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInCmsRq (DL_ESTABLISH_INDICATION_CM_SERVICE_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the MM CM SERVICE REQUEST message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	CMS_RQ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInLupRq (DL_ESTABLISH_INDICATION_LOCATION_UPDATING_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the MM LOCATION UPDATING REQUEST message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	LUP_RQ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInImsidIn (DL_ESTABLISH_INDICATION_IMSI_DETACH_INDICATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the MM IMSI DETACH INDICATION message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	IMSID_IN_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInPgRes (DL_ESTABLISH_INDICATION_PAGING_RESPONSE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the RR PAGING RESPONSE message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	PG_RES_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInCmreRq (DL_ESTABLISH_INDICATION_CM_REESTABLISHMENT_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the MM CM REESTABLISHMENT REQUEST message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	CMRE_RQ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstCo (DL_ESTABLISH_CONFIRM)	
PCO Type:	SAP0_3	
Comments:	The ASP is used by the L2 to inform the L3 about the establishment of multiple frame link (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_RelIn (DL_RELEASE_INDICATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the termination of an established multiple frame operation or to report an unsuccessful establishment attempt (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
release_mode	RelMode	
outstanding_indicator	BOOLEAN	
fn	FN	
Detailed Comments:	The outstanding indicator indicates whether or not there are outstanding acknowledgements or unsolved DL_DATA_REQUEST primitives.	

ASP Type Definition		
ASP Name:	DL_RacInChRq (DL_RANDOM_ACCESS_INDICATION_CHANNEL_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the arrival of an RR CHANNEL REQUEST message (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
fn	FN	
msg	CH_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_RacInHoacc (DL_RADOM_ACCESS_INDICATION_HANDOVER_ACCESS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the arrival of an RR HANDOVER ACCESS message (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HOACC_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmass (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	IMMASS_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmass_sp (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT message using unacknowledged operation (L3 -> L2) on the MS paging channel.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	IMMASS_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmassx (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT_EXTENDED)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT EXTENDED message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	IMMASSX_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmassx_sp (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT_EXTENDED)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT EXTENDED message using unacknowledged operation (L3 -> L2) on the MS paging channel.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	IMMASSX_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmassRej (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT REJECT message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	IMMASS_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmassRej_sp (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT REJECT message using unacknowledged operation (L3 -> L2) on the MS paging channel.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	IMMASS_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatInMsrRpt (DL_DATA_INDICATION_MEASUREMENT_REPORT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR MEASUREMENT REPORT message using unacknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MSR_RPT_PDU	
fn	FN	
Detailed Comments:	The ASPs are continuously received during the testing.	

ASP Type Definition		
ASP Name:	DL_UdatRqSMSCBData (DL_UNIT_DATA_REQUEST_SMSCB_DATA)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the SMSCB data using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SMSCB_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqPg1Rq (DL_UNIT_DATA_REQUEST_PAGING_REQUEST_TYPE1)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR PAGING REQUEST TYPE 1 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	PG1_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqPg2Rq (DL_UNIT_DATA_REQUEST_PAGING_REQUEST_TYPE2)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR PAGING REQUEST TYPE 2 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	PG2_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqPg3Rq (DL_UNIT_DATA_REQUEST_PAGING_REQUEST_TYPE3)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR PAGING REQUEST TYPE 3 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	PG3_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSchinfo (DL_UNIT_DATA_REQUEST_SYNCHRONIZATION_CHANNEL_INFORMATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYNCHRONIZATION CHANNEL INFORMATION message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SCHINFO_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo1 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE1)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 1 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO1_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo1_nh (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE1)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 3 message in case of non hopping	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO3_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo2 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE2)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 2 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO2_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo2bis (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE2bis)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 2bis message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO2bis_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo2ter (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE2ter)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 2ter message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO2ter_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo3 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE3)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 3 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO3_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo4 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE4)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 4 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO4_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo5 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE5)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 5 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO5_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo5bis (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE5bis)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 5bis message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO5bis_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo5ter (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE5ter)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 5ter message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO5ter_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo6 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE6)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 6 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO6_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo7 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE7)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 7 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO7_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo8 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE8)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 8 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO8_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatInClmChn (DL_UdatINDICATION_CLASSMARK_CHANGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR CLASSMARK CHANGE message using unacknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CLM_CHN_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAssCmd (DL_DATA_REQUEST_ASSIGNMENT_COMMAND)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR ASSIGNMENT COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ASS_CMD_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInAssCom (DL_DATA_INDICATION_ASSIGNMENT_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR ASSIGNMENT COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ASS_COM_PDU	
fn	FN	
Detailed Comments:	The ASP is a result of the ASP DL_RESUME_REQUEST of the SUT sending the ASSIGNMENT COMPLETE message.	

ASP Type Definition		
ASP Name:	DL_DatInAssfl (DL_DATA_INDICATION_ASSIGNMENT_FAILURE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR ASSIGNMENT FAILURE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ASSFL_PDU	
fn	FN	
Detailed Comments:	The ASP is a result of the ASP DL_RECONNECT_REQUEST of the SUT sending the ASSIGNMEN FAILURE message.	

ASP Type Definition		
ASP Name:	DL_DatInCmsRq (DL_DATA_INDICATION_CM_SERVICE_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM CM SERVICE REQUEST message (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CMS_RQ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqChmmo (DL_DATA_REQUEST_CHANNEL_MODE_MODIFY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CHANNEL MODE MODIFY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CHMMO_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInChmmoAck (DL_DATA_INDICATION_CHANNEL_MODE_MODIFY_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR CHANNEL MODE MODIFY ACKNOWLEDGE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CHMMO_ACK_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqChRel (DL_DATA_REQUEST_CHANNEL_RELEASE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CHANNEL_RELEASE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CH_REL_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqChRelErr (DL_DATA_REQUEST_CHANNEL_RELEASE_ERROR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CHANNEL_RELEASE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CH_REL_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCphmCmd (DL_DATA_REQUEST_CIPHERING_MODE_COMMAND)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CIPHERING_MODE_COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPHM_CMD_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCphmCmdErr (DL_DATA_REQUEST_CIPHERING_MODE_COMMAND_ERROR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the invalid RR CIPHERING_MODE_COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPHM_CMD_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCphmCom (DL_DATA_INDICATION_CIPHERING_MODE_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR CIPHERING MODE COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPHM_COM_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInClmChn (DL_DAT_INDICATION_CLASSMARK_CHANGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR CLASSMARK CHANGE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CLM_CHN_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqClmEnq (DL_DATA_REQUEST_CLASSMARK_ENQUIRY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CLASSMARK ENQUIRY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CLM_ENQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqFrqre (DL_DATA_REQUEST_FREQUENCY_REDEFINITION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR FREQUENCY REDEFINITION message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	FRQRE_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqHoCmd (DL_DATA_REQUEST_HANDOVER_COMMAND)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR HANDOVER COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HO_CMD_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInHoCom (DL_DATA_INDICATION_HANDOVER_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR HANDOVER COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HO_COM_PDU	
fn	FN	
Detailed Comments:	The ASP is a result of the ASP DL_RESUME_REQUEST of the SUT sending the HANDOVER COMPLETE message.	

ASP Type Definition		
ASP Name:	DL_DatInHofl (DL_DATA_INDICATION_HANDOVER_FAILURE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR HANDOVER FAILURE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HOFL_PDU	
fn	FN	
Detailed Comments:	The ASP is a result of the ASP DL_RECONNECT_REQUEST of the SUT sending the HANDOVER FAILURE message.	

ASP Type Definition		
ASP Name:	DL_DatRqPhyinfo (DL_DATA_REQUEST_PHYSICAL_INFORMATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR PHYSICAL INFORMATION message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	PHYINFO_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInRrst (DL_DATA_INDICATION_RR_STATUS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR STATUS message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	RRST_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAuthRej (DL_DATA_REQUEST_AUTHENTICATION_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM AUTHENTICATION REJECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	AUTH_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAuthRq (DL_DATA_REQUEST_AUTHENTICATION_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM AUTHENTICATION REQUEST message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	AUTH_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInAuthRes (DL_DATA_INDICATION_AUTHENTICATION_RESPONSE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM AUTHENTICATION RESPONSE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	AUTH_RES_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCmsAcp (DL_DATA_REQUEST_CM_SERVICE_ACCEPT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM CM SERVICE ACCEPT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CMS_ACP_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCmsRej (DL_DATA_REQUEST_CM_SERVICE_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM CM SERVICE REJECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CMS_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAbrt (DL_DATA_REQUEST_ABORT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM ABORT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ABRT_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqIdRq (DL_DATA_REQUEST_IDENTITY_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM IDENTITY REQUEST message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ID_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInIdRes (DL_DATA_INDICATION_IDENTIFY_RESPONSE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM IDENTITY RESPONSE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ID_RES_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqLupAcp (DL_DATA_REQUEST_LOCATION_UPDATING_ACCEPT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM LOCATION UPDATING ACCEPT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	LUP_ACP_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqLupAcpErr (DL_DATA_REQUEST_LOCATION_UPDATING_ACCEPT_ERR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM LOCATION UPDATING ACCEPT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	LUP_ACP_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqLupRej (DL_DATA_REQUEST_LOCATION_UPDATING_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM LOCATION UPDATING REJECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	LUP_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInMmst (DL_DATA_INDICATION_MM_STATUS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM STATUS message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MMST_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqTmsireCmd (DL_DATA_REQUEST_TMSI_REALLOCATION_COMMAND)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM TMSI REALLOCATION COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	TMSIRE_CMD_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInTmsireCom (DL_DATA_INDICATION_TMSI_REALLOCATION_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM TMSI REALLOCATION COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	TMSIRE_COM_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAlert (DL_DATA_REQUEST_ALERTING)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC ALERTING message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ALERT_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInAlert (DL_DATA_INDICATION_ALERTING)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC ALERTING message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ALERT_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCallCo (DL_DATA_INDICATION_CALL_CONFIRMED)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC CALL CONFIRMED message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CALL_CO_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCallProc (DL_DATA_REQUEST_CALL_PROCEEDING)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC CALL PROCEEDING message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CALL_PROC_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqConn (DL_DATA_REQUEST_CONNECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC CONNECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqConnErr (DL_DATA_REQUEST_CONNECT_ERR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC CONNECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInConn (DL_DATA_INDICATION_CONNECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC CONNECT message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqConnAck (DL_DATA_REQUEST_CONNECT_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC CONNECT ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInConnAck (DL_DATA_INDICATION_CONNECT_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the reception of the CC CONNECT ACKNOWLEDGE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_ACK_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqDisc (DL_DATA_REQUEST_DISCONNECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC DISCONNECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	DISC_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqDiscErr (DL_DATA_REQUEST_DISCONNECT_ERR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC DISCONNECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	DISC_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInDisc (DL_DATA_INDICATION_DISCONNECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC DISCONNECT message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	DISC_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInESetup (DL_DATA_INDICATION_EMERGENCY_SETUP)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC EMERGENCY_SETUP message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ESETUP_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqFac (DL_DATA_REQUEST_FACILITY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC FACILITY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	FAC_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInFac (DL_DATA_INDICATION_FACILITY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC FACILITY message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	FAC_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInHold (DL_DATA_INDICATION_HOLD)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC HOLD message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HOLD_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqHoldAck (DL_DATA_REQUEST_HOLD_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC HOLD ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HOLD_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqModify (DL_DATA_REQUEST_MODIFY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC MODIFY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInModify (DL_DATA_INDICATION_MODIFY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC MODIFY message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqModifyCom (DL_DATA_REQUEST_MODIFY_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC MODIFY COMPLETE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_COM_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqModifyRej (DL_DATA_REQUEST_MODIFY_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC MODIFY REJECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInModifyRej (DL_DATA_INDICATION_MODIFY_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC MODIFY REJECT message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_REJ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqNotify (DL_DATA_REQUEST_NOTIFY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC NOTIFY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	NOTIFY_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqProg (DL_DATA_REQUEST_PROGRESS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC PROGRESS message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	PROG_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqRegister (DL_DATA_REQUEST_REGISTER)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the SS REGISTER message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REGISTER_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInRegister (DL_DATA_INDICATION_REGISTER)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to receive the transmission of the SS REGISTER message using acknowledged operation.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REGISTER_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqRel (DL_DATA_REQUEST_RELEASE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC RELEASE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REL_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInRel (DL_DATA_INDICATION_RELEASE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC RELEASE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REL_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqRelCmp (DL_DATA_REQUEST_RELEASE_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC RELEASE COMPLETE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REL_COM_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInRelCmp (DL_DATA_INDICATION_RELEASE_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC RELEASE COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REL_COM_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqSetup (DL_DATA_REQUEST_SETUP)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC SETUP message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SETUP_MT_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInSetup (DL_DATA_INDICATION_SETUP)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC SETUP message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SETUP_MO_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInStartDtmf (DL_DATA_INDICATION_START_DTMF)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC START_DTMF message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	START_DTMF_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqStartDtmfAck (DL_DATA_REQUEST_START_DTMF_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC START DTMF ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	START_DTMF_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqStartDtmfRej (DL_DATA_REQUEST_START_DTMF_REJCT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC START DTMF REJCT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	START_DTMF_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInStopDtmf (DL_DATA_INDICATION_STOP_DTMF)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC STOP_DTMF message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	STOP_DTMF_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqStopDtmfAck (DL_DATA_REQUEST_STOP_DTMF_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC STOP DTMF ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	STOP_DTMF_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCcst (DL_DATA_REQUEST_CC_STATUS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC STATUS message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CCST_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCcst (DL_DATA_INDICATION_CC_STATUS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC STATUS message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CCST_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCcstEnq (DL_DATA_REQUEST_CC_STATUS_ENQ)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC STATUS_ENQ message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CCST_ENQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqUndefCC (DL_DATA_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to transmit any message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqUndefMM (DL_DATA_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to transmit any message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ID_RES_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqUndefRR (DL_DATA_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to transmit any message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	PART_REL_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqUnknown (DL_DATA_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to transmit any message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CCST_ENQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCpData (DL_DATA_INDICATION_CP_DATA)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the SMS CP DATA message using acknowledged operation (L2 -> L3) for MT.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CP_DATA_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCpData (DL_DATA_REQUEST_CP_DATA)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the SMS CP DATA message using acknowledged operation (L3 -> L2) for MO.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CP_DATA_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCpDataAck (DL_DATA_INDICATION_CPDATA_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the SMS CP DATA message using acknowledged operation (L2 -> L3) .	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPDATA_ACK_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCpDataAck (DL_DATA_REQUEST_CPDATA_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CP DATA ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPDATA_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCpError (DL_DATA_REQUEST_CP_ERROR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the SMS CP ERROR message using acknowledged operation (L3 -> L2) for MO.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPERR_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	MDL_RelRq (MDL_RELEASE_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the local end termiination of a previous established acknowledged mode operation (L3 -> L2). The local end termination procedure is initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
release_mode	RelMode	
Detailed Comments:		

PDU type definitions

TTCN PDU Type definitions

PDU Type Definition		
PDU Name:	ASS_CMD_PDU	
PCO Type:	SAP0_3	
Comments:	RR ASSSIGNMENT COMMAND n -> ms GSM 04.08, 9.1.2	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
ch1d_at	CHD	
pcmd	PCMD	
frql_at	FRQL	
cchd	CCHD	
ch1mod	CHMOD	
ch2d_at	CHD	
ch2mod	CHMOD	
ma_at	MA	
strt	STRT	
frql_bt	FRQL	
ch1d_bt	CHD	
ch2d_bt	CHD	
frqchs_bt	FRQCHS	
ma_bt	MA	
cphms	CPHMS	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ASS_COM_PDU	
PCO Type:	SAP0_3	
Comments:	RR ASSSIGNMENT COMPLETE ms -> n GSM 04.08, 9.1.3	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ASSFL_PDU	
PCO Type:	SAP0_3	
Comments:	RR ASSSIGNMENT FAILURE ms -> n GSM 04.08, 9.1.4	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CHMMO_PDU	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL MODE MODIFY n -> ms GSM 04.08, 9.1.5	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
chd	CHD	
chmod	CHMOD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CHMMO_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL MODE MODIFY ACKNOWLEDGE ms -> n GSM 04.08, 9.1.6	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
chd	CHD	
chmod	CHMOD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CH_REL_PDU	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL RELEASE n -> ms GSM 04.08, 9.1.7	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CH_REL_PDU_ERR	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL RELEASE containing additional unknown IE n -> ms GSM 04.08, 9.1.7	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
add	OCTETSTRING	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CH_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL REQUEST ms -> n GSM 04.08, 9.1.8	
Field Name	**Field Type**	**Comments**
ecau_rrf	BITSTRING [8]	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CPHM_CMD_PDU	
PCO Type:	SAP0_3	
Comments:	RR CIPHERING MODE COMMAND n -> ms GSM 04.08, 9.1.9	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
cph_res	CPH_RES	
cphms	CPHMS	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CPHM_CMD_PDU_ERR	
PCO Type:	SAP0_3	
Comments:	RR CIPHERING MODE COMMAND with additional unknown IE GSM 04.08, 9.1.9	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
cph_res	CPH_RES	
cphms	CPHMS	
add	OCTETSTRING	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CPHM_COM_PDU	
PCO Type:	SAP0_3	
Comments:	RR CIPHERING MODE COMPLETE ms -> n GSM 04.08, 9.1.10	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
mei	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CLM_CHN_PDU	
PCO Type:	SAP0_3	
Comments:	RR CLASSMARK CHANGE ms -> n GSM 04.08, 9.1.11	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
msclm	MSCLM2	
msclm_adi	MSCLM3	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CLM_ENQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR CLASSMARK ENQUIRY n -> ms GSM 04.08, 9.1.12	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	FRQRE_PDU	
PCO Type:	SAP0_3	
Comments:	RR FREQUENCY REDEFINITIONY n -> ms GSM 04.08, 9.1.13	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
chd	CHD	
ma	MA	
strt	STRT	
cchd	CCHD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HOACC_PDU	
PCO Type:	SAP0_3	
Comments:	RR HANDOVER ACCESS ms -> n GSM 04.08, 9.1.14	
Field Name	**Field Type**	**Comments**
horf	HORF	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HO_CMD_PDU	
PCO Type:	SAP0_3	
Comments:	RR HANDOVER COMMAND n -> ms GSM 04.08, 9.1.15	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
cd	CD	
ch1d_at	CHD	
horf	HORF	
pcmd	PCMD	
synchi	SYNCHI	
frqsl_at	FRQL	
frql_at	FRQL	
cchd	CCHD	
ch1mod	CHMOD	
ch2d_at	CHD	
ch2mod	CHMOD	
frqchs_at	FRQCHS	
ma_at	MA	
strt	STRT	
rtdif	TDIF	
ta	TA	
frqsl_bt	FRQL	
frql_bt	FRQL	
ch1d_bt	CHD	
ch2d_bt	CHD	
frqchs_bt	FRQCHS	
ma_bt	MA	
cphms	CPHMS	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HO_COM_PDU	
PCO Type:	SAP0_3	
Comments:	RR HANDOVER COMPLETE ms -> n GSM 04.08, 9.1.16	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
motdif	MTDIF	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HOFL_PDU	
PCO Type:	SAP0_3	
Comments:	RR HANDOVER FAILURE ms -> n GSM 04.08, 9.1.17	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	IMMASS_PDU	
PCO Type:	SAP0_3	
Comments:	RR IMMEDIATE ASSIGNMENT n -> ms GSM 04.08, 9.1.18	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
shoct	SHOCT	
pm	PM	
chd	CHD	
rqr	RQR	
ta	TA	
ma	MA	
strt	STRT	
iaroct	IARESTOCT	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	IMMASSX_PDU	
PCO Type:	SAP0_3	
Comments:	RR IMMEDIATE ASSIGNMENT n -> ms GSM 04.08, 9.1.19	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
shoct	SHOCT	
pm	PM	
chd1	CHD	
rqr1	RQR	
ta1	TA	
chd2	CHD	
rqr2	RQR	
ta2	TA	
ma	MA	
strt	STRT	
iaxroct	OCTETSTRING [1..4]	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	IMMASS_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	RR IMMEDIATE REJECT n -> ms GSM 04.08, 9.1.20	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
shoct	SHOCT	
pm	PM	
rqr1	RQR	
wi1	WI	
rqr2	RQR	
wi2	WI	
rqr3	RQR	
wi3	WI	
rqr4	RQR	
wi4	WI	
iarroct	OCTETSTRING [3]	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	MSR_RPT_PDU	
PCO Type:	SAP0_3	
Comments:	RR MEASUREMENT REPORT ms -> n GSM 04.08, 9.1.21	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
msrr	MSRR	
Detailed Comments:		

PDU Type Definition		
PDU Name:	PG1_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR PAGING REQUEST_TYPE1 n -> ms GSM 04.08, 9.1.22	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
chn_m1_2	CHNEED	
pm	PM	
mi1	MI	
mi2	MI	
p1roct	OCTETSTRING [0..17]	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	PG2_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR PAGING REQUEST TYPE2 n -> ms GSM 04.08, 9.1.23	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
chn_m1_2	CHNEED	
pm	PM	
mi1	TMSI	
mi2	TMSI	
mi3	MI	
p2roct	OCTETSTRING [1..11]	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	PG3_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR PAGING REQUEST TYPE3 n -> ms GSM 04.08, 9.1.24	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
chn_m1_2	CHNEED	
pm	PM	
mi1	TMSI	
mi2	TMSI	
mi3	TMSI	
mi4	TMSI	
p3roct	OCTETSTRING [3]	
Detailed Comments:	The message has an L2 pseudo length of 19 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	PG_RES_PDU	
PCO Type:	SAP0_3	
Comments:	RR PAGING RESPONSE ms -> n GSM 04.08, 9.1.25	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
shoct	SHOCT	
cphksn	CPHKSN	
msclm	MSCLM2	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	PART_REL_PDU	
PCO Type:	SAP0_3	
Comments:	RR PARTIAL RELEASE n -> ms GSM 04.08, 9.1.26	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
chd	MAC	
Detailed Comments:		

PDU Type Definition		
PDU Name:	PHYINFO_PDU	
PCO Type:	SAP0_3	
Comments:	RR PHYSICAL INFORMATION n -> ms GSM 04.08 clause 9.1.28	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
ta	TA	
Detailed Comments:		

PDU Type Definition		
PDU Name:	RRST_PDU	
PCO Type:	SAP0_3	
Comments:	RR STATUS n <-> ms GSM 04.08, 9.1.29	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SCHINFO_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYNCHRONIZATION CHANNEL INFORMATION n -> ms GSM 04.08, 9.1.30	
Field Name	**Field Type**	**Comments**
ncc	NCC	
bcc	BCC	
t1	T1	
t2	T2	
t3_	T3_	
Detailed Comments:	SCHINFO_PDU has a total lengh of 25 bits (GSM 04.04).	

PDU Type Definition		
PDU Name:	SYSINFO1_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE1 n -> ms GSM 04.08, 9.1.31	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
cchd	CCHD	
rachcp	RACHCP	
si1roct	OCTETSTRING [1]	
Detailed Comments:	The message has an L2 pseudo length of 21 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO2_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE2 n -> ms GSM 04.08, 9.1.32	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
bcchfl	NCD	
nccp	NCCP	
rachcp	RACHCP	
Detailed Comments:	The message has an L2 pseudo length of 22 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO2bis_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE2bis or 2ter n -> ms GSM 04.08, 9.1.33, 9.1.33a	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
xbcchfl	NCD	
rachcp	RACHCP	
si2bisroct	OCTETSTRING [1..4]	
Detailed Comments:	This PDU type is used both for SYSINFO2bis, as well as for 2ter. The 2bis message has an L2 pseudo length of 21 octets and a total length of 23 octets. The 2ter message has an L2 pseudo length of 18 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO2ter_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION 2ter n -> ms GSM 04.08, 9.1.34	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
xbcchfl	NCD2	
si2terroct	OCTETSTRING[4]	
Detailed Comments:	The 2ter message has an L2 pseudo length of 18 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO3_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE3 n -> ms GSM 04.08, 9.1.34	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
ci	CI	
lai	LAI	
ccd	CCD	
co	CO	
csp	CSP	
rachcp	RACHCP	
si3roct	SI3RO	
Detailed Comments:	The message has an L2 pseudo length of 18 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO4_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE4 n -> ms GSM 04.08, 9.1.35	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
lai	LAI	
csp	CSP	
rachcp	RACHCP	
cbchd	CHD	
cbchma	MA	
si4roct	SI4RO	
Detailed Comments:	The message has a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO5_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE5 n -> ms GSM 04.08, 9.1.36	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
bcchfl	NCD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SYSINFO5bis_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE5bis n -> ms GSM 04.08, 9.1.38	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
xbcchfl	NCD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SYSINFO5ter_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE5ter n -> ms GSM 04.08, 9.1.39	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
xbcchfl	NCD2	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SYSINFO6_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE6 n -> ms GSM 04.08, 9.1.38	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
ci	CI	
lai	LAI	
co	CO	
nccp	NCCP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SYSINFO7_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE7 n -> ms GSM 04.08, 9.1.41	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
si7roct	SI7RO	
Detailed Comments:	The message has a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO8_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE8 n -> ms GSM 04.08, 9.1.42	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
si8roct	SI7RO	
Detailed Comments:	The message has a total length of 23 octets.	

PDU Type Definition		
PDU Name:	ABRT_PDU	
PCO Type:	SAP0_3	
Comments:	MM ABORT n -> ms GSM 04.08, 9.2.8	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
rejcau	REJCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	AUTH_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	MM AUTHENTICATION REJECT n -> ms GSM 04.08, 9.2.1	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	AUTH_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM AUTHENTICATION REQUEST n -> ms GSM 04.08, 9.2.2	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
shoct	SHOCT	
cphksn	CPHKSN	
rand	RAND	
Detailed Comments:		

PDU Type Definition		
PDU Name:	AUTH_RES_PDU	
PCO Type:	SAP0_3	
Comments:	MM AUTHENTICATION RESPONSE ms -> n GSM 04.08, 9.2.3	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
sres	SRES	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CMRE_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM CM REESTABLISHMENT REQUEST ms -> n GSM 04.08, 9.2.4	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
shoct	SHOCT	
cphksn	CPHKSN	
msclm	MSCLM2	
mi	MI	
lai	LAI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CMS_ACP_PDU	
PCO Type:	SAP0_3	
Comments:	MM CM SERVICE ACCEPT n -> ms GSM 04.08, 9.2.5	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CMS_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	MM CM SERVICE REJECT n -> ms GSM 04.08, 9.2.6	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
mmcau	REJCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CMS_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM CM SERVICE REQUEST ms -> n GSM 04.08, 9.2.9	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
cphksn	CPHKSN	
svtype	CMSVTYPE	
msclm	MSCLM2	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ID_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM IDENTITY REQUEST n -> ms GSM 04.08, 9.2.10	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
shoct	SHOCT	
idtype	IDTYPE	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ID_RES_PDU	
PCO Type:	SAP0_3	
Comments:	MM IDENTITY RESPONSE ms -> n GSM 04.08, 9.2.11	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	IMSID_IN_PDU	
PCO Type:	SAP0_3	
Comments:	MM IMSI DETACH INDICATION ms -> n GSM 04.08, 9.2.12	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
msclm	MSCLM1	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	LUP_ACP_PDU	
PCO Type:	SAP0_3	
Comments:	MM LOCATION UPDATING ACCEPT n -> ms GSM 04.08, 9.2.13	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
lai	LAI	
mi	MI	
fop	IEI_8	
Detailed Comments:		

PDU Type Definition		
PDU Name:	LUP_ACP_PDU_ERR	
PCO Type:	SAP0_3	
Comments:	MM LOCATION UPDATING ACCEPT n -> ms GSM 04.08, 9.2.13	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
lai	LAI	
mi	MI	
dupmi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	LUP_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	MM LOCATION UPDATING REJECT n -> ms GSM 04.08, 9.2.14	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
rejcau	REJCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	LUP_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM LOCATION UPDATING REQUEST ms -> n GSM 04.08, 9.2.15	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
cphksn	CPHKSN	
lutype	LUT	
lai	LAI	
msclm	MSCLM1	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	MMST_PDU	
PCO Type:	SAP0_3	
Comments:	MM STATUS n <-> ms GSM 04.08, 9.2.16	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
rejcau	REJCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	TMSIRE_CMD_PDU	
PCO Type:	SAP0_3	
Comments:	MM TMSI REALLOCATION COMMAND n -> ms GSM 04.08, 9.2.17	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
lai	LAI	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	TMSIRE_COM_PDU	
PCO Type:	SAP0_3	
Comments:	MM TMSI REALLOCATION COMPLETE ms -> n GSM 04.08, 9.2.18	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ALERT_PDU	
PCO Type:	SAP0_3	
Comments:	CC ALERTING ms <-> n GSM 04.08, 9.3.1	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
fie	FIE	
pi	PI	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CALL_CO_PDU	
PCO Type:	SAP0_3	
Comments:	CC CALL CONFIRMED ms -> n GSM 04.08, 9.3.2	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcri	RPI	
bcap1	BCAP	
bcap2	BCAP	
cau	CAU	
cccap	CCCAP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CALL_PROC_PDU	
PCO Type:	SAP0_3	
Comments:	CC CALL PROCEEDING n -> ms GSM 04.08, 9.3.3	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcri	RPI	
bcap1	BCAP	
bcap2	BCAP	
fie	FIE	
pi	PI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CONN_PDU	
PCO Type:	SAP0_3	
Comments:	CC CONNECT n <-> ms GSM 04.08, 9.3.5	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
fie	FIE	
pi	PI	
cnn	CNN	
cns	CNS	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CONN_PDU_ERR	
PCO Type:	SAP0_3	
Comments:	CC CONNECT n <-> ms Used as invalid message	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
unknown	UNKWN	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CONN_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CC CONNECT ACKNOWLEDGE ms <-> n GSM 04.08, 9.3.6	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	DISC_PDU	
PCO Type:	SAP0_3	
Comments:	CC DISCONNECT ms <-> n (both directions) GSM 04.08, 9.3.7	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
fie	FIE	
pi	PI	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	DISC_PDU_ERR	
PCO Type:	SAP0_3	
Comments:		
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
unknown	UNKWN	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ESETUP_PDU	
PCO Type:	SAP0_3	
Comments:	CC EMERGENCY SETUP ms -> n GSM 04.08, 9.3.8	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcap	BCAP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	FAC_PDU	
PCO Type:	SAP0_3	
Comments:	CC FACILITY n <-> ms GSM 04.08, 9.3.9	
Field Name	**Field Type**	**Comments**
ti	TI	
cc_sspd	PD	
mt	MT	
fie	FIE	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HOLD_PDU	
PCO Type:	SAP0_3	
Comments:	CC HOLD ms -> n GSM 04.08, 9.3.10	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HOLD_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CC HOLD ACKNOWLEDGE n -> ms GSM 04.08, 9.3.11	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	MODIFY_PDU	
PCO Type:	SAP0_3	
Comments:	CC MODIFY ms <-> n GSM 04.08, 9.3.13	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcap	BCAP	
llcmp	LLCMP	
hlcmp	HLCMP	
rcsd	RCSD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	MODIFY_COM_PDU	
PCO Type:	SAP0_3	
Comments:	CC MODIFY COMPLETE ms <-> n GSM 04.08, 9.3.14	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcap	BCAP	
llcmp	LLCMP	
hlcmp	HLCMP	
rcsd	RCSD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	MODIFY_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	CC MODIFY REJECT ms <-> n GSM 04.08, 9.3.15	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcap	BCAP	
cau	CAU	
llcmp	LLCMP	
hlcmp	HLCMP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	NOTIFY_PDU	
PCO Type:	SAP0_3	
Comments:	CC NOTIFY ms <-> n GSM 04.08, 9.3.16	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
nti	NTI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	REGISTER_PDU	
PCO Type:	SAP0_3	
Comments:	SS REGISTER ms <-> n GSM 04.80, 2.4	
Field Name	**Field Type**	**Comments**
ti	TI	
sspd	PD	
mt	MT	
fie	FIE	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	PROG_PDU	
PCO Type:	SAP0_3	
Comments:	CC PROGRESS n -> ms GSM 04.08, 9.3.17	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
pi	PI	
uu	UU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	REL_PDU	
PCO Type:	SAP0_3	
Comments:	CC RELEASE n <-> ms GSM 04.08, 9.3.18	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
cau2	CAU	
fie	FIE	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	REL_COM_PDU	
PCO Type:	SAP0_3	
Comments:	CC or SS RELEASE COMPLETE n <-> ms GSM 04.08, 9.3.19; GSM 04.80, 2.5	
Field Name	**Field Type**	**Comments**
ti	TI	
cc_sspd	PD	
mt	MT	
cau	CAU	
fie	FIE	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SETUP_MO_PDU	
PCO Type:	SAP0_3	
Comments:	CC SETUP n <- ms GSM 04.08, 9.3.23.2	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcri	RPI	
bcap1	BCAP	
bcap2	BCAP	
fie	FIE	
cgps	CGPS	
cdpn	CDPN	
cdps	CDPS	
llcri	RPI	
llcmp1	LLCMP	
llcmp2	LLCMP	
hlcri	RPI	
hlcmp1	HLCMP	
hlcmp2	HLCMP	
uu	UU	
ssvi	SSVI	
clirsup	CLRSUP	
clirinv	CLRINV	
cccap	CCCAP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SETUP_MT_PDU	
PCO Type:	SAP0_3	
Comments:	CC SETUP n -> ms GSM 04.08, 9.3.23.1	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcri	RPI	
bcap1	BCAP	
bcap2	BCAP	
fie	FIE	
pi	PI	
sig	SIGNAL	
cgpn	CGPN	
cgps	CGPS	
cdpn	CDPN	
cdps	CDPS	
llcri	RPI	
llcmp1	LLCMP	
llcmp2	LLCMP	
hlcri	RPI	
hlcmp1	HLCMP	
hlcmp2	HLCMP	
uu	UU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SMSCB_PDU	
PCO Type:	SAP0_3	
Comments:	SMSCB message GSM 03.41, 9.3	
Field Name	**Field Type**	**Comments**
blocktype	BLOCKTYPE	
serial_number	SERIAL_NUMBER	
message_id	OCTETSTRING[2]	
dcs	TPDCS	
page_param	BITSTRING[8]	
message_contents	OCTETSTRING	
Detailed Comments:		

PDU Type Definition		
PDU Name:	START_DTMF_PDU	
PCO Type:	SAP0_3	
Comments:	CC START DTMF ms -> n GSM 04.08, 9.3.24	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
kpf	KPF	
Detailed Comments:		

PDU Type Definition		
PDU Name:	START_DTMF_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CC START DTMF ACKNOWLEDGE n -> ms GSM 04.08, 9.3.25	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
kpf	KPF	
Detailed Comments:		

PDU Type Definition		
PDU Name:	START_DTMF_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	CC START DTMF REJECT n -> ms GSM 04.08, 9.3.26	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CCST_PDU	
PCO Type:	SAP0_3	
Comments:	CC STATUS n <-> ms GSM 04.08, 9.3.27	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
cst	CST	
acst	ACST	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CCST_ENQ_PDU	
PCO Type:	SAP0_3	
Comments:	CC STATUS ENQUIRY ms <-> n GSM 04.08, 9.3.28	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	STOP_DTMF_PDU	
PCO Type:	SAP0_3	
Comments:	CC STOP DTMF ms -> n GSM 04.08, 9.3.29	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	STOP_DTMF_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CC STOP DTMF acknowledge n -> ms GSM 04.08, 9.3.30	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CP_DATA_PDU	
PCO Type:	SAP0_3	
Comments:	SMS CP DATA ms <-> n GSM 04.11, 7.2.1	
Field Name	**Field Type**	**Comments**
ti	TI	
smspd	PD	
mt	MT	
CPdata	CPDATA	
Detailed Comments:	CPDATA contains RPDU - RP_ACK or RP_ERROR, either returning to the MS in case of MO, or reporting the outcome of a MT messaging attempt in case of MT .	

PDU Type Definition		
PDU Name:	CPDATA_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CP DATA ACKNOWLEDGE ms <-> n GSM 04.11, 7.2.2	
Field Name	**Field Type**	**Comments**
ti	TI	
smspd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CPERR_PDU	
PCO Type:	SAP0_3	
Comments:	CP_ERROR n <-> ms GSM 04.11, 7.2.3	
Field Name	**Field Type**	**Comments**
ti	TI	
smspd	PD	
mt	MT	
cp_cause	CP_CAU	
Detailed Comments:	CP_UDAT contains RPDU, RP_ACK or RP_ERROR, either returning to the MS in case of MO, or reporting the outcome of a MT messaging attempt in case of MT .	

Constraints Part

Test suite type constraint declarations

Structured type constraint declarations

Structured Type Constraint		
Constraint Name:	Bcap_MO(rchr1:B_2; itc1:B_3; strc1:B_2; nirr:B_1; ra1 :B_2; sacp:B_3; sb, nsb1, ndb1:B_1; ur1:B_4; ir1:B_2; parity:B_3; ce1:B_2; modemt1:B_5)	
Structured Type:	BCAP	
Derivation Path:		
Comments:	BS bearer capability without OCTET 7 for direction ms -> n	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	'07'O	
extb3	'1'B	
rchr	rchr1	
cs	'0'B	
tm	'0'B	
itc	itc1	
oct_3a_etc_1	-	
oct_3a_etc_2	-	
oct_3a_etc_3	-	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	'1'B	
spb	'0'B	
strc	strc1	
dplxm	'1'B	
config	'0'B	
nirr	nirr	
est	'0'B	
extb5	'1'B	
accid	'00'B	
ra	ra1	
sacp	sacp	
extb6	'0'B	
l1id	'01'B	
uil1	'0000'B	
sb	sb	
extb6a	'0'B	
nsb	nsb1	
nb	'0'B	
ndb	ndb1	
ur	ur1	
extb6b	'0'B	
ir	ir1	
nictx	'0'B	
nicrx	'0'B	
pi	parity	
extb6c	'1'B	
ce	ce1	
modemt	modemt1	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Bcap_NoIEI(iel:LENGTH; extb3:EXTB; rchr:B_2; cs:B_1; tm:B_1; itc:B_3; extb4:EXTB; spb:SPB; strc:B_2; dplxm:B_1; config:B_1; nirr:B_1; est:B_1; extb5:EXTB; accid:B_2; ra:B_2; sacp:B_3; extb6:EXTB; l1id:B_2; uil1:B_4; sb:B_1; extb6a:EXTB; nsb:B_1; nb:B_1; ndb:B_1; ur:B_4; extb6b:EXTB; ir:B_2; nictx:B_1; nicrx:B_1; pi:B_3; extb6c:EXTB; ce:B_2; modemt:B_5)	
Structured Type:	BCAP	
Derivation Path:		
Comments:	BS bearer capability without iei for modify, modify ack and modify complete message	
Element Name	**Element Value**	**Comments**
iei	-	
iel	iel	
extb3	extb3	
rchr	rchr	
cs	cs	
tm	tm	
itc	itc	
oct_3a_etc_1	-	
oct_3a_etc_2	-	
oct_3a_etc_3	-	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	extb4	
spb	spb	
strc	strc	
dplxm	dplxm	
config	config	
nirr	nirr	
est	est	
extb5	extb5	
accid	accid	
ra	ra	
sacp	sacp	
extb6	extb6	
l1id	l1id	
uil1	uil1	
sb	sb	
extb6a	extb6a	
nsb	nsb	
nb	nb	
ndb	ndb	
ur	ur	
extb6b	extb6b	
ir	ir	
nictx	nictx	
nicrx	nicrx	
pi	pi	
extb6c	extb6c	
ce	ce	
modemt	modemt	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Bcap_MT(itc1:B_3; strc1:B_2; nirr:B_1; ra1 :B_2; sacp:B_3; sb:B_1; nbsb, nbdb:B_1; ur1:B_4; ir1:B_2; parity:B_3; ce1:B_2; modemt1:B_5)	
Structured Type:	BCAP	
Derivation Path:		
Comments:	BS bearer capability for direction n->ms	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	'07'O	
extb3	'1'B	
rchr	'01'B	
cs	'0'B	
tm	'0'B	
itc	itc1	
oct_3a_etc_1	-	
oct_3a_etc_2	-	
oct_3a_etc_3	-	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	'1'B	
spb	'0'B	
strc	strc1	
dplxm	'1'B	
config	'0'B	
nirr	nirr	
est	'0'B	
extb5	'1'B	
accid	'00'B	
ra	ra1	
sacp	sacp	
extb6	'0'B	
l1id	'01'B	
uil1	'0000'B	
sb	sb	
extb6a	'0'B	
nsb	nbsb	
nb	'0'B	
ndb	nbdb	
ur	ur1	
extb6b	'0'B	
ir	ir1	
nictx	'0'B	
nicrx	'0'B	
pi	parity	
extb6c	'1'B	
ce	ce1	
modemt	modemt1	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcapX_MO(rchr1:B_2; itc1:B_3; strc1:B_2; nirr:B_1; ra1 :B_2; sacp:B_3; sb, nsb1, ndb1:B_1; ur1:B_4; ir1:B_2; parity:B_3; ce1:B_2; modemt1:B_5; fctl:B_5)	
Structured Type:	BCAP	
Derivation Path:		
Comments:	BS bearer capability with OCTET 7 for direction ms -> n	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	('07'O, '08'O)	
extb3	'1'B	
rchr	rchr1	
cs	'0'B	
tm	'0'B	
itc	itc1	
oct_3a_etc_1	-	
oct_3a_etc_2	-	
oct_3a_etc_3	-	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	'1'B	
spb	'0'B	
strc	strc1	
dplxm	'1'B	
config	'0'B	
nirr	nirr	
est	'0'B	
extb5	'1'B	
accid	'00'B	
ra	ra1	
sacp	sacp	
extb6	'0'B	
l1id	'01'B	
uil1	'0000'B	
sb	sb	
extb6a	'0'B	
nsb	nsb1	
nb	'0'B	
ndb	ndb1	
ur	ur1	
extb6b	'0'B	
ir	ir1	
nictx	'0'B	
nicrx	'0'B	
pi	parity	
extb6c	'1'B	
ce	ce1	
modemt	modemt1	
extb7	'1'B IF_PRESENT	
l2id	'10'B IF_PRESENT	
uil2	fctl IF_PRESENT	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcapX_MT(itc1:B_3; strc1:B_2; nirr:B_1; ra1 :B_2; sacp:B_3; sb:B_1; nbsb, nbdb:B_1; ur1:B_4; ir1:B_2; parity:B_3; ce1:B_2; modemt1:B_5; flct:B_5)	
Structured Type:	BCAP	
Derivation Path:		
Comments:	BS bearer capability for direction n->ms BS 21, .. 26, except BS23	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	'08'O	
extb3	'1'B	
rchr	'01'B	
cs	'0'B	
tm	'0'B	
itc	itc1	
oct_3a_etc_1	-	
oct_3a_etc_2	-	
oct_3a_etc_3	-	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	'1'B	
spb	'0'B	
strc	strc1	
dplxm	'1'B	
config	'0'B	
nirr	nirr	
est	'0'B	
extb5	'1'B	
accid	'00'B	
ra	ra1	
sacp	sacp	
extb6	'0'B	
l1id	'01'B	
uil1	'0000'B	
sb	sb	
extb6a	'0'B	
nsb	nbsb	
nb	'0'B	
ndb	nbdb	
ur	ur1	
extb6b	'0'B	
ir	ir1	
nictx	'0'B	
nicrx	'0'B	
pi	parity	
extb6c	'1'B	
ce	ce1	
modemt	modemt1	
extb7	'1'B	
l2id	'10'B	
uil2	flct	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Bcap_02	
Structured Type:	BCAP	
Derivation Path:	Bcap_Speech_MT.	
Comments:	invalid information element, length = 1, arbitrary contents.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'01'O	
extb3	'1'B	
rchr	'11'B	
cs	'0'B	
tm	'1'B	
itc	'110'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Bcap_Speech_Efr_MO(rcr:B_2)	
Structured Type:	BCAP	
Derivation Path:		
Comments:	EFR speech bearer capability for direction ms->n	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	('03'O, '04'O, '05'O, '06'O, '07'O)	
extb3	'0'B	
rchr	rcr	
cs	'0'B	
tm	'0'B	
itc	'000'B	
oct_3a_etc_1	?	
oct_3a_etc_2	?	
oct_3a_etc_3	*	
oct_3a_etc_4	*	
oct_3a_etc_5	*	
oct_3a_etc_6	*	
extb4	-	
spb	-	
strc	-	
dplxm	-	
config	-	
nirr	-	
est	-	
extb5	-	
accid	-	
ra	-	
sacp	-	
extb6	-	
l1id	-	
uil1	-	
sb	-	
extb6a	-	
nsb	-	
nb	-	
ndb	-	
ur	-	
extb6b	-	
ir	-	
nictx	-	
nicrx	-	
pi	-	
extb6c	-	
ce	-	
modemt	-	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Bcap_Speech_MO(rcr:B_2)	
Structured Type:	BCAP	
Derivation Path:		
Comments:	speech bearer capability for direction ms->n.	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	'01'O	
extb3	'1'B	
rchr	rcr	
cs	'0'B	
tm	'0'B	
itc	'000'B	
oct_3a_etc_1	-	
oct_3a_etc_2	-	
oct_3a_etc_3	-	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	-	
spb	-	
strc	-	
dplxm	-	
config	-	
nirr	-	
est	-	
extb5	-	
accid	-	
ra	-	
sacp	-	
extb6	-	
l1id	-	
uil1	-	
sb	-	
extb6a	-	
nsb	-	
nb	-	
ndb	-	
ur	-	
extb6b	-	
ir	-	
nictx	-	
nicrx	-	
pi	-	
extb6c	-	
ce	-	
modemt	-	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Bcap_Speech_MT	
Structured Type:	BCAP	
Derivation Path:		
Comments:	speech bearer capability for direction n->ms	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	'01'O	
extb3	'1'B	
rchr	'01'B	
cs	'0'B	
tm	'0'B	
itc	'000'B	
oct_3a_etc_1	-	
oct_3a_etc_2	-	
oct_3a_etc_3	-	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	-	
spb	-	
strc	-	
dplxm	-	
config	-	
nirr	-	
est	-	
extb5	-	
accid	-	
ra	-	
sacp	-	
extb6	-	
l1id	-	
uil1	-	
sb	-	
extb6a	-	
nsb	-	
nb	-	
ndb	-	
ur	-	
extb6b	-	
ir	-	
nictx	-	
nicrx	-	
pi	-	
extb6c	-	
ce	-	
modemt	-	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Blocktype_01(seqnum: BITSTRING; lb: BITSTRING)	
Structured Type:	BLOCKTYPE	
Derivation Path:		
Comments:	Block type, GSM 04.12, 3.3.1	
Element Name	**Element Value**	**Comments**
spare1	'0'B	
lpd	'01'B	
lb	lb	
sequence_number	seqnum	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_Omit	
Structured Type:	NCD	
Derivation Path:		
Comments:	Omit Bcch Frequency list (Neighbour cells description)	
Element Name	**Element Value**	**Comments**
rfl2	-	
extind	-	
baind	-	
rfl4	-	
rfl	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_01	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under GSM900 with the ARFCN list = {10, 20, 40, 80, 90, 100, 110, 120}.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'0'B	
baind	'0'B	
rfl4	'0000'B	
rfl	'80200802008000000008000080200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_02	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Alternative neighbour cells description for SYSTEM INFORMATION 2 and 5 under GSM900. These are ARFCNs 15, 85, 95, 105, 115, and 122	
Element Name	**Element Value**	**Comments**
rfl4	'0010'B	
rfl	'04010040100000000000000004000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_03	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 for DCS1800 for cell B with the ARFCN list = {520, 600, 700, 780, 810, 870}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'04412C168E44000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_04	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Alternative neighbour cells description for SYSTEM INFORMATION 2 and 5 for DCS1800.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'09412C168E44000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_05	
Structured Type:	NCD	
Derivation Path:		
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 1 for idle mode testing of GSM900.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'0'B	
baind	'0'B	
rfl4	'1000'B	
rfl	'000001001000030000004000000040'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_06	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_05.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 2 for idle mode testing of GSM900.	
Element Name	**Element Value**	**Comments**
rfl4	'0100'B	
rfl	'0000020020000C0000008000000080'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_07	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_05.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 3 for idle mode testing of GSM900.	
Element Name	**Element Value**	**Comments**
rfl4	'0010'B	
rfl	'000004004000300000010000000100'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_08	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_05.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 4 for idle mode testing of GSM900.	
Element Name	**Element Value**	**Comments**
rfl4	'0001'B	
rfl	'000008008000C0000002000000200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_09	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_05.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 5 for idle mode testing of GSM900.	
Element Name	**Element Value**	**Comments**
rfl4	'0000'B	
rfl	'800010010003000000040000000400'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_10	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_05.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 6 for idle mode testing of GSM900.	
Element Name	**Element Value**	**Comments**
rfl4	'0000'B	
rfl	'40002002000C000000080000000800'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_11	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_05.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 7 for idle mode testing of GSM900.	
Element Name	**Element Value**	**Comments**
rfl4	'0000'B	
rfl	'200040040030000000100000001000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_12	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_05.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 8 for idle mode testing of GSM900.	
Element Name	**Element Value**	**Comments**
rfl4	'1000'B	
rfl	'00000000000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_13	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 1 for idle mode testing of DCS1800.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
rfl4	'1001'B	
rfl	'042DE8EDB149B80000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_14	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 2 for idle mode testing of DCS1800.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
rfl4	'1001'B	
rfl	'04AE28ECF0CBB80000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_15	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 3 for idle mode testing of DCS1800.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
rfl4	'1001'B	
rfl	'052E68EC304DB80000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_16	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 4 for idle mode testing of DCS1800.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
rfl4	'1001'B	
rfl	'05AEA8EB6FCFB80000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_17	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 5 for idle mode testing of DCS1800.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
rfl4	'1001'B	
rfl	'062F6869EF53B80000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_18	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 6 for idle mode testing of DCS1800.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
rfl4	'1001'B	
rfl	'06AF28E9EED3B80000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_19	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	cells description for SYSTEM INFORMATION 2 and 5 of cell 7 for idle mode testing of DCS1800.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
rfl4	'1001'B	
rfl	'072F5FE900D548000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_20	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for measurement testing. Empty BA list, format is bitmap 0, IE carries complete BA.	
Element Name	**Element Value**	**Comments**
baind	'1'B	
rfl	'00000000000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_21	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for measurement testing. BA list = {2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 28, 29, 30, 32, 34, 35, 36, 38, 40, 44}, format is bitmap 0, IE carries complete BA.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'0'B	
baind	'1'B	
rfl4	'0000'B	
rfl	'00000000000000000008AEBAFFBFFA'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_22	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for measurement testing. BA list = {2, 14, 20, 38, 44}, format is bitmap 0, IE carries complete BA.	
Element Name	**Element Value**	**Comments**
baind	'1'B	
rfl	'00000000000000000082000082002'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_24	
Structured Type:	NCD	
Derivation Path:		
Comments:	neighbour cells description with out channel for SYSTEM INFORMATION 5 for GSM. Empty BA-list. Format is bitmap 0, IE carries only a part of the BA.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'1'B	
baind	'1'B	
rfl4	'0000'B	
rfl	'00000000000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_25	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description without channel for SYSTEM INFORMATION 5 for DCS1800. Empty BA-list. Format is bitmap 0, IE carries only a part of the BA.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'0000'B	
rfl	'00000000000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_26	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5bis for GSM900. ARFCN 500 belongs to the BA list, format is 1024 range, IE carries only a part of the BA	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'0001'B	
rfl	'F40000000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_27	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for DCS1800. BA list = {514, 530, 549, 602, 665, 686, 762, 810}, format is 1024 range, IE carries complete BA.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'1'B	
rfl4	'0010'B	
rfl	'99C6187B6D0D4C3800000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_28	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for measurement testing.BA list = {2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 28, 29, 30, 32, 34, 35, 36, 38, 40, 44}, format is bitmap 0, IE carries only a part of the BA.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'1'B	
baind	'1'B	
rfl4	'0000'B	
rfl	'00000000000000000008AEBAFFBFFA'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_29	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for DCS1800 measurement testing. BA list = {549, 602, 665, 686, 810}. Format is range 1024, IE carries only a part of the BA.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'0010'B	
rfl	'99E0A472E1000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_30	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5bis for measurement testing. BA list {0, 800}, format is range 1024, IE carries only a part of the BA.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'0111'B	
rfl	'20000000000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_31	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for DCS1800 measurement testing. BA list {20, 514, 530, 549, 762}, format is range 1024, IE carries only a part of the BA.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'0000'B	
rfl	'1473FF8AFC0000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_32	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for DCS1800 measurement testing. BA list {514, 549, 602, 665, 810 }, format is range 1024, IE carries complete BA.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'1'B	
rfl4	'0000'B	
rfl	'5AE5B4375BC0000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_33	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for GSM900. BA list = {2, 14, 20, 38, 44}, format is bitmap 0, IE carries only a part of the BA.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'1'B	
baind	'1'B	
rfl4	'0000'B	
rfl	'00000000000000000000082000082002'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_34	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for DCS1800 measurement testing. BA list = {549, 602, 810 }, format is range 1024, IE carries only a part of the BA.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'0010'B	
rfl	'5AE5B4000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_34d	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for DCS1800 measurement testing. BA list = {514, 665 }, format is range 1024, IE carries only a part of the BA.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'0010'B	
rfl	'99B4800000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_35	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for measurement testing.	
Element Name	**Element Value**	**Comments**
baind	'1'B	
rfl	'00000000000000000000082082082082'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_36	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for measurement testing.	
Element Name	**Element Value**	**Comments**
extind	'1'B	
baind	'1'B	
rfl	'00000000000000000000082082082002'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_37	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for measurement testing.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'1'B	
baind	'1'B	
rfl4	'0000'B	
rfl	'00000000000000000000000000000A2'O	
Detailed Comments:	only used in TC_26_6_3_5	

Structured Type Constraint		
Constraint Name:	BcchFreqLst_38	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for DCS1800.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'0110'B	
rfl	'CBA3BEB89A9048C000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_39	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for DCS1800.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'1111'B	
rfl	'E2800000001000001450000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_40	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for DCS1800.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'1101'B	
rfl	'5B945800000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_41	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 BIS for GSM900.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'1001'B	
rfl	'097BBA32AE888C0000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_42	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for GSM900.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'1000'B	
rfl	'FA0787AE4B88000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_43	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for GSM900.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'1111'B	
rfl	'E2800000001000001440000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_44	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for DCS1800.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'1011'B	
rfl	'97E8E80CEF8000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_45	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description of cell B for SYSTEM INFORMATION 2 and 5 under GSM900.	
Element Name	**Element Value**	**Comments**
rfl	'802008020080000000000000000200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_46	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2bis and 5bis in cell A under EGSMwith the ARFCN list = {988, 990, 1003}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'0'B	
rfl4	'1011'B	
rfl	'EE07F300000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_47	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2bis and 5bis in cell B under EGSMwith the ARFCN list = {1005, 1010, 1015}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'0'B	
rfl4	'1101'B	
rfl	'F68AEC000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_48	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 for DCS1800 for cell B in RR, MM and SMS test cases with the ARFCN list = {520, 590, 600, 700, 780, 810, 870}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'04411307BB00C80000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_50	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_25.	
Comments:	Neighbour cells description without channel for SYSTEM INFORMATION 5 for DCS1800. Empty BA-list. Format is bitmap 0, IE carries the complete BA.	
Element Name	**Element Value**	**Comments**
extind	'0'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_51	
Structured Type:	NCD	
Derivation Path:		
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 for DCS1800 measurement testing. BA list { 514, 530, 762}, format is range 1024, IE carries only a part of the BA.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'1'B	
rfl4	'0010'B	
rfl	'12F83A00000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_52	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_33.	
Comments:	Neighbour cells description for SYSTEM INFORMATION 5 of cell B for EGSM. BA list = {10, 36, 40, 114, 118}, format is bitmap 0, IE carries only a part of the BA.	
Element Name	**Element Value**	**Comments**
baind	'0'B	
rfl4	'0000'B	
rfl	'22000000000000000000008800000200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_e201	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description any cell for SYSTEM INFORMATION 2 and 5 under EGSM.	
Element Name	**Element Value**	**Comments**
rfl	'038C0D0C000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_e202	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description any cell for SYSTEM INFORMATION 2 and 5 under EGSM.	
Element Name	**Element Value**	**Comments**
rfl	'058DEF07FE00000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_e203	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description any cell for SYSTEM INFORMATION 2 and 5 under EGSM.	
Element Name	**Element Value**	**Comments**
rfl	'078904797F83F98000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_e204	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description any cell for SYSTEM INFORMATION 2 and 5 under EGSM.	
Element Name	**Element Value**	**Comments**
rfl	'0987EDFF0E3CFD0000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_e205	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description any cell for SYSTEM INFORMATION 2 and 5 under EGSM.	
Element Name	**Element Value**	**Comments**
rfl	'078E00000200200000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_e206	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description any cell for SYSTEM INFORMATION 2 and 5 under EGSM.	
Element Name	**Element Value**	**Comments**
rfl	'10000000000000000000000000000200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_e207	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description any cell for SYSTEM INFORMATION 2 and 5 under EGSM.	
Element Name	**Element Value**	**Comments**
rfl	'078904797F83F98000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_e208	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description any cell for SYSTEM INFORMATION 2 and 5 under EGSM.	
Element Name	**Element Value**	**Comments**
rfl	'058C0026D00000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_e209	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description any cell for SYSTEM INFORMATION 2 and 5 under EGSM.	
Element Name	**Element Value**	**Comments**
rfl	'058A00137400000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_e210	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description any cell for SYSTEM INFORMATION 2 and 5 under EGSM.	
Element Name	**Element Value**	**Comments**
rfl	'0583E810050000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_e211	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description any cell for SYSTEM INFORMATION 2 and 5 under EGSM.	
Element Name	**Element Value**	**Comments**
rfl	'058FBA020014000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_e212	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description any cell for SYSTEM INFORMATION 2 and 5 under EGSM.	
Element Name	**Element Value**	**Comments**
rfl	'06890478FCC000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ca1_g01	
Structured Type:	CCHD	
Derivation Path:		
Comments:	ARFCNs = 1.. 64 for GSM of TC_26_6_6_1, ca_pgsm(1) = 64.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'00000000000000000FFFFFFFFFFFFFFF'O	
Detailed Comments:	B_ARFCN = 1, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Ca2_g01	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for GSM of TC_26_6_6_1.	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'0F00F000003C003C0000FE003C008081'O	
Detailed Comments:	ca_pgsm(2) = 30, B_ARFCN = 41, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_g02	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'0101F1F1010101012101011111111111'O	
Detailed Comments:	ca_pgsm(2) = 30, B_ARFCN = 45, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_g03	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00501010101010101010101010FFFF'O	
Detailed Comments:	ca_pgsm(2) = 30, B_ARFCN = 124, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_g04	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00909090909090909090909090909090'O	
Detailed Comments:	ca_pgsm(2) = 30, B_ARFCN = 5, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_g05	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'0909090909090909090909090909090900'O	
Detailed Comments:	ca_pgsm(2) = 30, B_ARFCN = 12, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_g06	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'0040C0C0C0E0C0C0C0C0C0C0C0C0C012'O	
Detailed Comments:	ca_pgsm(2) = 30, B_ARFCN = 120, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_g07	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00007F0000FF0000FF0000000002003F'O	
Detailed Comments:	ca_pgsm(2) = 30, B_ARFCN = 112, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_g08	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00020202021A1A1A1A1A1A3A02020202'O	
Detailed Comments:	ca_pgsm(2) = 30, B_ARFCN = 122, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_g09	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'0000000000000F0F0F0F000030F0F0F0'O	
Detailed Comments:	ca_pgsm(2) = 30, B_ARFCN = 30, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca3_g01	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00000000000000000000000000E0F0F1' O	
Detailed Comments:	ca_pgsm(3) = 12, B_ARFCN = 21, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_g02	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00010001010001012001010001018080' O	
Detailed Comments:	ca_pgsm(3) = 12, B_ARFCN = 121, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_g03	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00C0008000C000C000C000C000000010 'O	
Detailed Comments:	ca_pgsm(3) = 12, B_ARFCN = 103, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_g04	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00000000080018180000180018001880' O	
Detailed Comments:	ca_pgsm(3) = 12, B_ARFCN = 93, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_g05	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00000000002000240024002401240024' O	
Detailed Comments:	ca_pgsm(3) = 12, B_ARFCN = 83, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_g06	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00300300003000000300300003000000'O	
Detailed Comments:	ca_pgsm(3) = 12, B_ARFCN = 45, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_g07	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00101000100000080800080810128080'O	
Detailed Comments:	ca_pgsm(3) = 12, B_ARFCN = 85, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_g08	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'0F000000000000380000002000000F0'O	
Detailed Comments:	ca_pgsm(3) = 12, B_ARFCN = 67, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_g09	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00818100000000404600000000008181'O	
Detailed Comments:	ca_pgsm(3) = 12, B_ARFCN = 66, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca2_d01	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for DCS of TC_26_6_6_1.	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E0000000000000000001FFFFFFFFF'O	
Detailed Comments:	ca_dcs(2) = 30, Org_ARFCN = 700, B_ARFCN = 804, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_d02	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E7FFF00FF00F0000E000000000000'O	
Detailed Comments:	ca_dcs(2) = 30, Org_ARFCN = 700, B_ARFCN = 705, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_d03	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E3C00FF00FF00FF00180000000000'O	
Detailed Comments:	ca_dcs(2) = 30, Org_ARFCN = 700, B_ARFCN = 703, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_d04	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E000000000F0000F000FF54555554'O	
Detailed Comments:	ca_dcs(2) = 30, Org_ARFCN = 700, B_ARFCN = 759, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_d05	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E2828282B0000FFFF000010101010'O	
Detailed Comments:	ca_dcs(2) = 30, Org_ARFCN = 700, B_ARFCN = 730, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_d06	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E00FF00202222222222222202007E'O	
Detailed Comments:	ca_dcs(2) = 30, Org_ARFCN = 700, B_ARFCN = 794, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_d07	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E032B2B2B200000000000000000000'O	
Detailed Comments:	ca_dcs(2) = 30, Org_ARFCN = 700, B_ARFCN = 734, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_d08	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E405050501404F404000F00000000'O	
Detailed Comments:	ca_dcs(2) = 30, Org_ARFCN = 700, B_ARFCN = 779, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca2_d09	
Structured Type:	CCHD	
Derivation Path:		
Comments:	30 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E00F000F00078000C808FA0A0202F'O	
Detailed Comments:	ca_dcs(2) = 30, Org_ARFCN = 700, B_ARFCN = 744, not in MA(2)	

Structured Type Constraint		
Constraint Name:	Ca3_d01	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E000000000000000000000F0F0F0'O	
Detailed Comments:	ca_dcs(3) = 12, Org_ARFCN = 700, B_ARFCN = 804, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_d02	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E7F00F800000000000000000000000'O	
Detailed Comments:	ca_dcs(3) = 12, Org_ARFCN = 700, B_ARFCN = 705, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_d03	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E5000600C0030000C003000000000' O	
Detailed Comments:	ca_dcs(3) = 12, Org_ARFCN = 700, B_ARFCN = 703, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_d04	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E00000000000004F0000640300C00' O	
Detailed Comments:	ca_dcs(3) = 12, Org_ARFCN = 700, B_ARFCN = 759, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_d05	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E00601033000002404000000040400' O	
Detailed Comments:	ca_dcs(3) = 12, Org_ARFCN = 700, B_ARFCN = 730, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_d06	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E000000200020000080802025858' O	
Detailed Comments:	ca_dcs(3) = 12, Org_ARFCN = 700, B_ARFCN = 794, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_d07	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E000808006000004848080808080 0' O	
Detailed Comments:	ca_dcs(3) = 12, Org_ARFCN = 700, B_ARFCN = 734, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_d08	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E00F0000000000000000F40404F40'O	
Detailed Comments:	ca_dcs(3) = 12, Org_ARFCN = 700, B_ARFCN = 779, not in MA(3)	

Structured Type Constraint		
Constraint Name:	Ca3_d09	
Structured Type:	CCHD	
Derivation Path:		
Comments:	12 ARFCNs for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F5E00200304041800002020200600 00'O	
Detailed Comments:	ca_dcs(3) = 12, Org_ARFCN = 700, B_ARFCN = 744, not in MA(3)	

Structured Type Constraint		
Constraint Name:	CallCntrlCap	
Structured Type:	CCCAP	
Derivation Path:		
Comments:	CC state 'st'	
Element Name	**Element Value**	**Comments**
iei	'00010101'B	
iel	'01'O	
spr	'0000000'B	
dtmf	TSPX_DTMF	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CallState_01(st:INTEGER)	
Structured Type:	CST	
Derivation Path:		
Comments:	CC state 'st'	
Element Name	**Element Value**	**Comments**
cs	'11'B	
csv	INT_TO_BIT(st, 6)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_Def	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with any valid cause value, any valid location.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	('0000'B, '0001'B, '0010'B, '0011'B, '0100'B, '0101'B, '0111'B, '1010'B)	
extb3a	-	
rec	-	
extb4	?	
cau_class	?	
cau_va	?	
cau_di	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_01	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #16, location = user.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'0000'B	
extb3a	-	
rec	-	
extb4	'1'B	
cau_class	'001'B	
cau_va	'0000'B	
cau_di	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_01iei	
Structured Type:	CAU	
Derivation Path:	Cause_01.	
Comments:	optional (IEI present) cause information element with cause value = #16, location = user.	
Element Name	**Element Value**	**Comments**
iei	'00001000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_02	
Structured Type:	CAU	
Derivation Path:	Cause_01.	
Comments:	cause value #97	
Element Name	**Element Value**	**Comments**
iel	?	
location	?	
extb4	?	
cau_class	'110'B	
cau_va	'0001'B	
cau_di	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_03	
Structured Type:	CAU	
Derivation Path:	Cause_01.Cause_02.	
Comments:	cause #98 -- message type not compatible with protocol state	
Element Name	**Element Value**	**Comments**
cau_va	'0010'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_04	
Structured Type:	CAU	
Derivation Path:	Cause_01.Cause_02.	
Comments:	The cause value is #96 -- invalid mandatory information	
Element Name	**Element Value**	**Comments**
cau_va	'0000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_04iei	
Structured Type:	CAU	
Derivation Path:		
Comments:	The cause value is #96 -- invalid mandatory information	
Element Name	**Element Value**	**Comments**
iei	'00001000'B	
iel	?	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	?	
extb3a	-	
rec	-	
extb4	?	
cau_class	'110'B	
cau_va	'0000'B	
cau_di	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_06	
Structured Type:	CAU	
Derivation Path:	Cause_01.	
Comments:	Cause information element containing arbitrary spare bits	
Element Name	**Element Value**	**Comments**
spb	'1'B	
cau_class	'000'B	
cau_va	'0001'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_07	
Structured Type:	CAU	
Derivation Path:	Cause_01.	
Comments:	cause information element with cause value = #81.	
Element Name	**Element Value**	**Comments**
iei	'00001000'B	
iel	?	
location	?	
extb4	?	
cau_class	'101'B	
cau_va	'0001'B	
cau_di	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_08	
Structured Type:	CAU	
Derivation Path:		
Comments:	used as an unknown IE	
Element Name	**Element Value**	**Comments**
iei	'00100100'B	
iel	'01'O	
extb3	'1'B	
cs	'11'B	
spb	'1'B	
location	'1111'B	
extb3a	-	
rec	-	
extb4	-	
cau_class	-	
cau_va	-	
cau_di	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_09	
Structured Type:	CAU	
Derivation Path:	Cause_08.	
Comments:	used as an unknown IE	
Element Name	**Element Value**	**Comments**
iei	'01111101'B	
iel	'01'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'1110'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_10	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #1, location = user.	
Element Name	**Element Value**	**Comments**
iei	'00001000'B	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'0000'B	
extb3a	-	
rec	-	
extb4	'1'B	
cau_class	'000'B	
cau_va	'0001'B	
cau_di	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_11	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #31.	
Element Name	**Element Value**	**Comments**
iei	'00001000'B	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'0000'B	
extb3a	-	
rec	-	
extb4	'1'B	
cau_class	'001'B	
cau_va	'1111'B	
cau_di	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_12	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #88.	
Element Name	**Element Value**	**Comments**
iei	'00001000'B	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'????'B	
extb3a	-	
rec	-	
extb4	'1'B	
cau_class	'101'B	
cau_va	'1000'B	
cau_di	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_13	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #21.	
Element Name	**Element Value**	**Comments**
iei	'00001000'B	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'????'B	
extb3a	-	
rec	-	
extb4	'1'B	
cau_class	'001'B	
cau_va	'0101'B	
cau_di	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_14	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #102.	
Element Name	**Element Value**	**Comments**
iei	'00001000'B	
iel	?	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'????'B	
extb3a	-	
rec	-	
extb4	?	
cau_class	'110'B	
cau_va	'0110'B	
cau_di	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_15	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #47, n -> ms.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'0010'B	
extb3a	-	
rec	-	
extb4	'1'B	
cau_class	'010'B	
cau_va	'1111'B	
cau_di	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_16	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #58 bearer capability not presently available, n -> ms.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'0010'B	
extb3a	-	
rec	-	
extb4	'1'B	
cau_class	'011'B	
cau_va	'1010'B	
cau_di	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_17	
Structured Type:	CAU	
Derivation Path:		
Comments:	optional (IEI present) cause information element with cause value = #17 user busy. ms -> n.	
Element Name	**Element Value**	**Comments**
iei	'00001000'B	
iel	?	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'????'B	
extb3a	-	
rec	-	
extb4	?	
cau_class	'001'B	
cau_va	'0001'B	
cau_di	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_18	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #30 response to status enquiry. ms -> n.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'????'B	
extb3a	-	
rec	-	
extb4	?	
cau_class	'001'B	
cau_va	'1110'B	
cau_di	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_22	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #81.	
Element Name	**Element Value**	**Comments**
iei	'00001000'B	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'????'B	
extb3a	-	
rec	-	
extb4	'1'B	
cau_class	'101'B	
cau_va	'0001'B	
cau_di	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_23	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #102.	
Element Name	**Element Value**	**Comments**
iei	'00001000'B	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'0000'B	
extb3a	-	
rec	-	
extb4	'1'B	
cau_class	'110'B	
cau_va	'0110'B	
cau_di	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_26	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #16, location = user. n->ms	
Element Name	**Element Value**	**Comments**
iei	'00001000'B	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'0000'B	
extb3a	-	
rec	-	
extb4	'1'B	
cau_class	'001'B	
cau_va	'0000'B	
cau_di	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_27	
Structured Type:	CAU	
Derivation Path:	Cause_26.	
Comments:	cause value #88, ms->n.	
Element Name	**Element Value**	**Comments**
iel	?	
location	?	
extb4	?	
cau_class	'101'B	
cau_va	'1000'B	
cau_di	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_28	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #29 'facility rejected', n -> ms.	
Element Name	**Element Value**	**Comments**
iei	'00001000'B	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'0010'B	
extb3a	-	
rec	-	
extb4	'1'B	
cau_class	'001'B	
cau_va	'1101'B	
cau_di	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cdpn_01	
Structured Type:	CDPN	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'01011110'B	
iel	'02'O	
tonnpi	TonNpi_02	
digits	'01'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cdps_01	
Structured Type:	CDPS	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'01101101'B	
iel	'03'O	
subad	SubAdd_01	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_Omit	
Structured Type:	CCHD	
Derivation Path:		
Comments:	Omit descriptor	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_01	
Structured Type:	CCHD	
Derivation Path:		
Comments:	ARFCN = 124	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'380000000000000000000000000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_01d	
Structured Type:	CCHD	
Derivation Path:		
Comments:	ARFCN = 801	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'BF908000000000000000000000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_02	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in RR testing of GSM 900 with the ARFCN_list={20,30,50,70}.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'00000000000000200002000020080000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_03	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in RR testing of DCS1800 with the ARFCN list = {590, 650, 750, 850}..	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'892728271900000000000000000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_04	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell B in RR testing of GSM 900.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'00000000000000000000000000000200' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_05	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N1 in measurement testing of GSM 900.	
Element Name	**Element Value**	**Comments**
rfl	'00000000000000000000000000000080' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_05d	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N1 in measurement testing of DCS1800.	
Element Name	**Element Value**	**Comments**
rfl	'82120000000000000000000000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_06	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N2 in measurement testing of GSM 900.	
Element Name	**Element Value**	**Comments**
rfl	'00000000000000000000000000002000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_06d	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N2 in measurement testing of DCS.	
Element Name	**Element Value**	**Comments**
rfl	'825A0000000000000000000000002000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_07	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N3 in measurement testing of GSM 900.	
Element Name	**Element Value**	**Comments**
rfl	'00000000000000000000000000080000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_07d	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N3 in measurement testing of DCS1800.	
Element Name	**Element Value**	**Comments**
rfl	'829900000000000000000000000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_08	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N4 in measurement testing of GSM 900.	
Element Name	**Element Value**	**Comments**
rfl	'00000000000000000000000002000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_08d	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N4 in measurement testing of DCS1800.	
Element Name	**Element Value**	**Comments**
rfl	'82FA0000000000000000000000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_09	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N5 in measurement testing of GSM 900.	
Element Name	**Element Value**	**Comments**
rfl	'00000000000000000000000080000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_09d	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N5 in measurement testing of DCS1800.	
Element Name	**Element Value**	**Comments**
rfl	'82AE0000000000000000000080000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_10	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N6 in measurement testing of GSM 900.	
Element Name	**Element Value**	**Comments**
rfl	'00000000000000000000002000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_10d	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N6 in measurement testing of DCS1800.	
Element Name	**Element Value**	**Comments**
rfl	'82250000000000000000000000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_11	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N7 in measurement testing of GSM 900.	
Element Name	**Element Value**	**Comments**
rfl	'00000000000000000000080000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_11d	
Structured Type:	CCHD	
Derivation Path:	CellChDes_04.	
Comments:	for cell N7 in measurement testing of DCS1800.	
Element Name	**Element Value**	**Comments**
rfl	'832A0000000000000000000000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_12	
Structured Type:	CCHD	
Derivation Path:		
Comments:	bit map o format	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00000000000000000002000020000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_12d	
Structured Type:	CCHD	
Derivation Path:		
Comments:	variable bit map format	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F450000000000000000000000001000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_12man	
Structured Type:	CCHD	
Derivation Path:		
Comments:	bit map o format	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'00000000000000000002000020000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_12dman	
Structured Type:	CCHD	
Derivation Path:		
Comments:	variable bit map format	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'8F450000000000000000000000001000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_15	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in RR testing of GSM 900 with ARFCN_list={10,17,20,26,34,42,45,46,52,59}	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'00000000000000000408320202090200' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_15d	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in RR testing of DCS with the ARFCN_list={734,741,747,754,759,766,773,775,779,782}	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'8D6F20D0DCF0535E0000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_19	
Structured Type:	CCHD	
Derivation Path:		
Comments:	Frequency list for EGSM test case TC_26_10_2_2: 0, 30, 40, 66, 80, 1005, 1010, 1015.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'841EEA893EF981438000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_20_A	
Structured Type:	CCHD	
Derivation Path:	CellChDes_01.	
Comments:	Cell Channel Description for cell A in RR testing of GSM 900 HO cases with the ARFCN_list={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
rfl	'0002080000000F020408320202090200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_20_Aman	
Structured Type:	CCHD	
Derivation Path:		
Comments:	Cell Channel Description for any cell in RR testing of GSM 900 HO cases with the ARFCN_list={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'0002080000000F020408320202090200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_201_Ad	
Structured Type:	CCHD	
Derivation Path:		
Comments:	Cell Channel Description for cell A in RR testing of DCS1800 HO cases with the ARFCN_list={734,741,747,754,759,762,766,767,773,775,779,782,791,798,829,832,844} using 256 format.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'8B6F14F32FC602C59EFFA54999400000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_202_Ad	
Structured Type:	CCHD	
Derivation Path:	CellChDes_01.	
Comments:	Cell Channel Description for cell A in RR testing of DCS1800 HO cases with the Complete Cell Allocation of Cell A in 512 format. ARFCN_list={734,741,747,754,759,762,766,767,773,775,779,782,791,798,829,832,844} using 512 format.	
Element Name	**Element Value**	**Comments**
rfl	'896F0A7CC5FC700A8B9F7FF452463340'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_203_Ad	
Structured Type:	CCHD	
Derivation Path:		
Comments:	Cell Channel Description for cell A in TC_26_6_13 with the ARFCN_list={734,741,747,754,759,762,766,767,773,775,779,782,791,798,829,832,844} using 256 format.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'8B6F1473F32FC602C59EFFA549994000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_203_Adiei	
Structured Type:	CCHD	
Derivation Path:	CellChDes_01.	
Comments:	Cell Channel Description for cell A in TC_26_6_13 with the ARFCN_list={734,741,747,754,759,762,766,767,773,775,779,782,791,798,829,832,844}using 256 format.	
Element Name	**Element Value**	**Comments**
rfl	'8B6F1473F32FC602C59EFFA549994000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_202_Adman	
Structured Type:	CCHD	
Derivation Path:		
Comments:	Cell Channel Description for cell A in RR testing of DCS1800 HO cases with the Complete Cell Allocation of Cell A in HO cases using 512 format. ARFCN_list={734,741,747,754,759,762,766,767,773,775,779,782,791,798,829,832,844}	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'896F0A7CC5FC700A8B9F7FF452463340'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_20_B	
Structured Type:	CCHD	
Derivation Path:	CellChDes_01.	
Comments:	for cell B in RR testing of GSM 900 HO cases with the complete Cell Allocation ARFCN_list={14,18,22,24,30,31,38,40,60,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
rfl	'0002080000000F02080000A060A22000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_20_Bman	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell B in RR testing of GSM 900 HO cases with the complete Cell Allocation ARFCN_list={14,18,22,24,30,31,38,40,60,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'0002080000000F02080000A060A22000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_20_Be(par_rfl:OCTETSTRING)	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell B in RR testing of EGSM HO cases with the complete Cell Allocation ARFCN_list={par_rfl}.	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	par_rfl	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_201_Bd	
Structured Type:	CCHD	
Derivation Path:		
Comments:	Cell Channel Description for cell B in RR testing of DCS1800 HO cases with the ARFCN_list={739,743,746,749,756,758,761,764,771,779,782,791,798,829,832,844} using 256 format.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'8A71907137B602C5DEF7A348D8000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_201_Bdiei	
Structured Type:	CCHD	
Derivation Path:	CellChDes_01.	
Comments:	Cell Channel Description for cell B in RR testing of DCS1800 HO cases with the ARFCN_list={739,743,746,749,756,758,761,764,771,779,782,791,798,829,832,844} using 256 format.	
Element Name	**Element Value**	**Comments**
rfl	'8A71907137B602C5DEF7A348D8000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_20_B0d	
Structured Type:	CCHD	
Derivation Path:		
Comments:	the complete Cell Allocation List_ARFCN= complete cell allocation of cell B in HO cases. {739,743,746,749,756,758,761,764,771,779,782,791,798,829,832,844}	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F71892052408090081000000024004 0'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_21_B	
Structured Type:	CCHD	
Derivation Path:	CellChDes_01.	
Comments:	for any cell in RR testing of GSM 900 HO cases with the ARFCN_list={40,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
rfl	'0002080000000F020000008000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_21_Bd	
Structured Type:	CCHD	
Derivation Path:	CellChDes_01.	
Comments:	for cell B in RR testing of GSM 900 HO cases with the List_ARFCN={761,764,771,779,782,791,798,829,832}	
Element Name	**Element Value**	**Comments**
rfl	'897C87BD09BC61060F90000000000000 'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_22	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in EGSM testing: 20, 30, 50, 70	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'00000000000000200002000020080000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_23A	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in EGSM testing, TC_26_10_2_4_1 : 20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'0002080000007F020000008000080000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_23B	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in EGSM testing, TC_26_10_2_4_1 : 10, 12, 40, 60, 62, 63	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'00000000000000004800008000000A00' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_r01	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in RR testing of DCS1800 with the ARFCN_list={773, 775, 779, 782, 791, 798, 829, 832, 844}.	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8D8299C22EF52CC00000000000000000' 'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_r02	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in RR testing of GSM 900 with the ARFCN_list={741, 747}.	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'8F728200000000000000000000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_r03	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in RR testing of GSM 900 with the ARFCN_list={45, 46, 52, 59, 66, 73, 74, 75, 76, 108, 114}.	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'0002080000000F020408300000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_r04	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in RR testing of GSM 900 with the ARFCN_list={17, 20}.	
Element Name	**Element Value**	**Comments**
iei	'01100010'B	
rfl	'00000000000000000000000000090000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellDescrp(ncc1, bcc1:B_3; par_arfcn: INTEGER)	
Structured Type:	CD	
Derivation Path:		
Comments:	BCCH channel for Cell A	
Element Name	**Element Value**	**Comments**
bcch_arfcn_h	OC_MostBits(INT_TO_BIT(par_arfcn, 10), 2)	
ncc	ncc1	
bcc	bcc1	
bcch_arfcn_l	OC_LeastBits(INT_TO_BIT(par_arfcn, 10), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellOpt_01	
Structured Type:	CO	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
sprb	'0'B	
pwrc	'0'B	
dtx	'10'B	
rlt	'0001'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellOpt_02	
Structured Type:	CO	
Derivation Path:		
Comments:	parameters from 26.3.1 of GSM 11.10	
Element Name	**Element Value**	**Comments**
sprb	'0'B	
pwrc	'0'B	
dtx	'10'B	
rlt	'0101'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellOpt_03	
Structured Type:	CO	
Derivation Path:		
Comments:	used in measurement testing	
Element Name	**Element Value**	**Comments**
sprb	'0'B	
pwrc	'1'B	
dtx	'01'B	
rlt	'0001'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellOpt_04	
Structured Type:	CO	
Derivation Path:		
Comments:	spare bit is set to '1'B	
Element Name	**Element Value**	**Comments**
sprb	'1'B	
pwrc	'0'B	
dtx	'10'B	
rlt	'0001'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellOpt_05	
Structured Type:	CO	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
sprb	'0'B	
pwrc	'0'B	
dtx	'10'B	
rlt	'1111'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellOpt_06	
Structured Type:	CO	
Derivation Path:	CellOpt_04.	
Comments:	spare bit is set to '1'B	
Element Name	**Element Value**	**Comments**
pwrc	'1'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellSelPara(crh, mtmc:INTEGER; neci:B_1)	
Structured Type:	CSP	
Derivation Path:		
Comments:	Default value.	
Element Name	**Element Value**	**Comments**
crh	INT_TO_BIT(crh, 3)	
mtmc	INT_TO_BIT(mtmc, 5)	
acs	'0'B	
neci	neci	
ram	'000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellSelPara_01	
Structured Type:	CSP	
Derivation Path:		
Comments:	Default value for GSM900.	
Element Name	**Element Value**	**Comments**
crh	'110'B	
mtmc	'10011'B	
acs	'0'B	
neci	'0'B	
ram	'000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellSelPara_04	
Structured Type:	CSP	
Derivation Path:	CellSelPara_01.	
Comments:	Default value for DCS1800.	
Element Name	**Element Value**	**Comments**
mtmc	'01111'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cgpn_01	
Structured Type:	CGPN	
Derivation Path:		
Comments:	calling party BCD number with arbitrary spare bits	
Element Name	**Element Value**	**Comments**
iei	'01011100'B	
iel	'03'O	
tonnpi	TonNpi_01	
pisi	PiSi_01	
digits	'01'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cgps_01	
Structured Type:	CGPS	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'01011101'B	
iel	'03'O	
subad	SubAdd_01	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChDescrp_nfh(type:CH_TDMA; slot:SN; tsc:TSC; par_arfcn: INTEGER)	
Structured Type:	CHD	
Derivation Path:		
Comments:	Channel description for CC testing, non hopping, no starting time.	
Element Name	**Element Value**	**Comments**
iei	-	
cht_schn	type	
tn	slot	
tsc	tsc	
hch	'0'B	
maio	-	
hsn	-	
spr	'00'B	
arfcn	INT_TO_BIT(par_arfcn, 10)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChDescrp_nfhiei(type:CH_TDMA; slot:SN; tsc:TSC; par_arfcn: INTEGER)	
Structured Type:	CHD	
Derivation Path:		
Comments:	Channel description for CC testing, non hopping, no starting time, with iei.	
Element Name	**Element Value**	**Comments**
iei	'01100100'B	
cht_schn	type	
tn	slot	
tsc	tsc	
hch	'0'B	
maio	-	
hsn	-	
spr	'00'B	
arfcn	INT_TO_BIT(par_arfcn, 10)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChDescrp_fh(par_chtype: CH_TDMA; ts_ccch: SN; tsc:TSC; maio:MAIO; hsn:HSN)	
Structured Type:	CHD	
Derivation Path:		
Comments:	A derived constraint for <<Channel description >>ie with TCH/F freq. hopping in any CELL.	
Element Name	**Element Value**	**Comments**
iei	-	
cht_schn	par_chtype	
tn	ts_ccch	
tsc	tsc	
hch	'1'B	
maio	maio	
hsn	hsn	
spr	-	
arfcn	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChDescrp_fhiei(par_chtype: CH_TDMA; ts_ccch: SN; tsc:TSC; maio:MAIO; hsn:HSN)	
Structured Type:	CHD	
Derivation Path:		
Comments:	A derived constraint for <<Channel description >>ie with TCH/F freq. hopping in any CELL.	
Element Name	**Element Value**	**Comments**
iei	'00011100'B	
cht_schn	par_chtype	
tn	ts_ccch	
tsc	tsc	
hch	'1'B	
maio	maio	
hsn	hsn	
spr	-	
arfcn	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChDescrp_sdcch8_fh(slot:SN; tsc:TSC; sch:B_3; maio:MAIO; hsn:HSN)	
Structured Type:	CHD	
Derivation Path:		
Comments:	Channel description for SDCCH8.	
Element Name	**Element Value**	**Comments**
iei	-	
cht_schn	INT_TO_BIT((8 + BIT_TO_INT(sch)), 5)	
tn	slot	
tsc	tsc	
hch	'1'B	
maio	maio	
hsn	hsn	
spr	-	
arfcn	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChDescrp_sdcch8_nfh(slot:SN; tsc:TSC; subch:B_3; arfcn:INTEGER)	
Structured Type:	CHD	
Derivation Path:		
Comments:	Channel description for SDCCH8 in cell A for RR test.	
Element Name	**Element Value**	**Comments**
iei	-	
cht_schn	INT_TO_BIT((8 + BIT_TO_INT(subch)), 5)	
tn	slot	
tsc	tsc	
hch	'0'B	
maio	-	
hsn	-	
spr	'00'B	
arfcn	INT_TO_BIT(arfcn, 10)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChDescrp_tchf_fh(slot:SN; tsc:TSC; maio:MAIO; hsn:HSN)	
Structured Type:	CHD	
Derivation Path:		
Comments:	Channel description for TCH/F in cell A for RR test.	
Element Name	**Element Value**	**Comments**
iei	-	
cht_schn	'00001'B	
tn	slot	
tsc	tsc	
hch	'1'B	
maio	maio	
hsn	hsn	
spr	-	
arfcn	-	
Detailed Comments:	Used in TC_26_6_4_1 only	

Structured Type Constraint		
Constraint Name:	ChDescrp_tchh_fh(slot:SN; tsc:TSC; subch:B_1; maio:MAIO; hsn:HSN)	
Structured Type:	CHD	
Derivation Path:		
Comments:	Channel description for TCH/H in cell A for RR test.	
Element Name	**Element Value**	**Comments**
iei	-	
cht_schn	INT_TO_BIT((2 + BIT_TO_INT(subch)), 5)	
tn	slot	
tsc	tsc	
hch	'1'B	
maio	maio	
hsn	hsn	
spr	-	
arfcn	-	
Detailed Comments:	Used in TC_26_6_4_1 only	

Structured Type Constraint		
Constraint Name:	ChDescrp_tchh_nfh(slot:SN; tsc:TSC; subch:B_1; arfcn:INTEGER)	
Structured Type:	CHD	
Derivation Path:		
Comments:	Channel description for TCH/F in cell A for RR test.	
Element Name	**Element Value**	**Comments**
iei	-	
cht_schn	INT_TO_BIT((2 + BIT_TO_INT(subch)), 5)	
tn	slot	
tsc	tsc	
hch	'0'B	
maio	-	
hsn	-	
spr	'00'B	
arfcn	INT_TO_BIT(arfcn, 10)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod(mode:B_8)	
Structured Type:	CHMOD	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
mode	mode	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_iei(mode:B_8)	
Structured Type:	CHMOD	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'01100011'B	
mode	mode	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_r01	
Structured Type:	CHMOD	
Derivation Path:		
Comments:	for TC_26_6_13_1 after time	
Element Name	**Element Value**	**Comments**
iei	'01100011'B	
mode	TSPX_ChMod1	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_r02	
Structured Type:	CHMOD	
Derivation Path:		
Comments:	for TC_26_6_13_2 after time	
Element Name	**Element Value**	**Comments**
iei	'01100011'B	
mode	TSPX_ChMod2	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_r03	
Structured Type:	CHMOD	
Derivation Path:	ChMod_r01.	
Comments:	for TC_26_6_13_5	
Element Name	**Element Value**	**Comments**
mode	TSPX_ChMod4	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_r04	
Structured Type:	CHMOD	
Derivation Path:	ChMod_r01.	
Comments:	for TC_26_6_13_6	
Element Name	**Element Value**	**Comments**
mode	TSPX_ChMod5	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_r05	
Structured Type:	CHMOD	
Derivation Path:	ChMod_r01.	
Comments:	for TC_26_6_13_7	
Element Name	**Element Value**	**Comments**
mode	TSPX_ChMod6	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_r06	
Structured Type:	CHMOD	
Derivation Path:	ChMod_r01.	
Comments:	for TC_26_6_13_8	
Element Name	**Element Value**	**Comments**
mode	TSPX_ChMod7	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_sign	
Structured Type:	CHMOD	
Derivation Path:		
Comments:	signalling only	
Element Name	**Element Value**	**Comments**
iei	-	
mode	C_ChMod_s	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_sign_iei	
Structured Type:	CHMOD	
Derivation Path:		
Comments:	signalling only	
Element Name	**Element Value**	**Comments**
iei	'01100011'B	
mode	C_ChMod_s	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_speech	
Structured Type:	CHMOD	
Derivation Path:	ChMod_sign.	
Comments:	speech full or half rate.	
Element Name	**Element Value**	**Comments**
mode	C_ChMod_r	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_speech_iei	
Structured Type:	CHMOD	
Derivation Path:		
Comments:	speech full or half rate.	
Element Name	**Element Value**	**Comments**
iei	'01100011'B	
mode	C_ChMod_r	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_12k	
Structured Type:	CHMOD	
Derivation Path:	ChMod_sign.	
Comments:	data 9.6 kb/s full rate, (12.0 kb/s air interface)	
Element Name	**Element Value**	**Comments**
mode	C_ChMod_12k	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_6k	
Structured Type:	CHMOD	
Derivation Path:	ChMod_sign.	
Comments:	data 4.8 kb/s full rate, (6.0 kb/s air interface)	
Element Name	**Element Value**	**Comments**
mode	C_ChMod_6k	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_3k	
Structured Type:	CHMOD	
Derivation Path:	ChMod_sign.	
Comments:	data 2.4 kb/s full rate, (3.6 kb/s air interface)	
Element Name	**Element Value**	**Comments**
mode	C_ChMod_3k	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_rcv	
Structured Type:	CHMOD	
Derivation Path:	ChMod_sign.	
Comments:	used only in the OM_ChConf operation,	
Element Name	**Element Value**	**Comments**
mode	'11111111'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_omit	
Structured Type:	CHMOD	
Derivation Path:		
Comments:	parametrised mode.	
Element Name	**Element Value**	**Comments**
iei	-	
mode	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Chneed_01	
Structured Type:	CHNEED	
Derivation Path:		
Comments:	any channel	
Element Name	**Element Value**	**Comments**
ch2	'00'B	
ch1	'00'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Chneed_02	
Structured Type:	CHNEED	
Derivation Path:		
Comments:	SDCCH channel needed.	
Element Name	**Element Value**	**Comments**
ch2	'00'B	
ch1	'01'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Chneed_03	
Structured Type:	CHNEED	
Derivation Path:		
Comments:	TCH/F channel needed.	
Element Name	**Element Value**	**Comments**
ch2	'00'B	
ch1	'10'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Chneed_04	
Structured Type:	CHNEED	
Derivation Path:		
Comments:	Dual rate channel needed.	
Element Name	**Element Value**	**Comments**
ch2	'00'B	
ch1	'11'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CiphRes_01	
Structured Type:	CPH_RES	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
sprb	'000'B	
cr	'0'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CiphRes_02	
Structured Type:	CPH_RES	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
sprb	'000'B	
cr	'1'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ClassMark1	
Structured Type:	MSCLM1	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
spr1	'0'B	
rl	TSPX_RevLevel	
spr2	'0'B	
a5_1	TSPX_CiphAlgA5_1	
rfpc	TSPX_RfPwrCap	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ClassMark2	
Structured Type:	MSCLM2	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iel	'03'O	
spr1	'0'B	
rl	TSPX_RevLevel	
spr2	'0'B	
a5_1	TSPX_CiphAlgA5_1	
rfpc	TSPX_RfPwrCap	
spr3	'0'B	
psc	TSPX_pSyncCap	
sssi	TSPX_SSscrnInd	
smc	TSPX_SMCap	
spr4	'00'B	
fc	TSPX_frqCap	
cm3	TSPX_ClsMk3	
spr5	'00000'B	
a5_3	TSPX_CiphAlgA5_3	
a5_2	TSPX_CiphAlgA5_2	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ClassMark2Amp	
Structured Type:	MSCLM2	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iel	'03'O	
spr1	'0'B	
rl	TSPX_RevLevel	
spr2	'0'B	
a5_1	TSPX_CiphAlgA5_1	
rfpc	TSPX_RfPwrCapAmp	
spr3	'0'B	
psc	TSPX_pSyncCap	
sssi	TSPX_SSscrnInd	
smc	TSPX_SMCap	
spr4	'00'B	
fc	TSPX_frqCap	
cm3	TSPX_ClsMk3	
spr5	'00000'B	
a5_3	TSPX_CiphAlgA5_3	
a5_2	TSPX_CiphAlgA5_2	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ClassMark3	
Structured Type:	MSCLM3	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'00100000'B	
iel	'0D'O	
spr1	'0000'B	
a5_7	TSPX_CiphAlgA5_7	
a5_6	TSPX_CiphAlgA5_6	
a5_5	TSPX_CiphAlgA5_5	
a5_4	TSPX_CiphAlgA5_4	
spr2	'00000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CntrlChDscrp_inv	
Structured Type:	CCD	
Derivation Path:		
Comments:	One CCCH combined with SDCCH and attach/detach not allowed. Some spare bits are set to '1'B	
Element Name	**Element Value**	**Comments**
spr1	'1'B	
att	'0'B	
babr	'000'B	
ccch_con	'001'B	
spr2	'01100'B	
bpm	'011'B	
t3212	'00'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CntrlChDscrp(att: INTEGER; babr, cch_con, bpm:B_3; timer:OCTETSTRING)	
Structured Type:	CCD	
Derivation Path:		
Comments:	Generic Control channel descriptor.	
Element Name	**Element Value**	**Comments**
spr1	'0'B	
att	INT_TO_BIT(att,1)	
babr	babr	
ccch_con	cch_con	
spr2	'00000'B	
bpm	bpm	
t3212	timer	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphKeySN_01(cksn:B_3)	
Structured Type:	CPHKSN	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
sprb	'0'B	
ks	cksn	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphKeySN_02	
Structured Type:	CPHKSN	
Derivation Path:		
Comments:	An invalid ciphering key sequence number containing spare bit set to '1'B.	
Element Name	**Element Value**	**Comments**
sprb	'1'B	
ks	'000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphKeySN_06	
Structured Type:	CPHKSN	
Derivation Path:		
Comments:	ciphering key sequence number no key.	
Element Name	**Element Value**	**Comments**
sprb	'0'B	
ks	'111'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphKeySN_07(par: BITSTRING)	
Structured Type:	CPHKSN	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
sprb	'0'B	
ks	par	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphMod_omit	
Structured Type:	CPHMS	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
algid	-	
sc	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphMod_01	
Structured Type:	CPHMS	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
algid	TSPX_CphAlgDef	
sc	'1'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphMod_02	
Structured Type:	CPHMS	
Derivation Path:	CphMod_01.	
Comments:	No ciphering	
Element Name	**Element Value**	**Comments**
sc	'0'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphMod_02iei	
Structured Type:	CPHMS	
Derivation Path:		
Comments:	No ciphering	
Element Name	**Element Value**	**Comments**
iei	'1001'B	
algid	TSPX_CphAlgDef	
sc	'0'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphMod_03	
Structured Type:	CPHMS	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
algid	'000'B	
sc	'1'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphMod_04(alg:BITSTRING)	
Structured Type:	CPHMS	
Derivation Path:		
Comments:	starting ciphering with the ciphering algorithm `alg`.	
Element Name	**Element Value**	**Comments**
iei	-	
algid	alg	
sc	'1'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphMod_04iei(alg:BITSTRING)	
Structured Type:	CPHMS	
Derivation Path:		
Comments:	starting ciphering with the ciphering algorithm `alg`.	
Element Name	**Element Value**	**Comments**
iei	'1001'B	
algid	alg	
sc	'1'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	facilityIErcv(comp:Component)	
Structured Type:	FIE	
Derivation Path:		
Comments:	A received FIE may contain several components, but at least contains "comp".	
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
components_1	Comps_01(comp)	
components_t	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	facilityIErcviei(comp:Component)	
Structured Type:	FIE	
Derivation Path:		
Comments:	A received FIE may contain several components, but at least contains "comp".	
Element Name	**Element Value**	**Comments**
iei	'00011100'B	
iel	?	
components_1	Comps_01(comp)	
components_t	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	facilityIEtsnd(comp:Component_T)	
Structured Type:	FIE	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_LengthOfComp(comp)	
components_1	-	
components_t	comp	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	facilityIEtsndiei(comp:Component_T)	
Structured Type:	FIE	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'00011100'B	
iel	OC_LengthOfComp(comp)	
components_1	-	
components_t	comp	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Fn_01	
Structured Type:	FN	
Derivation Path:		
Comments:	not pertaining to the MS under test	
Element Name	**Element Value**	**Comments**
t1_	'00000'B	
t3	'000000'B	
t2	'00000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqBCCH(arfcn:INTEGER)	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Broadcast channel.	
Element Name	**Element Value**	**Comments**
hch	'0'B	
maio	-	
hsn	-	
spr	'00'B	
arfcn	INT_TO_BIT(arfcn, 10)	
maclength	'00'O	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freqchseq_omit	
Structured Type:	FRQCHS	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
larfcn	-	
incs	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freqchseq_01	
Structured Type:	FRQCHS	
Derivation Path:		
Comments:	Frequency channel sequence with the f_list={40,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	'01101001'B	
larfcn	'28'O	
incs	'0B71110026000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freqchseq_02	
Structured Type:	FRQCHS	
Derivation Path:		
Comments:	Frequency channel sequence with the List_ARFCN={14,18,22,24,30,31,38,40,66,73,74,75, 76,108} cross reference: Frql_20_B72	
Element Name	**Element Value**	**Comments**
iei	'01101001'B	
larfcn	'0E'O	
incs	'44261720B7111002'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freqchseq_03	
Structured Type:	FRQCHS	
Derivation Path:		
Comments:	Frequency channel sequence with the List_ARFCN={14,114}	
Element Name	**Element Value**	**Comments**
iei	'01101001'B	
larfcn	'0E'O	
incs	'000000A000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freqchseq_04	
Structured Type:	FRQCHS	
Derivation Path:		
Comments:	Frequency channel sequence with the List_ARFCN={40,114}	
Element Name	**Element Value**	**Comments**
iei	'01101001'B	
larfcn	'28'O	
incs	'0000E00000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freqchseq_05	
Structured Type:	FRQCHS	
Derivation Path:		
Comments:	Frequency channel sequence with the List_ARFCN={66, 75, 76, 108}	
Element Name	**Element Value**	**Comments**
iei	'01101001'B	
larfcn	'42'O	
incs	'9100200000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freqchseq_06	
Structured Type:	FRQCHS	
Derivation Path:		
Comments:	Frequency channel sequence with the List_ARFCN={14,18,22,24,30,31,38,53,66,73,74,75,76,108,114} see Frql_20_B5	
Element Name	**Element Value**	**Comments**
iei	'01101001'B	
larfcn	'0E'O	
incs	'442617FD71110026'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freqchseq_22	
Structured Type:	FRQCHS	
Derivation Path:		
Comments:	Frequency channel sequence with the List_ARFCN={10,17,20,26,59,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	'01101001'B	
larfcn	'0A'O	
incs	'7360037711100260'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCH(arfcn:INTEGER)	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Default Traffic channel non frequency hopping	
Element Name	**Element Value**	**Comments**
hch	'0'B	
maio	-	
hsn	-	
spr	'00'B	
arfcn	INT_TO_BIT(arfcn, 10)	
maclength	'00'O	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCH_omit	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Omitted FRQPARA when not needed in parameters list.	
Element Name	**Element Value**	**Comments**
hch	-	
maio	-	
hsn	-	
spr	-	
arfcn	-	
maclength	-	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa1	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Traffic channel for cell A, ARFCN = 124 hopping, for GSM900.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000000'B	
hsn	'111111'B	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'00000001'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_04	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa2	
Structured Type:	FRQPARA	
Derivation Path:	FreqTCHa1.	
Comments:	Traffic channel for cell A, ARFCN = 801 hopping, for DCS1800	
Element Name	**Element Value**	**Comments**
flst	Frql_05	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa3	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Traffic channel for cell A, hopping, GSM900 for TC_26_6_4_1	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000001'B	
hsn	'000001'B	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00000011'B	
mac_7n	'11111011'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_10	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa4	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Traffic channel for cell A, hopping, DCS1800 for TC_26_6_4_1	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000000'B	
hsn	'101000'B	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'00000001'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_15	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa5	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Traffic channel for cell A, hopping, GSM900 for TC_26_6_4_1	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000101'B	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'00111111'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_09	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa6	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Traffic channel for cell A, hopping, GSM900 for TC_26_6_4_1	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000011'B	
hsn	'001000'B	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00000001'B	
mac_7n	'11000111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_14	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa7	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Traffic channel for cell A, hopping, GSM900 for TC_26_6_4_1	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000011'B	
hsn	'001000'B	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00000111'B	
mac_7n	'11100011'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_11	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa8	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Traffic channel for cell A, hopping, DCS1800 for TC_26_6_4_1	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000001'B	
hsn	'000001'B	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00000011'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_13	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa9	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Traffic channel for cell A, hopping, GSM900 for TC_26_6_4_1	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000101'B	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'00111111'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_08	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa10	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Traffic channel for cell A, hopping, GSM900 for TC_26_6_4_1	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000000'B	
hsn	'101000'B	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'00000001'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_12	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_21g	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	GSM 900 traffic channel of cell A, hopping for TC_26_5_6_3.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000000'B	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'00000010'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_38	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_21d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	DCS 1800 traffic channel of cell A, hopping for TC_26_5_6_3.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000000'B	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'00000010'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_38d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCH_ef1(par_ma: BITSTRING; par_freqlist: OCTETSTRING; par_flistl: LENGTH; n :INTEGER)	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Broadcast channel of cell A for EGSM.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD n), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	par_ma	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_egsm(par_freqlist, par_flistl)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCH_ef2(par_ma1: BITSTRING;par_ma2: BITSTRING; par_freqlist: OCTETSTRING; par_flistl: LENGTH; n :INTEGER)	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Broadcast channel of cell A for EGSM.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD n), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	par_ma2	
mac_7n	par_ma1	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_egsm(par_freqlist, par_flistl)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof1	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping traffic channel of cell A for GSM.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 12), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00001111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof1d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping channel of cell A for DCS1800	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 9), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_Ad	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof2	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping full rate channel of cell A, but only one frequency (114) for GSM.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000000'B	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000001'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof2d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping full rate channel of cell A but only one frequency (844) for DCS.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000000'B	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000001'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof3	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping channel of cell A with hopping frequencies : {10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 17), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	'11111111'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof3d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping channel of cell A. Indicating complete Cell Allocation of Cell A (DCS)	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 17), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	'11111111'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof4	
Structured Type:	FRQPARA	
Derivation Path:	FreqTCHa_hof3.	
Comments:	a hopping channel of cell A with hopping frequencies : {10, 17, 26, 34, 42, 45, 46, 52, 59, 66, 73, 74, 75, 76, 108, 114}	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 16), 6)	
hsn	INT_TO_BIT(C_HSN_0, 6)	
mac_6n	'11111011'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof4d	
Structured Type:	FRQPARA	
Derivation Path:	FreqTCHa_hof3d.	
Comments:	a hopping channel of cell A. Indicating complete Cell Allocation of Cell A (DCS) {734,741,754,759,762,766,767,773,775,779,782,791,798,829,832,844}	
Element Name	**Element Value**	**Comments**
maio	INT_TO_BIT((TSPX_MAIO MOD 17), 6)	
hsn	INT_TO_BIT(C_HSN_0, 6)	
mac_6n	'11111111'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof1	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Broadcast channel of cell B.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 12), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00001111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B3	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof1d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Broadcast channel of cell B for hopping DCS1800	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 12), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00001111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B3d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof2	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping traffic channel	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 8), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'11111111'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B4	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof2d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Broadcast channel of cell B for hopping DCS1800	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 8), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'11111111'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B4d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof3	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 15), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'01111111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B5	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof3d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 11), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00000111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B5d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof4	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 3), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'00011100'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B4	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof4d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	TC_26_6_5_2_2	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 3), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00000000'B	
mac_7n	'11100000'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B6d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof5	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	full rate hopping channel of cell B for GSM	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 5), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'10100111'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof5d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a full rate hopping channel of cell B for DCS	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 9), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B9d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof6	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 2), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'10000000'B	
mac_7n	'00000001'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof6d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 3), 6)	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00000110'B	
mac_7n	'10000000'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B8d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof7	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	channel of cell B.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 2), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'10000000'B	
mac_7n	'10000000'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof7d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 15), 6)	
hsn	INT_TO_BIT(C_HSN_0, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'01111111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_11d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof8	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	traffic channel of cell B. Hopping on List_ARFCN={66, 75, 76, 108}	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 4), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'01110010'B	
mac_7n	'00000000'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof8d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	traffic channel of cell B. List_ARFCN={758,761,771}	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 3), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'00000111'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B12d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof9	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	full rate hopping channel of cell B for GSM	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 16), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'11111111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B2	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof9d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a full rate hopping channel of cell B for DCS	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 13), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00111111'B	
mac_7n	'10111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B0d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCHa_hof1	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping with {73,74,75} of cell allocation A.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 3), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000000'B	
mac_7n	'00111000'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCHa_hof1d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping channel of cell A with {773,775,779} of cell allocation A.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 3), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000000'B	
mac_7n	'00000111'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCHa_hof2	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping channel with hopping frequencies are the Complete Cell Allocation of cell A except for BCCH. For GSM	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 16), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	'11111011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCHa_hof2d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping channel of cell A. the hopping frequencies are the Complete Cell Allocation of cell A except for BCCH. For DCS.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 16), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	'11111011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCHa_hof3	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping channel with the Complete Cell Allocation of cell A except 20 and 52.(GSNM)	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 15), 6)	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111110'B	
mac_6n	'11111011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCHa_hof3d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping channel of cell A with the Complete Cell Allocation of cell A with except 747 and 767. (DCS)	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 15), 6)	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	'01111011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCH8b_hof1	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping SDCCH8 channel	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 15), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'01111111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B7	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCH8b_hof1d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping SDCCH8 channel	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 2), 6)	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'00000011'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B7d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCH8b_hof2	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping SDCCH8 channel	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 14), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00111111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B72	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCH8b_hof2d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping SDCCH8 channel in cell B, hopping frequencies are the complete cell allocation of cell B (512 format)	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 16), 6)	
hsn	INT_TO_BIT(C_HSN_0, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'11111111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B0d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCH8b_hof3	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping SDCCH8 channel including complete cell allocation of cell B.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 16), 6)	
hsn	INT_TO_BIT(C_HSN_0, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'11111111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCH8_rg1	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping SDCCH8 channel of TC_26_6_13_1 for GSM .	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT(TSPX_Maio1, 6)	
hsn	INT_TO_BIT(TSPX_Hsn1, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma1) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma1) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma1) MOD 256), 8)	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCH8_rg2	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.	
Comments:	hopping SDCCH8 channel of TC_26_6_13_2 for GSM .	
Element Name	**Element Value**	**Comments**
maio	INT_TO_BIT(TSPX_Maio4, 6)	
hsn	INT_TO_BIT(TSPX_Hsn4, 6)	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma4) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma4) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma4) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCH8_rd1	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.	
Comments:	hopping SDCCH8 channel of TC_26_6_13_1 for DCS .	
Element Name	**Element Value**	**Comments**
flst	Frql_20_A1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCH8_rd2	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.FreqSDCCH8_rg2.	
Comments:	hopping SDCCH8 channel of TC_26_6_13_2 for DCS .	
Element Name	**Element Value**	**Comments**
flst	Frql_20_A1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCH8_e	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping SDCCH8 channel of TC_26_6_10_3 for EGSM .	
Element Name	**Element Value**	**Comments**
hch	'0'B	
maio	-	
hsn	-	
spr	'00'B	
arfcn	INT_TO_BIT(40, 10)	
maclength	'00'O	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCH8_e1	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	SDCCH8 channel of TC_26_6_10_2_2 for EGSM .	
Element Name	**Element Value**	**Comments**
hch	'0'B	
maio	-	
hsn	-	
spr	'00'B	
arfcn	INT_TO_BIT(1015, 10)	
maclength	'00'O	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCH8_e2	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping with indicates to the ARFCN_list={0, 80, 1005, 1010} of cell allocation A.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 4), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'10111000'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0E	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg2	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.	
Comments:	hopping channel of TC_26_6_13_1 for GSM , before time.	
Element Name	**Element Value**	**Comments**
maio	INT_TO_BIT(TSPX_Maio3, 6)	
hsn	INT_TO_BIT(TSPX_Hsn3, 6)	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma3) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma3) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma3) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd2	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.Freq_rg2.	
Comments:	hopping channel of TC_26_6_13_1 for DCS, before time.	
Element Name	**Element Value**	**Comments**
flst	Frql_20_A1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg4	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.	
Comments:	hopping channel of TC_26_6_13_2 for GSM, after time.	
Element Name	**Element Value**	**Comments**
maio	INT_TO_BIT(TSPX_Maio5, 6)	
hsn	INT_TO_BIT(TSPX_Hsn5, 6)	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma5) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma5) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma5) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd4	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.Freq_rg4.	
Comments:	hopping channel of TC_26_6_13_2 for DCS, after time.	
Element Name	**Element Value**	**Comments**
flst	Frql_20_A1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg5	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping channel of TC_26_6_13_3 for GSM, immediate assignment.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT(TSPX_Maio6, 6)	
hsn	INT_TO_BIT(TSPX_Hsn6, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma6) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma6) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma6) MOD 256), 8)	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd5	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg5.	
Comments:	hopping channel of TC_26_6_13_3 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_20_A1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg7	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.	
Comments:	hopping channel of TC_26_6_13_4 for GSM, immediate assignment.	
Element Name	**Element Value**	**Comments**
maio	INT_TO_BIT(TSPX_Maio10, 6)	
hsn	INT_TO_BIT(TSPX_Hsn10, 6)	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma10) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma10) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma10) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd7	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.Freq_rg7.	
Comments:	hopping channel of TC_26_6_13_4 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_20_A1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg8	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping channel of TC_26_4_2_2 for GSM, assignment.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000000'B	
hsn	'000100'B	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'00001110'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_07	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd8	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg8.	
Comments:	hopping channel of TC_26_6_4_2_2 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_17	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg9	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.	
Comments:	hopping channel of TC_26_6_13_5 for GSM, immediate assignment. cell A	
Element Name	**Element Value**	**Comments**
maio	INT_TO_BIT(TSPX_Maio14, 6)	
hsn	INT_TO_BIT(TSPX_Hsn14, 6)	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma14) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma14) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma14) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd9	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.Freq_rg9.	
Comments:	hopping channel of TC_26_6_13_5 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_20_A1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg10	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping channel of TC_26_6_13_5 for GSM, hanfover command after time in Cell B	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT(TSPX_Maio15, 6)	
hsn	INT_TO_BIT(TSPX_Hsn15, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma15) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma15) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma15) MOD 256), 8)	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg11	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg10.	
Comments:	hopping channel of TC_26_6_13_5 for GSM, hanfover command before time in Cell B	
Element Name	**Element Value**	**Comments**
maio	INT_TO_BIT(TSPX_Maio16, 6)	
hsn	INT_TO_BIT(TSPX_Hsn16, 6)	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma16) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma16) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma16) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd11	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg10.Freq_rg11.	
Comments:	hopping channel of TC_26_6_13_5 for DCS, handover command before time in Cell B	
Element Name	**Element Value**	**Comments**
flst	Frql_20_B1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg12	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.	
Comments:	hopping channel of TC_26_6_13_6 for GSM, immediate assignment. cell A	
Element Name	**Element Value**	**Comments**
maio	INT_TO_BIT(TSPX_Maio17, 6)	
hsn	INT_TO_BIT(TSPX_Hsn17, 6)	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma17) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma17) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma17) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd12	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.Freq_rg12.	
Comments:	hopping channel of TC_26_6_13_6 for DCS, immediate assignment. Cell A	
Element Name	**Element Value**	**Comments**
flst	Frql_20_A1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg13	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg10.	
Comments:	hopping channel of TC_26_6_13_6 for GSM, hanfover command after time in Cell B	
Element Name	**Element Value**	**Comments**
maio	INT_TO_BIT(TSPX_Maio18, 6)	
hsn	INT_TO_BIT(TSPX_Hsn18, 6)	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma18) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma18) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma18) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd13	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg10.	
Comments:	hopping channel of TC_26_6_13_6 for DCS, handover command after time in Cell B	
Element Name	**Element Value**	**Comments**
flst	Frql_20_B1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg14	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.	
Comments:	hopping channel of TC_26_6_13_7 for GSM, immediate assignment.	
Element Name	**Element Value**	**Comments**
maio	INT_TO_BIT(TSPX_Maio20, 6)	
hsn	INT_TO_BIT(TSPX_Hsn20, 6)	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma20) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma20) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma20) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd14	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.Freq_rg14.	
Comments:	hopping channel of TC_26_6_13_7 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_20_A1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg15	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.	
Comments:	hopping channel of TC_26_6_13_8 for GSM, immediate assignment.	
Element Name	**Element Value**	**Comments**
maio	INT_TO_BIT(TSPX_Maio24, 6)	
hsn	INT_TO_BIT(TSPX_Hsn24, 6)	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma24) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma24) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma24) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd15	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.Freq_rg15.	
Comments:	hopping channel of TC_26_6_13_8 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_20_A1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg16	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.	
Comments:	hopping channel of TC_26_6_13_9 for GSM, immediate assignment before time.	
Element Name	**Element Value**	**Comments**
maio	INT_TO_BIT(TSPX_Maio29, 6)	
hsn	INT_TO_BIT(TSPX_Hsn28, 6)	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma29) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma29) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma29) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd16	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.Freq_rg16.	
Comments:	hopping channel of TC_26_6_13_8 for DCS, immediate assignment before time.	
Element Name	**Element Value**	**Comments**
flst	Frql_20_A1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg17	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.	
Comments:	hopping channel of TC_26_6_13_10 for GSM, immediate assignment after time.	
Element Name	**Element Value**	**Comments**
maio	INT_TO_BIT(TSPX_Maio30, 6)	
hsn	INT_TO_BIT(TSPX_Hsn30, 6)	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma30) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma30) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma30) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd17	
Structured Type:	FRQPARA	
Derivation Path:	FreqSDCCH8_rg1.Freq_rg17.	
Comments:	hopping channel of TC_26_6_13_8 for DCS, immediate assignment after time.	
Element Name	**Element Value**	**Comments**
flst	Frql_20_A1d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg18	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping channel of TC_26_6_6_1 for GSM, immediate assignment.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'001001'B	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'08'O	
mac_8n	'00000000'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00000000'B	
mac_2n	'00111111'B	
mac_1n	'11111110'B	
flst	Frql_20	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd18	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.	
Comments:	hopping channel of TC_26_6_6_1 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_29	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg19	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.	
Comments:	hopping channel of TC_26_6_6_1 for GSM, immediate assignment.	
Element Name	**Element Value**	**Comments**
maio	'000100'B	
mac_8n	'10000000'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	'10000000'B	
mac_4n	'00100000'B	
mac_3n	'00000000'B	
mac_2n	'00000000'B	
mac_1n	'00000101'B	
flst	Frql_21	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd19	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.Freq_rg19.	
Comments:	hopping channel of TC_26_6_6_1 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_30	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg20	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.	
Comments:	hopping channel of TC_26_6_6_1 for GSM, immediate assignment.	
Element Name	**Element Value**	**Comments**
maio	'000010'B	
mac_8n	'01000000'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	'00000010'B	
mac_4n	'00000000'B	
mac_3n	'00000000'B	
mac_2n	'00000000'B	
mac_1n	'01100101'B	
flst	Frql_22	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd20	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.Freq_rg20.	
Comments:	hopping channel of TC_26_6_6_1 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_31	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rg21	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.	
Comments:	hopping channel of TC_26_6_6_1 for GSM, immediate assignment.	
Element Name	**Element Value**	**Comments**
maio	'000110'B	
mac_8n	'00000001'B	
mac_7n	'00000000'B	
mac_6n	'01000010'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00000001'B	
mac_2n	'01110000'B	
mac_1n	'00000000'B	
flst	Frql_23	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd21	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.Freq_rg21.	
Comments:	hopping channel of TC_26_6_6_1 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_32	
Detailed Comments:	ARFCN's: 717, ..., 724, 733, ..., 744, 757, ..., 760, 773, ..., 812	

Structured Type Constraint		
Constraint Name:	Freq_rg22	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.	
Comments:	hopping channel of TC_26_6_6_1 for GSM, immediate assignment.	
Element Name	**Element Value**	**Comments**
maio	'000101'B	
mac_8n	'01001000'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00001101'B	
mac_2n	'01000100'B	
mac_1n	'00000001'B	
flst	Frql_24	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd22	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.Freq_rg22.	
Comments:	hopping channel of TC_26_6_6_1 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_33	
Detailed Comments:	64 ARFCN's: 702, ..., 732, 749, 750, 754, ..., 756, 758, 762, ..., 764, 766, 770, ..., 772, 774, 778, ..., 780, 782, 786, ..., 788, 790, 794, ..., 796, 798, 802, ..., 804, 806, 810, ..., 812	

Structured Type Constraint		
Constraint Name:	Freq_rg23	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.	
Comments:	hopping channel of TC_26_6_6_1 for GSM, immediate assignment.	
Element Name	**Element Value**	**Comments**
maio	'001000'B	
mac_8n	'11111111'B	
mac_7n	'10000000'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00000000'B	
mac_2n	'00011111'B	
mac_1n	'00000000'B	
flst	Frql_25	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd23	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.Freq_rg23.	
Comments:	hopping channel of TC_26_6_6_1 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_34	
Detailed Comments:	ARFCN's: 707, 709, ..., 713, 715, 717, ..., 721, 723, 725, ..., 729, 731, 733, ..., 737, 739, 741, ..., 745, 747, 749, ..., 753, 755, 757, ..., 761, 763, 765, ..., 769,771, 779, 787, 795, 798,..., 803, 806, ..., 811	

Structured Type Constraint		
Constraint Name:	Freq_rg24	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.	
Comments:	hopping channel of TC_26_6_6_1 for GSM, immediate assignment.	
Element Name	**Element Value**	**Comments**
maio	'000011'B	
mac_8n	'11111000'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00000000'B	
mac_2n	'00011111'B	
mac_1n	'00000000'B	
flst	Frql_26	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd24	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.Freq_rg24.	
Comments:	hopping channel of TC_26_6_6_1 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_35	
Detailed Comments:	ARFCN's: 705, ..., 736, 749, 751, 753, 755, 757, 759, 761, 763, 765, 767, 769, 771, 773, 775, 777, 779, 781, 783, 785, 787, 789, 791, 793, 795, 797, 799, 801, 803, 805, 807, 809, 811	

Structured Type Constraint		
Constraint Name:	Freq_rg25	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.	
Comments:	hopping channel of TC_26_6_6_1 for GSM, immediate assignment.	
Element Name	**Element Value**	**Comments**
maio	'000111'B	
mac_8n	'00000000'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00001111'B	
mac_2n	'11111110'B	
mac_1n	'00000000'B	
flst	Frql_27	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd25	
Structured Type:	FRQPARA	
Derivation Path:	Freq_rg18.Freq_rg25.	
Comments:	hopping channel of TC_26_6_6_1 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
flst	Frql_36	
Detailed Comments:	ARFCN's: 702, 704, 706, 708, 710, 712, 714, 716, 718, 720, 722, 724, 726, 728, 730, 732, 734, 76, 738, 740, 742, 744, 746, 748, 750, 852, 754, 756, 758, 760, 762, 764, ..., 780, 789, ..., 796, 805, 812	

Structured Type Constraint		
Constraint Name:	Freq_rg26	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping channel of TC_26_6_6_1 for GSM, immediate assignment.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000001'B	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'08'O	
mac_8n	'11111111'B	
mac_7n	'11110000'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00000000'B	
mac_2n	'00000000'B	
mac_1n	'00000000'B	
flst	Frql_28	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freq_rd26	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping channel of TC_26_6_6_1 for DCS, immediate assignment.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'001001'B	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'08'O	
mac_8n	'11111111'B	
mac_7n	'11110000'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00000000'B	
mac_2n	'00000000'B	
mac_1n	'00000000'B	
flst	Frql_37	
Detailed Comments:	ARFCN's: 717, ..., 748, 765, ..., 796	

Structured Type Constraint		
Constraint Name:	Frql(par_flist, par_flistl:LENGTH)	
Structured Type:	FRQL	
Derivation Path:		
Comments:	parameterized frequncy list information element.	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	par_flistl	
fl	par_flist	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_omit	
Structured Type:	FRQL	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
fl	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_short(flist:OCTETSTRING)	
Structured Type:	FRQL	
Derivation Path:		
Comments:	parameterized frequency short list information element.	
Element Name	**Element Value**	**Comments**
iei	'00000010'B	
iel	-	
fl	flist	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_01	
Structured Type:	FRQL	
Derivation Path:		
Comments:	coded as length = 1 and unrecognised IE contents.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'01'O	
fl	'00'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_02	
Structured Type:	FRQL	
Derivation Path:		
Comments:	coded as length = 1 and unrecognised IE contents.	
Element Name	**Element Value**	**Comments**
iei	'11011010'B	
iel	-	
fl	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_04	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list for GSM900 hopping channel with only one frequency, ARFCN = 124	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'08000000000000000000000000000000'O	
Detailed Comments:	used in TC_26_5_7_1_4	

Structured Type Constraint		
Constraint Name:	Frql_05	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list for DCS1800 hopping channel with only one frequency ARFCN 801.	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'03'O	
fl	'8F9080'O	
Detailed Comments:	used in TC_26_5_7_1_4	

Structured Type Constraint		
Constraint Name:	Frql_07	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs 20, 30, 50, 70	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'00000000000000200002000020080000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_08	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs 10, 34, 52, 73, 108, 114.	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'00020800000001000008000200000200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_09	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs 734, 741, 759, 766, 773, 832, 844	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'82FEF390BE7144830000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_10	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs 10, 17, 20, 26, 34, 42, 45, 46, 52, 59	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'00000000000000000408320202090200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_11	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs 45, 46, 52, 59, 66, 73, 74, 75, 76, 108, 114	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020408300000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_12	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs 17, 20	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'00000000000000000000000000090000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_13	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs 734, 741, 747, 754, 759, 766, 773, 775, 779, 782	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'82FEF6837EBF0300FCFEF600000000 0'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_14	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs: 773, 775, 779, 782, 791, 789, 829, 832, 844	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'8315FB0AFF3F40C33F5A000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_15	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs 741, 747	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'82EBFD00000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_17	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs 590, 650, 750, 850	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'EECE193100000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 1.. 64 for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'00000000000000000FFFFFFFFFFFFFFF'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_A0	
Structured Type:	FRQL	
Derivation Path:		
Comments:	Complete Cell Allocation of cell A in HO cases. ARFCN_list={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020408320202090200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_A0d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	Complete Cell Allocation of Cell A in HO cases using 512 format. frequency list, after time for a target cell used in HO_case in cell A for DCS1800. List_ARFCN={734,741,747,754,759,762,766,767,773,775,779,782,791,798,829,832,844}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'896F0A7CC5FC700A8B9F7FF452463340'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_egsm(par_flist: OCTETSTRING; par_flistl: LENGTH)	
Structured Type:	FRQL	
Derivation Path:		
Comments:	Complete Cell Allocation in EGSM cases.	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	par_flistl	
fl	par_flist	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_A	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time used in HO_case in cell A. List_ARFCN={10,17,20,26,59,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020400000002090200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_Ad	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list used in HO_case in cell A for DCS1800 using 256 format for the List_ARFCN={747, 775, 779, 782, 791, 798, 829, 832, 844}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'09'O	
fl	'8B7599F045EFA499C0'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_A1d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B used in TC_26_6_13 for cell A (DCS) using 256 format for the List_ARFCN = {734, 741, 747, 754, 759, 762, 766, 767, 773, 775, 779, 782, 791, 798, 829, 832, 844}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'0F'O	
fl	'8B6F1473F32FC602C59EFFA5499940'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_A2	
Structured Type:	FRQL	
Derivation Path:		
Comments:	Complete Cell Allocation of cell A in HO cases. ARFCN_list={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020408320202090200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_A2d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	Complete Cell Allocation of Cell A in HO cases using 512 format. frequency list, after time for a target cell used in HO_case in cell A for DCS1800. List_ARFCN={734,741,747,754,759,762,766,767,773,775,779,782,791,798,829,832,844}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'896F0A7CC5FC700A8B9F7FF452463340'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B0	
Structured Type:	FRQL	
Derivation Path:		
Comments:	Complete Cell Allocation of cell B in HO cases. all of the CA: ARFCN_list={14,18,22,24,30,31,38,40,60,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F02080000A060A22000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B0d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B using 512 format. List_ARFCN= complete cell allocation of cell B in HO cases. 739,743,746,749,756,758,761,764,771,779,782,791,798,829,832,844	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'0F'O	
fl	'8971883C46FB700A8BDF7DF4324330'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B1	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. List_ARFCN={14,18,22,24,30,31,38,53,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020010002060222000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B1d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, used in TC_26_6_13 for cell B.List_ARFCN = {739,743,746,749,756,758,761,764,771,779,782,791,798,829,832,844} using 256 format.	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'0D'O	
fl	'8A71907137B602C5DEF7A348D8'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B2	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. Complete Cell Allocation of cell B in HO cases. List_ARFCN={14,18,22,24,30,31,38,40,60,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F02080000A060A22000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B3	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. List_ARFCN={14,18,22,24,60,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020800000000A22000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B4	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. List_ARFCN={40, 66, 73, 74, 75, 76, 108, 114}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020000008000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B3d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B using 1024 format. List_ARFCN={749, 758, 761, 764, 771, 779, 782, 791, 798, 829, 832, 844}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'0D'O	
fl	'830EF70BFEB843C3FBF9F1D018'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B4d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. variable bit map format. List_ARFCN={764, 779, 782, 791, 798, 829, 832, 844}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'0D'O	
fl	'8F7E000120102000000480080'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B5	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. List_ARFCN={14,18,22,24,30,31,38,53,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020010002060A22000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B5d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. List_ARFCN={756, 758, 761, 771, 779, 782, 791, 798, 829, 832, 844}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'09'O	
fl	'8D7A23B07B3D7FD4D0'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B6d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. List_ARFCN={761,764,771,779,782,791,798,829,832} range 512 format.	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'0A'O	
fl	'897C87BD09BC61060F90'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B7	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. List_ARFCN={14,18,22,24,30,31,38,60,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020800002060A22000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B72	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. List_ARFCN={14,18,22,24,30,31,38,40,66,73,74,75, 76,108}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0000080000000F02000000A060A22000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B7d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency short list, after time for cell B using 128 format. List_ARFCN={746, 779}	
Element Name	**Element Value**	**Comments**
iei	'00000010'B	
iel	'03'O	
fl	'8D7521'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B8d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B using 1024 format. List_ARFCN= complete cell allocation of cell B in HO cases. 739,743,746,749,756,758,761,764,771,779,782,791,798,829,832,844	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'8303F886FDBC0148BEFDFBF41890633C'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B9d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B using 256 format List_ARFCN={746,749,756,761,764,798,829,832,844}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'09'O	
fl	'8B751A5B45DFA19990'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B10d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B using range 256 format, ARFCN={764,779,782}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'05'O	
fl	'8B7E097D00'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_11d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B using variable bit map format for List_ARFCN={739,743,746,749,756,758,764,771,779,782,791,798,829,832,844}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'0F'O	
fl	'8F71892050408090081000000024004 0'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B12d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency short list, after time for cell B using 128 format List_ARFCN={758,761,771}	
Element Name	**Element Value**	**Comments**
iei	'00000010'B	
iel	'04'O	
fl	'8D7B0DD8'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_A0E	
Structured Type:	FRQL	
Derivation Path:		
Comments:	Complete Cell Allocation of cell A in EGSM cases using of 1024 format. ARFCN_list={0, 30, 40, 66, 1005, 1010, 1015}.	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'841EEA893EF981438000000000000000 0'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_21	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 61.. 124 for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0FFFFFFFFFFFFFFFF000000000000000'O	
Detailed Comments:	B_ARFCN = 62, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_22	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 1, 3, 5, ...,123, 2, 4 for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'05555555555555555555555555555555F'O	
Detailed Comments:	B_ARFCN = 5, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_23	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs= 2, 4, 6, ..., 124, 1, 3 for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0AAAAAAAAAAAAAAAAAAAAAAAAAAAAF'O	
Detailed Comments:	B_ARFCN = 8, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_24	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 21, ..., 84, for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'00000000000FFFFFFFFFFFFFFF00000'O	
Detailed Comments:	B_ARFCN = 25, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_25	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 81, ..., 124, 1, ..., 20, for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0FFFFFFFFFFF000000000000000FFFF'O	
Detailed Comments:	B_ARFCN = 86, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_26	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 11, ..., 74, for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'00000000000003FFFFFFFFFFFFFFC00'O	
Detailed Comments:	B_ARFCN = 18, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_27	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 31, ..., 94, for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'000000003FFFFFFFFFFFFFFC000000'O	
Detailed Comments:	B_ARFCN = 38, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_28	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 51, ...,114, for GSM of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0003FFFFFFFFFFFFFC000000000000'O	
Detailed Comments:	B_ARFCN = 59, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_29	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 749, ..., 812, for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'8F5E800000000000FFFFFFFFFFFFFFF'O	
Detailed Comments:	Org_ARFCN = 701, B_ARFCN = 805, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_30	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 749, ..., 812, for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'8F5EFFFFFFFFFFFFFFF800000000000'O	
Detailed Comments:	Org_ARFCN = 701, B_ARFCN = 706, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_31	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 702, 704, ..., 812, 733, 741, 749, 757, 765, 773, 781, 789, for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'8F5ED5555555D5D5D5D5D5D5D5D55555'O	
Detailed Comments:	Org_ARFCN = 701, B_ARFCN = 704, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_32	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 717, ..., 724, 733, ..., 744, 757, ..., 760, 773, ..., 812, for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'8F5E8000FF00FFF000F000FFFFFFFFF'O	
Detailed Comments:	Org_ARFCN = 701, B_ARFCN = 760, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_33	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs: 702, ..., 732, 749, 750, 754, ..., 756, 758, 762, ..., 764, 766, 770, ..., 772, 774, 778, ..., 780, 782, 786, ..., 788, 790, 794, ..., 796, 798, 802, ..., 804, 806, 810, ..., 812 for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'8F5EFFFFFFFF00044747474747474747'O	
Detailed Comments:	Org_ARFCN = 701, B_ARFCN = 731, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_34	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 707, 709, ..., 713, 715, 717, ..., 721, 723, 725, ..., 729, 731, 733, ..., 737, 739, 741, ..., 745, 747, 749, ..., 753, 755, 757, ..., 761, 763, 765, ..., 769,771, 779, 787, 795, 798,..., 803, 806, ..., 811, for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'8F5E82FAFAFAFAFAFAFAFA0202027E7E'O	
Detailed Comments:	Org_ARFCN = 701, B_ARFCN = 795, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_35	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 705, ..., 736, 749, 751, 753, 755, 757, 759, 761, 763, 765, 767, 769, 771, 773, 775, 777, 779, 781, 783, 785, 787, 789, 791, 793, 795, 797, 799, 801, 803, 805, 807, 809, 811 , for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'8F5E8FFFFFFFF000AAAAAAAAAAAAAAAAA'O	
Detailed Comments:	Org_ARFCN = 701, B_ARFCN = 735, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_36	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 702, 704, 706, 708, 710, 712, 714, 716, 718, 720, 722, 724, 726, 728, 730, 732, 734, 76, 738, 740, 742, 744, 746, 748, 750, 852, 754, 756, 758, 760, 762, 764, ..., 780, 789, ..., 796, 805, 812, for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'8F5ED555555555555555FFFF00FF00FF'O	
Detailed Comments:	Org_ARFCN = 701, B_ARFCN = 780, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_37	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCNs = 717, ..., 748, 765, ..., 796 , for DCS of TC_26_6_6_1	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'8F5E8000FFFFFFFF0000FFFFFFFF0000'O	
Detailed Comments:	Org_ARFCN = 701, B_ARFCN = 745, not in MA(1)	

Structured Type Constraint		
Constraint Name:	Frql_38	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCN 30 and 50	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'00000000000000000002000020000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_38d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	ARFCN 650 and 750, variable bit map	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'8F450000000000000000000000001000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Hlcmp_AnyOrOmit	
Structured Type:	HLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE with parameter "any"' or "omit".	
Element Name	**Element Value**	**Comments**
iei	'01111101'B IF_PRESENT	
iel	*	
extb3	*	
cs	*	
in	*	
pmpp	*	
extb4	*	
hlci	*	
extb4a	*	
ehlci	*	
Detailed Comments:	used in test cases where the HLC is not checked	

Structured Type Constraint		
Constraint Name:	Hlcmp_NotApplicable	
Structured Type:	HLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE wich is not applicable for a specific service (empty IE).	
Element Name	**Element Value**	**Comments**
iei	'01111101'B	
iel	'00'O	
extb3	-	
cs	-	
in	-	
pmpp	-	
extb4	-	
hlci	-	
extb4a	-	
ehlci	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Hlcmp_TS61	
Structured Type:	HLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE for teleservice 61.	
Element Name	**Element Value**	**Comments**
iei	'01111101'B	
iel	'02'O	
extb3	'1'B	
cs	'00'B	
in	'100'B	
pmpp	'01'B	
extb4	'?'B	
hlci	'0000100'B	
extb4a	*	
ehlci	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Hlcmp_TS62	
Structured Type:	HLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE for teleservice 62.	
Element Name	**Element Value**	**Comments**
iei	'01111101'B	
iel	'02'O	
extb3	'1'B	
cs	'00'B	
in	'100'B	
pmpp	'01'B	
extb4	'?'B	
hlci	'0000100'B	
extb4a	*	
ehlci	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct(iel:LENGTH)	
Structured Type:	IARESTOCT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iaroct1	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O,(11 - OC_OctToInt(iel)))	
iaroct2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct_01	
Structured Type:	IARESTOCT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iaroct1	'2B2B2B2B2B2B2B2B2B2B2B'O	
iaroct2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct_02	
Structured Type:	IARESTOCT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iaroct1	'2B2B2B2B2B2B2B2B2B2B2B'O	
iaroct2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct_03	
Structured Type:	IARESTOCT	
Derivation Path:		
Comments:	used in TC_26_6_13_9.	
Element Name	**Element Value**	**Comments**
iei	-	
iaroct1	-	
iaroct2	IaRestOct2_01	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct_04	
Structured Type:	IARESTOCT	
Derivation Path:		
Comments:	used in TC_26_6_13_10.	
Element Name	**Element Value**	**Comments**
iei	-	
iaroct1	-	
iaroct2	IaRestOct2_02	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct_05	
Structured Type:	IARESTOCT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iaroct1	'1A4C6B8EAF37B21A2D5B65'O	
iaroct2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct_06	
Structured Type:	IARESTOCT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iaroct1	'2B2B2B'O	
iaroct2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct_08	
Structured Type:	IARESTOCT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iaroct1	'2B2B2B2B2B2B2B2B'O	
iaroct2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct2_01	
Structured Type:	IARESTOCT2	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
p	'10'B	
frqparalen	'000100'B	
spbt1	'00'B	
maio	INT_TO_BIT(TSPX_Maio29, 6)	
ma	TSPX_Ma29	
spbt2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct2_02	
Structured Type:	IARESTOCT2	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
p	'10'B	
frqparalen	'000100'B	
spbt1	'00'B	
maio	INT_TO_BIT(TSPX_Maio31, 6)	
ma	TSPX_Ma31	
spbt2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	KeyPad_01(character:IA5String)	
Structured Type:	KPF	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'00101100'B	
extb	'0'B	
kpf_info	character	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	KeyPad_02	
Structured Type:	KPF	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'00101100'B	
extb	'0'B	
kpf_info	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Llcmp_AnyOrOmit	
Structured Type:	LLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE which has any or omit value in all fields.	
Element Name	**Element Value**	**Comments**
iei	'01111100'B IF_PRESENT	
iel	*	
extb3	*	
cs	*	
itc	*	
extb3a	*	
negind	*	
spb3a	*	
extb4	*	
tm	*	
itr	*	
extb4_1	*	
rateml	*	
extb5	*	
l1id	*	
uil1	*	
extb5a	*	
sb	*	
neg	*	
ur	*	
extb5b1	*	
ir	*	
nictx	*	
nicrx	*	
fctx	*	
fcrx	*	
spb5b1	*	
extb5b2	*	
hdrb	*	
mfs	*	
mode	*	
llineg	*	
ass	*	
ibob	*	
spb5b2	*	
extb5c	*	
nsb	*	
ndb	*	
pi	*	
extb5d	*	
dplxm	*	
modemt	*	
extb6	*	
l2id	*	
uil2	*	
extb6a1	*	
modl2	*	
spb6a1	*	
q933	*	
extb6a2	*	
usl2pi	*	
extb6b	*	
ws	*	
extb7	*	
l3id	*	
uil3	*	
extb7a1	*	
modl3	*	

spb7a1	*	
extb7a2	*	
usl3pi	*	
extb7b	*	
spb7b	*	
dps	*	
extb7c	*	
pws	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Llcmp_NotApplicable	
Structured Type:	LLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE which is not applicable for a specific service (empty IE).	
Element Name	**Element Value**	**Comments**
iei	'01111100'B	
iel	'00'O	
extb3	-	
cs	-	
itc	-	
extb3a	-	
negind	-	
spb3a	-	
extb4	-	
tm	-	
itr	-	
extb4_1	-	
rateml	-	
extb5	-	
l1id	-	
uil1	-	
extb5a	-	
sb	-	
neg	-	
ur	-	
extb5b1	-	
ir	-	
nictx	-	
nicrx	-	
fctx	-	
fcrx	-	
spb5b1	-	
extb5b2	-	
hdrb	-	
mfs	-	
mode	-	
llineg	-	
ass	-	
ibob	-	
spb5b2	-	
extb5c	-	
nsb	-	
ndb	-	
pi	-	
extb5d	-	
dplxm	-	
modemt	-	
extb6	-	
l2id	-	
uil2	-	
extb6a1	-	
modl2	-	
spb6a1	-	
q933	-	
extb6a2	-	
usl2pi	-	
extb6b	-	
ws	-	
extb7	-	
l3id	-	
uil3	-	
extb7a1	-	
modl3	-	

spb7a1	-	
extb7a2	-	
usl3pi	-	
extb7b	-	
spb7b	-	
dps	-	
extb7c	-	
pws	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Llcmp_BS2xOrBS4x_UDI	
Structured Type:	LLCMP	
Derivation Path:		
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.2.1 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iei	'01111100'B	
iel	('06'O, '07'O, '08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'01000'B	
extb3a	'1'B IF_PRESENT	
negind	*	
spb3a	'000000'B IF_PRESENT	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb4_1	-	
rateml	-	
extb5	'0'B	
l1id	'01'B	
uil1	'00001'B	
extb5a	'0'B	
sb	'1'B	
neg	'0'B	
ur	('00010'B, '00011'B, '00101'B, '01000'B, '10111'B, '11110'B)	
extb5b1	'0'B	
ir	('01'B, '10'B)	
nictx	'?'B	
nicrx	'?'B	
fctx	'0'B	
fcrx	'0'B	
spb5b1	'0'B	
extb5b2	-	
hdrb	-	
mfs	-	
mode	-	
llineg	-	
ass	-	
ibob	-	
spb5b2	-	
extb5c	'?'B	
nsb	('01'B, '11'B)	
ndb	('10'B, '11'B)	
pi	('000'B,'010'B, '011'B, '100'B, '101'B)	
extb5d	'1'B IF_PRESENT	
dplxm	*	
modemt	*	
extb6	*	
l2id	*	
uil2	*	
extb6a1	*	
modl2	*	
spb6a1	*	
q933	*	
extb6a2	*	
usl2pi	*	
extb6b	*	
ws	*	
extb7	*	
l3id	*	
uil3	*	

extb7a1	*	
modl3	*	
spb7a1	*	
extb7a2	*	
usl3pi	*	
extb7b	*	
spb7b	*	
dps	*	
extb7c	*	
pws	*	
Detailed Comments:	Corresponding to the Bearer Services 21 ... 26, 41 ... 46 Unrestrict digital information transfer capability.	

Structured Type Constraint		
Constraint Name:	Llcmp_BS2xOrBS61aOrBS81a_3100	
Structured Type:	LLCMP	
Derivation Path:	Llcmp_BS2xOrBS4x_UDI.	
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.2.2 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iel	('07'O,'08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'10000'B	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb5	'0'B	
l1id	'01'B	
uil1	'00011'B	
extb5a	'0'B	
sb	?	
neg	?	
ur	?	
extb5b1	'0'B	
ir	?	
nictx	?	
nicrx	?	
fctx	?	
fcrx	?	
spb5b1	'0'B	
extb5c	'0'B	
nsb	?	
ndb	?	
pi	?	
extb5d	'1'B	
dplxm	?	
modemt	('010001'B, '010010'B, '010011'B, '010100'B, '010111'B, '011100'B)	
Detailed Comments:	Corresponding to the Bearer Services 21 ... 26, 61 Asynchronous or 81 Asynchronous 3.1 kHz audio ex-PLMN information transfer capability.	

Structured Type Constraint		
Constraint Name:	Llcmp_BS3x_UDI	
Structured Type:	LLCMP	
Derivation Path:	Llcmp_BS2xOrBS4x_UDI.	
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.3.1 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iel	('05'O, '06'O, '07'O, '08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'01000'B	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb5	'0'B	
l1id	'01'B	
uil1	('00001'B, '01001'B)	
extb5a	'0'B	
sb	'0'B	
neg	'0'B	
ur	('00010'B, '00011'B, '00101'B, '01000'B)	
extb5b1	'?'B	
ir	('01'B, '10'B)	
nictx	'?'B	
nicrx	'?'B	
fctx	'?'B	
fcrx	'?'B	
spb5b1	'0'B	
extb5c	'0'B IF_PRESENT	
nsb	*	
ndb	*	
pi	*	
extb5d	'1'B IF_PRESENT	
dplxm	*	
modemt	*	
extb6	*	
l2id	'10'B IF_PRESENT	
uil2	('00110'B, '00111'B) IF_PRESENT	
extb7	*	
l3id	'11'B IF_PRESENT	
uil3	('00110'B, '00111'B) IF_PRESENT	
Detailed Comments:	Corresponding to the Bearer Services 31 ... 34 unrestricted digital information transfer capability.	

Structured Type Constraint		
Constraint Name:	Llcmp_BS3xOrBS61sOrBS81s_3100	
Structured Type:	LLCMP	
Derivation Path:	Llcmp_BS2xOrBS4x_UDI.	
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.3.2 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iel	('03'O,'04'O, '08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'10000'B	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb5	'?'B	
l1id	'01'B	
uil1	'00011'B	
extb5a	'0'B IF_PRESENT	
sb	*	
neg	*	
ur	*	
extb5b1	*	
ir	*	
nictx	*	
nicrx	*	
fctx	*	
fcrx	*	
spb5b1	*	
extb5c	'0'B IF_PRESENT	
nsb	*	
ndb	*	
pi	*	
extb5d	'1'B IF_PRESENT	
dplxm	*	
modemt	('010010'B, '010011'B, '010111'B, '011100'B) IF_PRESENT	
extb6	*	
l2id	'10'B IF_PRESENT	
uil2	('00110'B, '00111'B) IF_PRESENT	
extb7	*	
l3id	'11'B IF_PRESENT	
uil3	('00110'B, '00111'B) IF_PRESENT	
Detailed Comments:	Corresponding to the Bearer Services 31 ... 34, 61 Synchronous or 81 Synchronous 3.1kHz audio information transfer capability.	

Structured Type Constraint		
Constraint Name:	Llcmp_BS5x	
Structured Type:	LLCMP	
Derivation Path:	Llcmp_BS2xOrBS4x_UDI.	
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.5.1 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iel	('06'O,'07'O, '08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'01000'B	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb5	'0'B	
l1id	'01'B	
uil1	'01001'B	
extb5a	'?'B	
sb	'0'B	
neg	'0'B	
ur	('00011'B, '00101'B, '01000'B)	
extb5b1	*	
ir	*	
nictx	*	
nicrx	*	
fctx	*	
fcrx	*	
spb5b1	*	
extb5c	*	
nsb	*	
ndb	*	
pi	*	
extb5d	*	
dplxm	*	
modemt	*	
extb6	?	
l2id	'10'B	
uil2	('00110'B, '00111'B)	
extb7	?	
l3id	'11'B	
uil3	('00110'B, '00111'B)	
Detailed Comments:	Corresponding to the Bearer Services 51 ... 53 unrestricted digital information transfer capability.	

Structured Type Constraint		
Constraint Name:	LocAreaId_anyomit	
Structured Type:	LAI	
Derivation Path:		
Comments:	Omittted constraint	
Element Name	**Element Value**	**Comments**
iei	*	
mcc	*	
mnc	*	
lac	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	LocAreaId(mcc, mnc, lac:OCTETSTRING)	
Structured Type:	LAI	
Derivation Path:		
Comments:	Cell A default value for L3 test	
Element Name	**Element Value**	**Comments**
iei	-	
mcc	mcc	
mnc	mnc	
lac	lac	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	LocAreaIdiei(mcc, mnc, lac:OCTETSTRING)	
Structured Type:	LAI	
Derivation Path:		
Comments:	Cell A default value for L3 test	
Element Name	**Element Value**	**Comments**
iei	'00010011'B	
mcc	mcc	
mnc	mnc	
lac	lac	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	LocUpType(locup:B_2)	
Structured Type:	LUT	
Derivation Path:		
Comments:	location updating type	
Element Name	**Element Value**	**Comments**
foreq	'?'B	
sprb	'0'B	
lut	locup	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma1_g01	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 64 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'08'O	
mac_8n	'00000000'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00000000'B	
mac_2n	'00111111'B	
mac_1n	'11111110'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma1_g02	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 64 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'08'O	
mac_8n	'10000000'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	'10000000'B	
mac_4n	'00100000'B	
mac_3n	'00000000'B	
mac_2n	'00000000'B	
mac_1n	'00000101'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma1_g03	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 64 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'08'O	
mac_8n	'01000000'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	'00000010'B	
mac_4n	'00000000'B	
mac_3n	'00000000'B	
mac_2n	'00000000'B	
mac_1n	'01100101'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma1_g04	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 64 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'08'O	
mac_8n	'00000001'B	
mac_7n	'00000000'B	
mac_6n	'01000010'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00000001'B	
mac_2n	'01110000'B	
mac_1n	'00000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma1_g05	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 64 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'08'O	
mac_8n	'01001000'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00001101'B	
mac_2n	'01000100'B	
mac_1n	'00000001'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma1_g06	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 64 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'08'O	
mac_8n	'11111111'B	
mac_7n	'10000000'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00000000'B	
mac_2n	'00011111'B	
mac_1n	'00000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma1_g07	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 64 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'08'O	
mac_8n	'11111000'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00000000'B	
mac_2n	'00011111'B	
mac_1n	'00000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma1_g08	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 64 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'08'O	
mac_8n	'00000000'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00001111'B	
mac_2n	'11111110'B	
mac_1n	'00000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma1_g09	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 64 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'08'O	
mac_8n	'11111111'B	
mac_7n	'11110000'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	'00000000'B	
mac_3n	'00000000'B	
mac_2n	'00000000'B	
mac_1n	'00000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma2_g01	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 30 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'04'O	
mac_8n	'00111111'B	
mac_7n	'11111100'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma2_g02	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 30 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'04'O	
mac_8n	'00000000'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	'00001111'B	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma2_g03	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 30 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'04'O	
mac_8n	'00000000'B	
mac_7n	'00011111'B	
mac_6n	'00000000'B	
mac_5n	'00000000'B	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma2_g04	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 30 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'04'O	
mac_8n	'00000000'B	
mac_7n	'00001010'B	
mac_6n	'10101010'B	
mac_5n	'00000000'B	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma2_g05	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 30 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'04'O	
mac_8n	'00010101'B	
mac_7n	'01000000'B	
mac_6n	'00000000'B	
mac_5n	'10101000'B	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma2_g06	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 30 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'04'O	
mac_8n	'00011010'B	
mac_7n	'10100000'B	
mac_6n	'00000000'B	
mac_5n	'00011100'B	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma2_g07	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 30 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'04'O	
mac_8n	'00011100'B	
mac_7n	'00000011'B	
mac_6n	'10011100'B	
mac_5n	'00000000'B	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma2_g08	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 30 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'04'O	
mac_8n	'00000001'B	
mac_7n	'00100100'B	
mac_6n	'10010000'B	
mac_5n	'00011111'B	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma2_g09	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 30 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'04'O	
mac_8n	'00110000'B	
mac_7n	'00000011'B	
mac_6n	'11010000'B	
mac_5n	'00001111'B	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma3_g01	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 12 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	'00001110'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma3_g02	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 12 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	'00000100'B	
mac_7n	'00000011'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma3_g03	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 12 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	'00001010'B	
mac_7n	'10100000'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma3_g04	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 12 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	'00000111'B	
mac_7n	'11000000'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma3_g05	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 12 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	'00001010'B	
mac_7n	'10100101'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma3_g06	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 12 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	'00001111'B	
mac_7n	'00001011'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma3_g07	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 12 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	'00001010'B	
mac_7n	'11110011'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma3_g08	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 12 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	'00001111'B	
mac_7n	'00101111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Ma3_g09	
Structured Type:	MA	
Derivation Path:		
Comments:	Used in TC_26_6_6_1 for GSM and DCS, cell allocation contains 12 ARFCHs	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	'00001111'B	
mac_7n	'11101110'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Mi_01	
Structured Type:	MI	
Derivation Path:		
Comments:	used as a comprehension required IE	
Element Name	**Element Value**	**Comments**
iei	'00000000'B	
iel	'01'O	
idigit_1	'1110'B	
oei	'1'B	
toi	'111'B	
idigits_other	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Mi_02	
Structured Type:	MI	
Derivation Path:		
Comments:	used as an unknown IEI.	
Element Name	**Element Value**	**Comments**
iei	'10101110'B	
iel	-	
idigit_1	-	
oei	-	
toi	-	
idigits_other	-	
Detailed Comments:	used in TC_26_5_6_1_1.	

Structured Type Constraint		
Constraint Name:	Mi_05	
Structured Type:	MI	
Derivation Path:		
Comments:	used as an unknown IE	
Element Name	**Element Value**	**Comments**
iei	'00000010'B	
iel	'E0'O	
idigit_1	'1001'B	
oei	'0'B	
toi	'000'B	
idigits_other	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Mi_06	
Structured Type:	MI	
Derivation Path:		
Comments:	used as an unknown IEI.	
Element Name	**Element Value**	**Comments**
iei	'00010011'B	
iel	'02'O	
idigit_1	'1010'B	
oei	'0'B	
toi	'111'B	
idigits_other	'78'O	
Detailed Comments:	used in TC_26_5_6_1_2.	

Structured Type Constraint		
Constraint Name:	MiImei_01	
Structured Type:	MI	
Derivation Path:		
Comments:	IMEI	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'08'O	
idigit_1	OC_FirstDigi(TSPX_IMEI)	
oei	'1'B	
toi	'010'B	
idigits_other	OC_OtherDigi(TSPX_IMEI)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiImeisv_01	
Structured Type:	MI	
Derivation Path:		
Comments:	TSPX_IMEISV	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'09'O	
idigit_1	OC_FirstDigi(TSPX_IMEISV)	
oei	'0'B	
toi	'011'B	
idigits_other	OC_OtherDigi(TSPX_IMEISV)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiImeisv_01iei	
Structured Type:	MI	
Derivation Path:		
Comments:	TSPX_IMEISV	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	'09'O	
idigit_1	OC_FirstDigi(TSPX_IMEISV)	
oei	'0'B	
toi	'011'B	
idigits_other	OC_OtherDigi(TSPX_IMEISV)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiImsi_01	
Structured Type:	MI	
Derivation Path:		
Comments:	the TSPX_IMSI is the IMSI of the MS under test	
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct(((LENGTH_OF(TSPX_IMSI))/2 + 1),1)	
idigit_1	OC_FirstDigi(TSPX_IMSI)	
oei	OC_OeBit(TSPX_IMSI)	
toi	'001'B	
idigits_other	OC_OtherDigi(TSPX_IMSI)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiImsi_01iei	
Structured Type:	MI	
Derivation Path:		
Comments:	the TSPX_IMSI is the IMSI of the MS under test	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	OC_IntToOct(((LENGTH_OF(TSPX_IMSI))/2 + 1),1)	
idigit_1	OC_FirstDigi(TSPX_IMSI)	
oei	OC_OeBit(TSPX_IMSI)	
toi	'001'B	
idigits_other	OC_OtherDigi(TSPX_IMSI)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiImsi_02	
Structured Type:	MI	
Derivation Path:		
Comments:	This constraint is used in paging filling message. No mobile is paged (Type of identity is set to '000'B, i.e. no identity).	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'01'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'000'B	
idigits_other	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiImsi_31	
Structured Type:	MI	
Derivation Path:		
Comments:	An another IMSI differing from MiImsi_01for RR testing, which is shorter than the maximum length.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'05'O	
idigit_1	'0000'B	
oei	'1'B	
toi	'001'B	
idigits_other	'10102143'O	
Detailed Comments:	IMSI has 9 digits: '00 10 11 23 4'	

Structured Type Constraint		
Constraint Name:	MiImsi_31iei	
Structured Type:	MI	
Derivation Path:		
Comments:	An another IMSI differing from MiImsi_01for RR testing, which is shorter than the maximum length.	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	'05'O	
idigit_1	'0000'B	
oei	'1'B	
toi	'001'B	
idigits_other	'10102143'O	
Detailed Comments:	IMSI has 9 digits: '00 10 11 23 4'. The iei appears in the messages, e.g. location updating accept.	

Structured Type Constraint		
Constraint Name:	MiImsi_r01	
Structured Type:	MI	
Derivation Path:		
Comments:	An another IMSI differing from MiImsi_01 for RR testing.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'07'O	
idigit_1	'0000'B	
oei	'1'B	
toi	'001'B	
idigits_other	'981032547698'O	
Detailed Comments:	IMSI has 13 digits: '08 90 12 34 56 78 9'	

Structured Type Constraint		
Constraint Name:	MiImsi_r01iei	
Structured Type:	MI	
Derivation Path:		
Comments:	An another IMSI differing from MiImsi_01 for RR testing.	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	'07'O	
idigit_1	'0000'B	
oei	'1'B	
toi	'001'B	
idigits_other	'981032547698'O	
Detailed Comments:	IMSI has 13 digits: '08 90 12 34 56 78 9'	

Structured Type Constraint		
Constraint Name:	MiMsi_omit	
Structured Type:	MI	
Derivation Path:		
Comments:	omit MSI for testing.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
idigit_1	-	
oei	-	
toi	-	
idigits_other	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_01	
Structured Type:	MI	
Derivation Path:		
Comments:	default TMSI for testing.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'100'B	
idigits_other	TSPX_TMSI	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_01iei	
Structured Type:	MI	
Derivation Path:		
Comments:	default TMSI for testing.	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'100'B	
idigits_other	TSPX_TMSI	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_02	
Structured Type:	MI	
Derivation Path:		
Comments:	A new TMSI	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'100'B	
idigits_other	TSPX_TMSI1	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_02iei	
Structured Type:	MI	
Derivation Path:		
Comments:	A new TMSI	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'100'B	
idigits_other	TSPX_TMSI1	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_r01	
Structured Type:	MI	
Derivation Path:		
Comments:	A TMSI differs from MiTmsi_01	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'100'B	
idigits_other	OC_IncTmsi(TSPX_TMSI, '01'O)	
Detailed Comments:	Note: the TSPX_TMSI + '01'O shall not be identical to TSPX_TMSI1	

Structured Type Constraint		
Constraint Name:	MiTmsi_r02	
Structured Type:	MI	
Derivation Path:		
Comments:	default TMSI for testing with toi setting to no id.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'000'B	
idigits_other	TSPX_TMSI	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_r02iei	
Structured Type:	MI	
Derivation Path:		
Comments:	default TMSI for testing with toi setting to no id.	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'000'B	
idigits_other	TSPX_TMSI	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_01	
Structured Type:	MA	
Derivation Path:		
Comments:	non hopping	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'00'O	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_02	
Structured Type:	MA	
Derivation Path:	MoblAllc_01.	
Comments:	hopping	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'01'O	
mac_8n	'00000010'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_04	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping used in TC_26_6_4_2_2.	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'01'O	
mac_8n	'00001110'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_A0	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with 0001 1111 1111 1111 1111 indicates all of the frequencies in CA of cell A with ARFCN_list={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	'11111111'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_A1	
Structured Type:	MA	
Derivation Path:		
Comments:	indicates all of the frequencies in CA of cell A with ARFCN_list={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114} except for the BCCH frequency 20.	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	'11111011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_A3d	
Structured Type:	MA	
Derivation Path:		
Comments:	indicates all of the frequencies in CA of cell A ARFCN_list={734,741,747,754,759,762,766,767,773,775,779,782,791,798,829,832,844} except 747 and 767.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	'01111011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_252	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with 0001 1100 indicates ARFCN_list={40,66,73,74,75,76,108,114} to the ARFCN_list={73,74,75}.	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'01'O	
mac_8n	'00011100'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_252d	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with 0111 0000 0 List_ARFCN={761,764,771,779,782,791,798,829,832} to the ARFCN_list={791,798,829}.	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'02'O	
mac_8n	'00000000'B	
mac_7n	'11100000'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_22	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with all of f's from cell allocation of cell B with except of {764,832,844}	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'02'O	
mac_8n	'00111111'B	
mac_7n	'01111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_281	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with indicates to the ARFCN_list={73,74,75} of cell allocation A.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000000'B	
mac_7n	'00111000'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_281d	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with indicates to the ARFCN_list={773,775,779} of cell allocation A.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000000'B	
mac_7n	'00000111'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_281e	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping sequence with EGSM frequencies. The frequencies which are used in the mobile hopping sequence are {0, 80, 1005, 1010} .	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'01'O	
mac_8n	'10111000'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_281e2	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with indicates to the ARFCN_list={20,66,78} of E-GSM cell allocation with ARFCN_List={20,40,66,73,74,75,76,77,78,79,108,114}	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	'00000001'B	
mac_7n	'00000101'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_282	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with indicates to the Complete Cell Allocation of cell A with except for BCCH-f	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	'11111011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_B1	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with 1111 1111 1111 1111 ARFCN_list={14,18,22,24,30,31,38,40,60,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	'11111111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_B1iei	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with ARFCN_list={14,18,22,31,40}.	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'01'O	
mac_8n	'10100111'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_Be1(par_ma: BITSTRING)	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with par_ma	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'01'O	
mac_8n	par_ma	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_Be1iei(par_ma: BITSTRING)	
Structured Type:	MA	
Derivation Path:	MoblAllc_20_Be1.	
Comments:	hopping with par_ma	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_Be2(par_ma1: BITSTRING; par_ma2: BITSTRING)	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with par_ma1 and par_ma2	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	par_ma1	
mac_7n	par_ma2	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_Be2iei(par_ma1: BITSTRING; par_ma2: BITSTRING)	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with par_ma1 and par_ma2	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'02'O	
mac_8n	par_ma1	
mac_7n	par_ma2	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_A2	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with default Cell Allocation of HO cases and only one frequency in hopping	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_A2iei	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with 1000 0000 0000 0000 indicates from default Cell Allocation of HO cases with the ARFCN_List={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114} the ARFCN_list={114}	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_A3	
Structured Type:	MA	
Derivation Path:		
Comments:	indicates all of the frequencies in CA of cell A ARFCN_list={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114} except 20 and 52.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111110'B	
mac_6n	'11111011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_omit	
Structured Type:	MA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r01	
Structured Type:	MA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'02'O	
mac_8n	'00000011'B	
mac_7n	'11111011'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r02	
Structured Type:	MA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'02'O	
mac_8n	'00000111'B	
mac_7n	'11100011'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r03	
Structured Type:	MA	
Derivation Path:	MoblAllc_r01.	
Comments:		
Element Name	**Element Value**	**Comments**
iel	'01'O	
mac_8n	'00000001'B	
mac_7n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r04	
Structured Type:	MA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'02'O	
mac_8n	'00000001'B	
mac_7n	'11000111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r05	
Structured Type:	MA	
Derivation Path:		
Comments:	used in TC_26_6_13_1 for immediate assignment command	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma1) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma1) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma1) MOD 256), 8)	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r06	
Structured Type:	MA	
Derivation Path:		
Comments:	used in TC_26_6_13_1 after time	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'03'O	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma2) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma2) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma2) MOD 256), 8)	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r07	
Structured Type:	MA	
Derivation Path:		
Comments:	used in TC_26_6_13_1 and TC_26_6_13_2 before time	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'03'O	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma3) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma3) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma3) MOD 256), 8)	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r08	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_2 for immediate asignment command	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma4) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma4) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma4) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r09	
Structured Type:	MA	
Derivation Path:		
Comments:	used in TC_26_6_13_2 after time	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'03'O	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma5) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma5) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma5) MOD 256), 8)	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r10	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_3 for immediate assignment	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma6) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma6) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma6) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r11	
Structured Type:	MA	
Derivation Path:		
Comments:	used in TC_26_6_13_3 for frequency redifinition	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma7) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma7) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma7) MOD 256), 8)	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r12	
Structured Type:	MA	
Derivation Path:	MoblAllc_r11.	
Comments:	used in TC_26_6_13_3 for assignment command after time	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma8) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma8) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma8) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r13	
Structured Type:	MA	
Derivation Path:		
Comments:	used in TC_26_6_13_3 for assignment command before time	
Element Name	**Element Value**	**Comments**
iei	'00100001'B	
iel	'03'O	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma9) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma9) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma9) MOD 256), 8)	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r14	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_4 for immediate assignment	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma10) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma10) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma10) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r15	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_4 for frequency redifinition	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma11) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma11) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma11) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r16	
Structured Type:	MA	
Derivation Path:		
Comments:	used in TC_26_6_13_4 for assignment command after time	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'03'O	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma12) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma12) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma12) MOD 256), 8)	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r17	
Structured Type:	MA	
Derivation Path:		
Comments:	used in TC_26_6_13_4 for assignment command before time	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'03'O	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma13) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma13) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma13) MOD 256), 8)	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r18	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_5 for immediate assignment	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma14) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma14) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma14) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r18iei	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.MoblAllc_r18.	
Comments:	used in TC_26_6_13_5 for immediate assignment	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r20	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_5 for handover before time	
Element Name	**Element Value**	**Comments**
iei	'00100001'B	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma16) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma16) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma16) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r21	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_6 for immediate assignment	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma17) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma17) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma17) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r22	
Structured Type:	MA	
Derivation Path:		
Comments:	used in TC_26_6_13_6 for handover after time	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'03'O	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma18) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma18) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma18) MOD 256), 8)	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r23	
Structured Type:	MA	
Derivation Path:		
Comments:	used in TC_26_6_13_6 for handover before time	
Element Name	**Element Value**	**Comments**
iei	'00100001'B	
iel	'03'O	
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma19) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma19) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma19) MOD 256), 8)	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r24	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_7 for immediate assignment	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma20) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma20) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma20) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r25	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_7 for frequency redifinition	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma21) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma21) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma21) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r26	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_7 for HANDOVER COMMAND after time	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma22) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma22) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma22) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r27	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_7 for HANDOVER COMMAND before time	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma23) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma23) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma23) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r28	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_8 for immediate assignment	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma24) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma24) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma24) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r29	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_8 for frequency redifinition	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma25) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma25) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma25) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r30	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_8 for HANDOVER COMMAND after time	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma26) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma26) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma26) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r30iei	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.MoblAllc_r30.	
Comments:	used in TC_26_6_13_8 for HANDOVER COMMAND after time	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r31	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_8 for HANDOVER COMMAND before time	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma27) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma27) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma27) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r31iei	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.MoblAllc_r31.	
Comments:	used in TC_26_6_13_8 for HANDOVER COMMAND before time	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r32	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_9 for immediate assignment, after time	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma28) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma28) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma28) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_r33	
Structured Type:	MA	
Derivation Path:	MoblAllc_r05.	
Comments:	used in TC_26_6_13_10 for immediate assignment	
Element Name	**Element Value**	**Comments**
mac_8n	INT_TO_BIT((OC_OctToInt(TSPX_Ma30) / 65536), 8)	
mac_7n	INT_TO_BIT(((OC_OctToInt(TSPX_Ma30) / 256) MOD 256), 8)	
mac_6n	INT_TO_BIT((OC_OctToInt(TSPX_Ma30) MOD 256), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_01	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing no measurement results.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	('000'B, '111'B)	
rxlev_nc1	'000000'B	
bcchfrq_nc1	'00000'B	
bsic_nc1	'000000'B	
rxlev_nc2	'000000'B	
bcchfrq_nc2	'00000'B	
bsic_nc2	'000000'B	
rxlev_nc3	'000000'B	
bcchfrq_nc3	'00000'B	
bsic_nc3	'000000'B	
rxlev_nc4	'000000'B	
bcchfrq_nc4	'00000'B	
bsic_nc4	'000000'B	
rxlev_nc5	'000000'B	
bcchfrq_nc5	'00000'B	
bsic_nc5	'000000'B	
rxlev_nc6	'000000'B	
bcchfrq_nc6	'00000'B	
bsic_nc6	'000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_02	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE matching any value.	
Element Name	**Element Value**	**Comments**
ba_used	?	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	?	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	?	
rxlev_nc1	*	
bcchfrq_nc1	*	
bsic_nc1	*	
rxlev_nc2	*	
bcchfrq_nc2	*	
bsic_nc2	*	
rxlev_nc3	*	
bcchfrq_nc3	*	
bsic_nc3	*	
rxlev_nc4	*	
bcchfrq_nc4	*	
bsic_nc4	*	
rxlev_nc5	*	
bcchfrq_nc5	*	
bsic_nc5	*	
rxlev_nc6	*	
bcchfrq_nc6	*	
bsic_nc6	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_03	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing 6 measurement results.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'110'B	
rxlev_nc1	?	
bcchfrq_nc1	?	
bsic_nc1	?	
rxlev_nc2	?	
bcchfrq_nc2	?	
bsic_nc2	?	
rxlev_nc3	?	
bcchfrq_nc3	?	
bsic_nc3	?	
rxlev_nc4	?	
bcchfrq_nc4	?	
bsic_nc4	?	
rxlev_nc5	?	
bcchfrq_nc5	?	
bsic_nc5	?	
rxlev_nc6	?	
bcchfrq_nc6	?	
bsic_nc6	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_03e1	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing 6 measurement results for EGSM.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'110'B	
rxlev_nc1	?	
bcchfrq_nc1	'00001'B	
bsic_nc1	'001011'B	
rxlev_nc2	?	
bcchfrq_nc2	'00010'B	
bsic_nc2	'001111'B	
rxlev_nc3	?	
bcchfrq_nc3	'00100'B	
bsic_nc3	'001101'B	
rxlev_nc4	?	
bcchfrq_nc4	'00101'B	
bsic_nc4	'001001'B	
rxlev_nc5	?	
bcchfrq_nc5	'00110'B	
bsic_nc5	'001111'B	
rxlev_nc6	?	
bcchfrq_nc6	'00111'B	
bsic_nc6	'001101'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_03e2	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing 6 measurement results for EGSM.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'110'B	
rxlev_nc1	?	
bcchfrq_nc1	'00010'B	
bsic_nc1	'001011'B	
rxlev_nc2	?	
bcchfrq_nc2	'00011'B	
bsic_nc2	'001111'B	
rxlev_nc3	?	
bcchfrq_nc3	'00100'B	
bsic_nc3	'001101'B	
rxlev_nc4	?	
bcchfrq_nc4	'00101'B	
bsic_nc4	'001001'B	
rxlev_nc5	?	
bcchfrq_nc5	'00110'B	
bsic_nc5	'001111'B	
rxlev_nc6	?	
bcchfrq_nc6	'00001'B	
bsic_nc6	'001100'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_03e3	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing 6 measurement results for EGSM.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'110'B	
rxlev_nc1	?	
bcchfrq_nc1	'00010'B	
bsic_nc1	'001011'B	
rxlev_nc2	?	
bcchfrq_nc2	'00011'B	
bsic_nc2	'001111'B	
rxlev_nc3	?	
bcchfrq_nc3	'00110'B	
bsic_nc3	'001001'B	
rxlev_nc4	?	
bcchfrq_nc4	'01000'B	
bsic_nc4	'001111'B	
rxlev_nc5	?	
bcchfrq_nc5	'01000'B	
bsic_nc5	'001101'B	
rxlev_nc6	?	
bcchfrq_nc6	'00001'B	
bsic_nc6	'001100'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_03e4	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing 6 measurement results for EGSM.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'110'B	
rxlev_nc1	?	
bcchfrq_nc1	'00010'B	
bsic_nc1	'001011'B	
rxlev_nc2	?	
bcchfrq_nc2	'00011'B	
bsic_nc2	'001111'B	
rxlev_nc3	?	
bcchfrq_nc3	'00110'B	
bsic_nc3	'001101'B	
rxlev_nc4	?	
bcchfrq_nc4	'00111'B	
bsic_nc4	'001001'B	
rxlev_nc5	?	
bcchfrq_nc5	'01000'B	
bsic_nc5	'001111'B	
rxlev_nc6	?	
bcchfrq_nc6	'00001'B	
bsic_nc6	'001100'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_03e5	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing 6 measurement results for EGSM.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'110'B	
rxlev_nc1	?	
bcchfrq_nc1	'00010'B	
bsic_nc1	'001011'B	
rxlev_nc2	?	
bcchfrq_nc2	'00011'B	
bsic_nc2	'001111'B	
rxlev_nc3	?	
bcchfrq_nc3	'00101'B	
bsic_nc3	'001101'B	
rxlev_nc4	?	
bcchfrq_nc4	'00110'B	
bsic_nc4	'001001'B	
rxlev_nc5	?	
bcchfrq_nc5	'00111'B	
bsic_nc5	'001111'B	
rxlev_nc6	?	
bcchfrq_nc6	'00001'B	
bsic_nc6	'001100'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_04	
Structured Type:	MSRR	
Derivation Path:	MsrResult_03.	
Comments:	A measurement results IE containing 4 measurement results.	
Element Name	**Element Value**	**Comments**
no_nc	'100'B	
bcchfrq_nc1	('00000'B, '00001'B, '00011'B, '00100'B)	
bsic_nc1	('001011'B, '001111'B, '001001'B)	
bcchfrq_nc2	('00000'B, '00001'B, '00011'B, '00100'B)	
bsic_nc2	('001011'B, '001111'B, '001001'B)	
bcchfrq_nc3	('00000'B, '00001'B, '00011'B, '00100'B)	
bsic_nc3	('001011'B, '001111'B, '001001'B)	
bcchfrq_nc4	('00000'B, '00001'B, '00011'B, '00100'B)	
bsic_nc4	('001011'B, '001111'B, '001001'B)	
rxlev_nc5	'000000'B	
bcchfrq_nc5	'00000'B	
bsic_nc5	'000000'B	
rxlev_nc6	'000000'B	
bcchfrq_nc6	'00000'B	
bsic_nc6	'000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_04e	
Structured Type:	MSRR	
Derivation Path:	MsrResult_03.	
Comments:	A measurement results IE containing 3 measurement results for EGSM	
Element Name	**Element Value**	**Comments**
no_nc	'100'B	
rxlev_nc1	?	
bcchfrq_nc1	'00001'B	
bsic_nc1	'001011'B	
rxlev_nc2	?	
bcchfrq_nc2	'00011'B	
bsic_nc2	'001101'B	
rxlev_nc3	?	
bcchfrq_nc3	'00110'B	
bsic_nc3	'001001'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_05	
Structured Type:	MSRR	
Derivation Path:	MsrResult_03.	
Comments:	A measurement results IE containing 6 measurement results and DTX was used.	
Element Name	**Element Value**	**Comments**
dtx_used	'1'B	
bcchfrq_nc1	('00000'B, '00011'B, '00100'B, '00101'B, '00110'B, '00111'B)	
bsic_nc1	('001011'B, '001101'B, '001111'B, '001001'B)	
bcchfrq_nc2	('00000'B, '00011'B, '00100'B, '00101'B, '00110'B, '00111'B)	
bsic_nc2	('001011'B, '001101'B, '001111'B, '001001'B)	
bcchfrq_nc3	('00000'B, '00011'B, '00100'B, '00101'B, '00110'B, '00111'B)	
bsic_nc3	('001011'B, '001101'B, '001111'B, '001001'B)	
bcchfrq_nc4	('00001'B, '00100'B, '00101'B, '00110'B, '00111'B, '01000'B)	
bsic_nc4	('001011'B, '001101'B, '001111'B, '001001'B)	
bcchfrq_nc5	('00001'B, '00100'B, '00101'B, '00110'B, '00111'B, '01000'B)	
bsic_nc5	('001011'B, '001101'B, '001111'B, '001001'B)	
bcchfrq_nc6	('00001'B, '00100'B, '00101'B, '00110'B, '00111'B, '01000'B)	
bsic_nc6	('001011'B, '001101'B, '001111'B, '001001'B)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_06	
Structured Type:	MSRR	
Derivation Path:	MsrResult_03.MsrResult_05.	
Comments:	A measurement results IE containing 6 measurement results and DTX is not checked.	
Element Name	**Element Value**	**Comments**
dtx_used	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_07	
Structured Type:	MSRR	
Derivation Path:	MsrResult_03.	
Comments:	A measurement results IE containing 2 measurement results.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'010'B	
rxlev_nc1	?	
bcchfrq_nc1	('00000'B, '00010'B)	
bsic_nc1	('001011'B, '001101'B)	
rxlev_nc2	?	
bcchfrq_nc2	('00000'B, '00010'B)	
bsic_nc2	('001011'B, '001101'B)	
rxlev_nc3	'000000'B	
bcchfrq_nc3	'00000'B	
bsic_nc3	'000000'B	
rxlev_nc4	'000000'B	
bcchfrq_nc4	'00000'B	
bsic_nc4	'000000'B	
rxlev_nc5	'000000'B	
bcchfrq_nc5	'00000'B	
bsic_nc5	'000000'B	
rxlev_nc6	'000000'B	
bcchfrq_nc6	'00000'B	
bsic_nc6	'000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Pcmd_19(powerlevel:BITSTRING)	
Structured Type:	PCMD	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
sprb	'000'B	
pl	powerlevel	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Pcmd_20(powerlevel:B_5)	
Structured Type:	PCMD	
Derivation Path:		
Comments:	PowerCmd used in synchronized and non synchronized HO cases.	
Element Name	**Element Value**	**Comments**
sprb	'011'B	
pl	powerlevel	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	PiSi_01	
Structured Type:	PI_SI	
Derivation Path:		
Comments:	calling party BCD number with arbitrary spare bits	
Element Name	**Element Value**	**Comments**
extb	'1'B	
pi	'00'B	
sp3b	'110'B	
si	'00'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Pm(pgm:B_2)	
Structured Type:	PM	
Derivation Path:		
Comments:	normal paging mode	
Element Name	**Element Value**	**Comments**
sprb	'00'B	
pgm	pgm	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Pm_02	
Structured Type:	PM	
Derivation Path:		
Comments:	normal paging mode with arbitrary spare bits.	
Element Name	**Element Value**	**Comments**
sprb	'11'B	
pgm	'00'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ProcessUSSData_02(Invkid :OCTETSTRING; prevbits: OCTETSTRING; follbits: OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	Return Result for Process Unstructured SS Data	
Element Name	**Element Value**	**Comments**
comp_part1	prevbits	
invokeId	Invkid	
comp_part2	follbits	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ProcessUSSRequest_02(Invkid :OCTETSTRING; prevbits: OCTETSTRING; follbits: OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	Reject or Return Error for Process Unstructured SS request	
Element Name	**Element Value**	**Comments**
comp_part1	prevbits	
invokeId	Invkid	
comp_part2	follbits	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ProgInd_omit	
Structured Type:	PI	
Derivation Path:		
Comments:	Omitted Progress Indicator.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
extb3	-	
cs	-	
spb	-	
loc	-	
extb4	-	
prd	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ProgInd_01	
Structured Type:	PI	
Derivation Path:		
Comments:	Progress Indicator with progress description #8 (inband info now available)	
Element Name	**Element Value**	**Comments**
iei	'00011110'B	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'1'B	
loc	'0000'B	
extb4	'1'B	
prd	'0001000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ProgInd_02	
Structured Type:	PI	
Derivation Path:		
Comments:	Progress Indicator containing progress indicator #4.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
loc	'0001'B	
extb4	'1'B	
prd	'0000100'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ProgInd_03	
Structured Type:	PI	
Derivation Path:		
Comments:	Progress Indicator containing progress indicator #8.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
loc	'0001'B	
extb4	'1'B	
prd	'0001000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ProgInd_03iei	
Structured Type:	PI	
Derivation Path:		
Comments:	Progress Indicator containing progress indicator #8.	
Element Name	**Element Value**	**Comments**
iei	'00011110'B	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
loc	'0001'B	
extb4	'1'B	
prd	'0001000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RachCntrlPara(maxtx:B_2; txint:B_4; re:B_1)	
Structured Type:	RACHCP	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
maxrtx	maxtx	
txint	txint	
cba	'0'B	
re	re	
acc_2	'00000'B	
ec	'0'B	
acc_1	'0000000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RachCntrlPara_01	
Structured Type:	RACHCP	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
maxrtx	'00'B	
txint	'0010'B	
cba	'0'B	
re	'0'B	
acc_2	'00000'B	
ec	'0'B	
acc_1	'0000000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RachCntrlPara_r01	
Structured Type:	RACHCP	
Derivation Path:		
Comments:	Call reestablishment is not allowed in the cell.	
Element Name	**Element Value**	**Comments**
maxrtx	'00'B	
txint	'0010'B	
cba	'0'B	
re	'1'B	
acc_2	'00000'B	
ec	'0'B	
acc_1	'0000000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RegisterSSRslt_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A2210201'O	
invokeId	id	
comp_part2	'301C02010AA01704012A301230208301 10840107850581003421438701 05'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RegisterSSRslt_02(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A2800201'O	
invokeId	id	
comp_part2	'308002010AA0800401213080308083016 08401078505810034214300000000000 00000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RegisterSSErr_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A3060201'O	
invokeId	id	
comp_part2	'02010A'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RegisterSSRej_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A4800201'O	
invokeId	id	
comp_part2	'8101030000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RegPasswdSSRslt_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A2800201'O	
invokeId	id	
comp_part2	'308002011112043938373600000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RegPasswdSSErr_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A3060201'O	
invokeId	id	
comp_part2	'020113'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RegPasswdSSErr_02(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A3800201'O	
invokeId	id	
comp_part2	'0201260000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RegPasswdSSErr_03(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A3090201'O	
invokeId	id	
comp_part2	'0201250A0102'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SI3RestOctetDef	
Structured Type:	SI3RO	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
pi	'0'B	
cbq	'0'B	
cro	'101011'B	
to	'001'B	
pt	'01011'B	
poi	'0'B	
po	'01'B	
ti	'0'B	
ecsc	'1'B	
spr1	'011'B	
spr2	'00101011'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SI3RestOctet_inv	
Structured Type:	SI3RO	
Derivation Path:		
Comments:	Invalid rest octet (not all '2B'O).	
Element Name	**Element Value**	**Comments**
pi	'0'B	
cbq	'0'B	
cro	'101011'B	
to	'001'B	
pt	'01011'B	
poi	'0'B	
po	'01'B	
ti	'0'B	
ecsc	'1'B	
spr1	'011'B	
spr2	'11101110'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SI4RestOctetDef	
Structured Type:	SI4RO	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
pi	'0'B	
cbq	'0'B	
cro	'101011'B	
to	'001'B	
pt	'01011'B	
poi	'0'B	
po	'01'B	
spr1	'01011'B	
spr2	'00101011'B	
spr3	'00101011'B	
spr4	'00101011'B	
spr5	'00101011'B	
spr6	'00101011'B	
spr7	'00101011'B	
spr8	'00101011'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SI4RestOctet_CBMS	
Structured Type:	SI4RO	
Derivation Path:		
Comments:	Rest octets for SMSCB SysInf4 message.	
Element Name	**Element Value**	**Comments**
pi	'0'B	
cbq	'0'B	
cro	'101011'B	
to	'001'B	
pt	'01011'B	
poi	'0'B	
po	'01'B	
spr1	'01011'B	
spr2	'00101011'B	
spr3	-	
spr4	-	
spr5	-	
spr6	-	
spr7	-	
spr8	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SI4RestOctet_inv	
Structured Type:	SI4RO	
Derivation Path:		
Comments:	Invalid rest octet (not all '2B'O).	
Element Name	**Element Value**	**Comments**
pi	'0'B	
cbq	'0'B	
cro	'101011'B	
to	'001'B	
pt	'01011'B	
poi	'0'B	
po	'01'B	
spr1	'01011'B	
spr2	'00101011'B	
spr3	'00101011'B	
spr4	'00101011'B	
spr5	'00101011'B	
spr6	'00101011'B	
spr7	'00101011'B	
spr8	'11101110'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SerialNumber_01	
Structured Type:	SERIAL_NUMBER	
Derivation Path:		
Comments:	Serial number for first SMSCB, GSM 3.41, 9.3.2	
Element Name	**Element Value**	**Comments**
gs	'00'B	
message_code	'0000000000'B	
update_number	'0000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SerialNumber_02	
Structured Type:	SERIAL_NUMBER	
Derivation Path:		
Comments:	Serial number for second SMSCB, GSM 3.41, 9.3.2	
Element Name	**Element Value**	**Comments**
gs	'00'B	
message_code	'0000000001'B	
update_number	'0000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SerialNumber_03	
Structured Type:	SERIAL_NUMBER	
Derivation Path:		
Comments:	Serial number for third SMSCB, same message code as second SMSCB but updated GSM 3.41, 9.3.2	
Element Name	**Element Value**	**Comments**
gs	'00'B	
message_code	'0000000001'B	
update_number	'0001'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	USSDReq_01(Invkid :OCTETSTRING; prevbits: OCTETSTRING; follbits: OCTETSTRING; ussdString: IA5String)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	ReturnResult for Process Unstructured SS request	
Element Name	**Element Value**	**Comments**
comp_part1	prevbits	
invokeId	Invkid	
comp_part2	follbits	
comp_part3	OC_CodingOfUssdString(ussdString)	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	USSDReq_03(Invkid :OCTETSTRING; ussdstring: IA5String)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	Invoke for UnstructuredSS-Request	
Element Name	**Element Value**	**Comments**
comp_part1	'A11E0201'O	
invokeId	Invkid	
comp_part2	'02013C0F'O	
comp_part3	OC_CodingOfUssdString(ussdstring)	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	USSDReq_06(Invkid :OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	Unstructured SS - Request	
Element Name	**Element Value**	**Comments**
comp_part1	'A2030201'O	
invokeId	Invkid	
comp_part2	-	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	EraseSSRslt_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A21D0201'O	
invokeId	id	
comp_part2	'301802010BA0800401283080308083016 0840104000000000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	EraseSSRslt_02(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A2140201'O	
invokeId	id	
comp_part2	'300F02010BA00A04012B30053003840104'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	EraseSSErr_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A3060201'O	
invokeId	id	
comp_part2	'02010B'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	EraseSSRej_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A4800201'O	
invokeId	id	
comp_part2	'8101030000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ActivateSSRslt_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	CF all sychronous services	
Element Name	**Element Value**	**Comments**
comp_part1	'A2800201'O	
invokeId	id	
comp_part2	'301402010CA0800401203008300682016884010700000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ActivateSSRslt_02(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	CFU all basic services	
Element Name	**Element Value**	**Comments**
comp_part1	'A2180201'O	
invokeId	id	
comp_part2	'308002010CA08004012130053003840107000000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ActivateSSRslt_03(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	BAOC all synchronous services	
Element Name	**Element Value**	**Comments**
comp_part1	'A21B0201'O	
invokeId	id	
comp_part2	'308002010CA18004019230083006820168840107000000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ActivateSSRslt_04(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	BICRoam for all basic service groups.	
Element Name	**Element Value**	**Comments**
comp_part1	'A2180201'O	
invokeId	id	
comp_part2	'301302010CA10E04019B308030808401 0700000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ActivateSSErr_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	BOIC	
Element Name	**Element Value**	**Comments**
comp_part1	'A3060201'O	
invokeId	id	
comp_part2	'020113'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ActivateSSErr_02(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	BOIC	
Element Name	**Element Value**	**Comments**
comp_part1	'A3800201'O	
invokeId	id	
comp_part2	'0201260000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	DeactivateSSRslt_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	CFC speech	
Element Name	**Element Value**	**Comments**
comp_part1	'A21B0201'O	
invokeId	id	
comp_part2	'301602010DA0800401283080300683011084010600000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	DeactivateSSRslt_02(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	CFNRc all facsimile	
Element Name	**Element Value**	**Comments**
comp_part1	'A2190201'O	
invokeId	id	
comp_part2	'301402010DA00F04012B300A30808301608401060000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	DeactivateSSRslt_03(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	DeactivateSSRslt for Speech	
Element Name	**Element Value**	**Comments**
comp_part1	'A2800201'O	
invokeId	id	
comp_part2	'300C02010DA1073005300383011000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	DeactivateSSRslt_04(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	DeactivateSSRslt for all facsimile	
Element Name	**Element Value**	**Comments**
comp_part1	'A2800201'O	
invokeId	id	
comp_part2	'300C02010DA107300530038301600000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	DeactivateSSErr_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A3800201'O	
invokeId	id	
comp_part2	'0201130000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	DeactivateSSErr_02(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A3060201'O	
invokeId	id	
comp_part2	'020126'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_01	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1270201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11881013C8202008C830164840200FA85010086010087020258000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_02	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1250201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1168101008201008 30164840203E8850100860100870100000 00'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_03	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1290201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11A810209C48202 00A0830200C8840213888501008601008 70202580000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_04	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1240201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11581010A82010A830164840100850164860 10A87010A0000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_05	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1270201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11881017D8202012C830164840200FA85016486010A8702012C0000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_06	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1240201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1158101008201008301008401008501008601008701000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_07	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1250201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1168101008201008301648402 03E885010086010087010000 00'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_08	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1270201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11881013C8202008C830164840200FA85010086010087022580000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_09	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1240201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11581010A82010A830164840100850100860100870010A0000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_10	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1270201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11881017D8202012C830164840200FA85010086010087020 12C0000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_11	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1260201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1178101648202011883016484016485010086010087020258 0000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_12	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1260201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1178101648202008C830164840132850100860100870202580000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_13	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1250201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1168101648202019083016484010085010086010087010000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_14	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1190201'O	
invokeId	'00'O	
comp_part2	'02017D3011800172A1808101648202019 08301640000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_15	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1250201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1168101468202019 08301648401008501008601008701000 00'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_16	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1260201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1178102008282020190830164840100850100860100870100000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_17	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1260201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A117810200BE8202019083016484010085010086010087010000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_18	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1260201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1178102012282020190830164840100850100860100870100000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_19	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1800201'O	
invokeId	'00'O	
comp_part2	'02017D301B800172A116810164820202268301648401648501008601008701640000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_20	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1800201'O	
invokeId	'00'O	
comp_part2	'02017D301B800172A11681010A8202012C830164840100850100860100870100000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_21	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging, charging value is 0.	
Element Name	**Element Value**	**Comments**
comp_part1	'A1800201'O	
invokeId	'00'O	
comp_part2	'02017D301A800172A115810100820100830100840100850100860100870100000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	GetPasswdSS_01(linkid:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	getpassword	
Element Name	**Element Value**	**Comments**
comp_part1	'A10C0201'O	
invokeId	'00'O	
comp_part2	'8001'O	
comp_part3	linkid	
comp_part4	'0201120A0100'O	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	GetPasswdSS_02(linkid:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	getpassword (enter new password)	
Element Name	**Element Value**	**Comments**
comp_part1	'A10C0201'O	
invokeId	'00'O	
comp_part2	'8001'O	
comp_part3	linkid	
comp_part4	'0201120A0101'O	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	GetPasswdSS_03(linkid:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	getpassword (enter new password again)	
Element Name	**Element Value**	**Comments**
comp_part1	'A10C0201'O	
invokeId	'00'O	
comp_part2	'8001'O	
comp_part3	linkid	
comp_part4	'0201120A0102'O	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	InterrogateSSRslt_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	CFB all basic services	
Element Name	**Element Value**	**Comments**
comp_part1	'A20D0201'O	
invokeId	id	
comp_part2	'308002010E8001040000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	InterrogateSSRslt_02(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	CFNRy Speech	
Element Name	**Element Value**	**Comments**
comp_part1	'A2190201'O	
invokeId	id	
comp_part2	'301402010EA30F300D83011184010785 058100342143'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	InterrogateSSRslt_03(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	BAIC	
Element Name	**Element Value**	**Comments**
comp_part1	'A20D0201'O	
invokeId	id	
comp_part2	'300802010EA203830111'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	InterrogateSSRslt_04(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	BAIC	
Element Name	**Element Value**	**Comments**
comp_part1	'A20B0201'O	
invokeId	id	
comp_part2	'300602010E800107'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	InterrogateSSErr_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	CFNRc	
Element Name	**Element Value**	**Comments**
comp_part1	'A3060201'O	
invokeId	id	
comp_part2	'020112'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	InterrogateSSErr_02(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A3800201'O	
invokeId	id	
comp_part2	'0201120000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	InterrogateSSRej_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A4800201'O	
invokeId	id	
comp_part2	'8101030000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	InterrogateSSRej_02(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
comp_part1	'A4060201'O	
invokeId	id	
comp_part2	'810103'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	NotificationSS_01	
Structured Type:	Component_T	
Derivation Path:		
Comments:	incoming call forwarded.	
Element Name	**Element Value**	**Comments**
comp_part1	'A1100201'O	
invokeId	'01'O	
comp_part2	'020110308081012985010200000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	NotificationSS_02	
Structured Type:	Component_T	
Derivation Path:		
Comments:	CFU provisioned, registered and active	
Element Name	**Element Value**	**Comments**
comp_part1	'A1800201'O	
invokeId	'00'O	
comp_part2	'020110300681012184010700000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	NotificationSS_03	
Structured Type:	Component_T	
Derivation Path:		
Comments:	CFC provisioned, registered and active	
Element Name	**Element Value**	**Comments**
comp_part1	'A1100201'O	
invokeId	'01'O	
comp_part2	'020110308081012884010700000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	NotificationSS_04	
Structured Type:	Component_T	
Derivation Path:		
Comments:	CFNRc forwarded call	
Element Name	**Element Value**	**Comments**
comp_part1	'A1800201'O	
invokeId	'00'O	
comp_part2	'020110300681012B8501010000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	NotificationSS_05	
Structured Type:	Component_T	
Derivation Path:		
Comments:	BI	
Element Name	**Element Value**	**Comments**
comp_part1	'A10E0201'O	
invokeId	'00'O	
comp_part2	'0201103006810199840107'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	NotificationSS_06(Invkid :OCTETSTRING; ussdstring: IA5String)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	UnstructuredSS-Notify	
Element Name	**Element Value**	**Comments**
comp_part1	'A1140201'O	
invokeId	Invkid	
comp_part2	'02013D0F'O	
comp_part3	OC_CodingOfUssdString(ussdstring)	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	NotificationSS_08(Invkid :OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	UnstructuredSS-Notify	
Element Name	**Element Value**	**Comments**
comp_part1	'A2030201'O	
invokeId	Invkid	
comp_part2	-	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BuildMptySSRslt_01(id:OCTETSTRING)	
Structured Type:	Component_T	
Derivation Path:		
Comments:	ReturnResult for buildMPTY	
Element Name	**Element Value**	**Comments**
comp_part1	'A2800201'O	
invokeId	id	
comp_part2	'0000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RelTmdDif_omit	
Structured Type:	TDIF	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
value	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RelTmdDif_01(ta_a: INTEGER)	
Structured Type:	TDIF	
Derivation Path:		
Comments:	ta_a: timing advance value in cell A.	
Element Name	**Element Value**	**Comments**
iei	'01111011'B	
iel	'01'O	
value	OC_IntToOct(((2*ta_a + 10) MOD 256), 1)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Rqr1(Rr: BITSTRING; Fn: FN)	
Structured Type:	RQR	
Derivation Path:		
Comments:	not address the MS under test	
Element Name	**Element Value**	**Comments**
ra	INT_TO_BIT((BIT_TO_INT(Rr) + 1), 8)	
fn	OC_FnInc(Fn, 2)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Rqr2(Rr: BITSTRING; Fn: FN)	
Structured Type:	RQR	
Derivation Path:		
Comments:	To address the MS under test	
Element Name	**Element Value**	**Comments**
ra	Rr	
fn	Fn	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Rqr3	
Structured Type:	RQR	
Derivation Path:		
Comments:	not pertaining to the MS under test	
Element Name	**Element Value**	**Comments**
ra	'00000000'B	
fn	Fn_01	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Signal_01	
Structured Type:	SIGNAL	
Derivation Path:		
Comments:	signal value is arbitrarily selected.	
Element Name	**Element Value**	**Comments**
iei	'00110100'B	
sigv	'00000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Signal_02	
Structured Type:	SIGNAL	
Derivation Path:		
Comments:	Signal IE with value #7 "call waiting tone on"	
Element Name	**Element Value**	**Comments**
iei	'00110100'B	
sigv	'00000111'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	StartingTm_01(fn:FN)	
Structured Type:	STRT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'01111100'B	
fn	fn	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	StartingTm_omit	
Structured Type:	STRT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
fn	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SubAdd_01	
Structured Type:	SUBAD	
Derivation Path:		
Comments:	containing arbitrary spare bits	
Element Name	**Element Value**	**Comments**
extb	'1'B	
tos	'000'B	
oei	'0'B	
sp3b	'111'B	
si	'5001'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Synchi_omit	
Structured Type:	SYNCHI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
nci	-	
rot	-	
si	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Synchi(rot:B_1; si:B_2)	
Structured Type:	SYNCHI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'1101'B	
nci	'0'B	
rot	rot	
si	si	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Synchi_01	
Structured Type:	SYNCHI	
Derivation Path:		
Comments:	coded as a comprehension required IEI.	
Element Name	**Element Value**	**Comments**
iei	'0000'B	
nci	'0'B	
rot	'0'B	
si	'00'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TimingAdv(bits:INTEGER)	
Structured Type:	TA	
Derivation Path:		
Comments:	Time Advance value is pameterized.	
Element Name	**Element Value**	**Comments**
iei	-	
sprb	'00'B	
value	INT_TO_BIT(bits, 6)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TimingAdv_iei(bits:INTEGER)	
Structured Type:	TA	
Derivation Path:		
Comments:	0 time advance.	
Element Name	**Element Value**	**Comments**
iei	'01111101'B	
sprb	'00'B	
value	INT_TO_BIT(bits, 6)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TimingAdv_inv	
Structured Type:	TA	
Derivation Path:		
Comments:	containing arbitrary spare bits	
Element Name	**Element Value**	**Comments**
iei	-	
sprb	'11'B	
value	'000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TimingAdv_omit	
Structured Type:	TA	
Derivation Path:		
Comments:	Time Advance value is pameterized.	
Element Name	**Element Value**	**Comments**
iei	-	
sprb	-	
value	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_01	
Structured Type:	TI	
Derivation Path:		
Comments:	used by the MS in the transaction initiated by the test system.	
Element Name	**Element Value**	**Comments**
ti_f	'1'B	
ti_v	'000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_02	
Structured Type:	TI	
Derivation Path:		
Comments:	used in the messages sent to the MS in the transaction initiated by test system.	
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	'000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_03	
Structured Type:	TI	
Derivation Path:		
Comments:	used in the messages sent to the MS in the transaction initiated by test system.	
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	'110'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_04	
Structured Type:	TI	
Derivation Path:		
Comments:	arbitrary value	
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	'011'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_05	
Structured Type:	TI	
Derivation Path:		
Comments:	used by the MS in the transaction initiated by the test system.	
Element Name	**Element Value**	**Comments**
ti_f	'1'B	
ti_v	'110'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_06	
Structured Type:	TI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	'111'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_07(ti :TI_V)	
Structured Type:	TI	
Derivation Path:		
Comments:	used in the messages sent to the MS in the transaction initiated by the MS.	
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	ti	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_08(ti:TI_V)	
Structured Type:	TI	
Derivation Path:		
Comments:	used in the messages sent to the MS in the transaction initiated by the system simulator.	
Element Name	**Element Value**	**Comments**
ti_f	'1'B	
ti_v	ti	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_09	
Structured Type:	TI	
Derivation Path:		
Comments:	used by the MS in the transaction initiated by the MS.	
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Tmsi_01	
Structured Type:	TMSI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
tmsi_val	TSPX_TMSI	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Tmsi_r01	
Structured Type:	TMSI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
tmsi_val	OC_IncTmsi(TSPX_TMSI, '01'O)	
Detailed Comments:	Note: the TSPX_TMSI + '01'O shall not be identical to TSPX_TMSI1	

Structured Type Constraint		
Constraint Name:	Tmsi_r03	
Structured Type:	TMSI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
tmsi_val	OC_IncTmsi(TSPX_TMSI, '02'O)	
Detailed Comments:	Note: the TSPX_TMSI + '02'O shall not be identical to TSPX_TMSI1	

Structured Type Constraint		
Constraint Name:	Tmsi_r04	
Structured Type:	TMSI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
tmsi_val	OC_IncTmsi(TSPX_TMSI, '03'O)	
Detailed Comments:	Note: the TSPX_TMSI + '03'O shall not be identical to TSPX_TMSI1	

Structured Type Constraint		
Constraint Name:	Tmsi_r05	
Structured Type:	TMSI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
tmsi_val	OC_IncTmsi(TSPX_TMSI, '04'O)	
Detailed Comments:	Note: the TSPX_TMSI + '04'O shall not be identical to TSPX_TMSI1	

Structured Type Constraint		
Constraint Name:	TonNpi_01	
Structured Type:	TON_NPI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
extb	'0'B	
ton	'000'B	
npi	'0000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TonNpi_02	
Structured Type:	TON_NPI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
extb	'1'B	
ton	'000'B	
npi	'0000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TonNpi_03	
Structured Type:	TON_NPI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
extb	'1'B	
ton	'001'B	
npi	'0001'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TonNpi_04	
Structured Type:	TON_NPI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
extb	'1'B	
ton	?	
npi	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	UnknownIE_01	
Structured Type:	UNKWN	
Derivation Path:		
Comments:	An invalid IE coded as comprehension required .	
Element Name	**Element Value**	**Comments**
iei	'00000000'B	
iel	'01'O	
contents	'54'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	UnknownIE_02	
Structured Type:	UNKWN	
Derivation Path:		
Comments:	An invalid FIE coded as unknown IE.	
Element Name	**Element Value**	**Comments**
iei	'01001100'B	
iel	'01'O	
contents	'FF'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	UnknownIE	
Structured Type:	CHD	
Derivation Path:		
Comments:	used as unkown IE	
Element Name	**Element Value**	**Comments**
iei	'01101001'B	
cht_schn	'00000'B	
tn	'010'B	
tsc	'101'B	
hch	'1'B	
maio	'001100'B	
hsn	'101010'B	
spr	-	
arfcn	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_01(tpoa1, rpoa_mt:BCDN; rpmr: MR; timezone:TZONES)	
Structured Type:	CPDATA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((159 + OC_LengthOfBCDN(tpoa1) + OC_LengthOfBCDN(rpoa_mt)), 1)	
rpack	-	
rpdata	RpData_01(rpoa_mt, tpoa1, rpmr, timezone)	
rperr	-	
rpsmma	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_02(rpmr: MR)	
Structured Type:	CPDATA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
rpack	RpAck_01(rpmr)	
rpdata	-	
rperr	-	
rpsmma	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_03(rpdat:RPDATA)	
Structured Type:	CPDATA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
rpack	-	
rpdata	rpdat	
rperr	-	
rpsmma	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_04(rpmr: MR)	
Structured Type:	CPDATA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
rpack	RpAck_02(rpmr)	
rpdata	-	
rperr	-	
rpsmma	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_05(tpoa1, rpoa_mt:BCDN; rpmr: MR; timezone:TZONES)	
Structured Type:	CPDATA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((159 + OC_LengthOfBCDN(tpoa1) + OC_LengthOfBCDN(rpoa_mt)), 1)	
rpack	-	
rpdata	RpData_05(rpoa_mt, tpoa1, rpmr, timezone)	
rperr	-	
rpsmma	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_06(tpoa1, rpoa_mt:BCDN; rpmr: MR; timezone:TZONES)	
Structured Type:	CPDATA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((159 + OC_LengthOfBCDN(tpoa1) + OC_LengthOfBCDN(rpoa_mt)), 1)	
rpack	-	
rpdata	RpData_06(rpoa_mt, tpoa1, rpmr, timezone)	
rperr	-	
rpsmma	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_07(rpmr: MR)	
Structured Type:	CPDATA	
Derivation Path:		
Comments:	RP Error: Protocol Error, unspecified	
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
rpack	-	
rpdata	-	
rperr	RpError_01(rpmr)	
rpsmma	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_08(rpmr: MR)	
Structured Type:	CPDATA	
Derivation Path:		
Comments:	RP Error: Memory Capability Exceeded	
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
rpack	-	
rpdata	-	
rperr	RpError_02(rpmr)	
rpsmma	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_09	
Structured Type:	CPDATA	
Derivation Path:		
Comments:	RP_SMAA	
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
rpack	-	
rpdata	-	
rperr	-	
rpsmma	RpSMMA_01	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_10(tpoa1, rpoa_mt:BCDN; rpmr: MR; timezone:TZONES)	
Structured Type:	CPDATA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((159 + OC_LengthOfBCDN(tpoa1) + OC_LengthOfBCDN(rpoa_mt)), 1)	
rpack	-	
rpdata	RpData_07(rpoa_mt, tpoa1, rpmr, timezone)	
rperr	-	
rpsmma	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_12(tpda, rpoa_mt:BCDN; tpmr: MR; rpmr: MR; timezone:TZONES)	
Structured Type:	CPDATA	
Derivation Path:		
Comments:	n -> ms, RP DATA(SMS-STATUS-REPORT)	
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((25 + OC_LengthOfBCDN(tpda) + OC_LengthOfBCDN(rpoa_mt)), 1)	
rpack	-	
rpdata	RpData_09(rpoa_mt, tpda, tpmr, rpmr, timezone)	
rperr	-	
rpsmma	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_15(tpoa1: BCDN; rpoa_mt: BCDN; smtype: INTEGER; text: IA5String; rpmr: MR; timezone:TZONES)	
Structured Type:	CPDATA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((159 + OC_LengthOfBCDN(tpoa1) + OC_LengthOfBCDN(rpoa_mt)), 1)	
rpack	-	
rpdata	RpData_12(tpoa1, rpoa_mt, smtype, text, rpmr, timezone)	
rperr	-	
rpsmma	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_16(tpoa1: BCDN; rpoa_mt: BCDN; text: IA5String; rpmr: MR; timezone:TZONES)	
Structured Type:	CPDATA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((159 + OC_LengthOfBCDN(tpoa1) + OC_LengthOfBCDN(rpoa_mt)), 1)	
rpack	-	
rpdata	RpData_13(tpoa1, rpoa_mt, text, rpmr, timezone)	
rperr	-	
rpsmma	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CpData_17(tpda,rpda:BCDN; tpud: TPUD)	
Structured Type:	CPDATA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
rpack	-	
rpdata	RpData_14(tpda, rpda,tpud)	
rperr	-	
rpsmma	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpData_01(rpoa_mt, tpoa1:BCDN; rpmr: MR; timezone:TZONES)	
Structured Type:	RPDATA	
Derivation Path:		
Comments:	n->ms	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'001'B	
rpmr	rpmr	
rpOaddr	RpOrigAddr_01(rpoa_mt)	
rpDaddr	RpDestAddr_01	
rpusrdat	RpUsrData_01(tpoa1, timezone)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpData_03	
Structured Type:	RPDATA	
Derivation Path:		
Comments:	ms->n	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'000'B	
rpmr	?	
rpOaddr	RpOrigAddr_02	
rpDaddr	RpDestAddr_02	
rpusrdat	RpUsrData_02	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpData_05(rpoa_mt, tpoa1:BCDN; rpmr: MR; timezone:TZONES)	
Structured Type:	RPDATA	
Derivation Path:		
Comments:	n->ms	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'001'B	
rpmr	rpmr	
rpOaddr	RpOrigAddr_01(rpoa_mt)	
rpDaddr	RpDestAddr_01	
rpusrdat	RpUsrData_03(tpoa1, timezone)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpData_06(rpoa_mt, tpoa1:BCDN; rpmr: MR; timezone:TZONES)	
Structured Type:	RPDATA	
Derivation Path:		
Comments:	n->ms	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'001'B	
rpmr	rpmr	
rpOaddr	RpOrigAddr_01(rpoa_mt)	
rpDaddr	RpDestAddr_01	
rpusrdat	RpUsrData_04(tpoa1, timezone)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpData_07(rpoa_mt, tpoa1:BCDN; rpmr: MR; timezone:TZONES)	
Structured Type:	RPDATA	
Derivation Path:		
Comments:	n->ms	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'001'B	
rpmr	rpmr	
rpOaddr	RpOrigAddr_01(rpoa_mt)	
rpDaddr	RpDestAddr_01	
rpusrdat	RpUsrData_05(tpoa1, timezone)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpData_08	
Structured Type:	RPDATA	
Derivation Path:		
Comments:	ms->n, status report requested	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'000'B	
rpmr	?	
rpOaddr	RpOrigAddr_02	
rpDaddr	RpDestAddr_02	
rpusrdat	RpUsrData_06	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpData_09(rpoa_mt, tpda:BCDN; tpmr: MR; rpmr: MR; timezone:TZONES)	
Structured Type:	RPDATA	
Derivation Path:		
Comments:	n->ms, RP DATA(SMS-STATUS-REPORT)	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'001'B	
rpmr	rpmr	
rpOaddr	RpOrigAddr_01(rpoa_mt)	
rpDaddr	RpDestAddr_01	
rpusrdat	RpUsrData_07(tpda, tpmr, timezone)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpData_11(rpmr: MR; tpcom:SMCMD)	
Structured Type:	RPDATA	
Derivation Path:		
Comments:	ms->n, RP data(SMS-COMMAND(Delete))	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'000'B	
rpmr	OC_IntToOct((OC_OctToInt(rpmr) + 1), 1)	
rpOaddr	RpOrigAddr_02	
rpDaddr	RpDestAddr_02	
rpusrdat	RpUsrData_09(tpcom)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpData_12(tpoa1: BCDN; rpoa_mt: BCDN; smtype: INTEGER; text: IA5String; rpmr: MR; timezone:TZONES)	
Structured Type:	RPDATA	
Derivation Path:		
Comments:	n->ms	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'001'B	
rpmr	rpmr	
rpOaddr	RpOrigAddr_03(rpoa_mt)	
rpDaddr	RpDestAddr_01	
rpusrdat	RpUsrData_10(tpoa1, smtype, text, timezone)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpData_13(tpoa: BCDN; rpoa_mt: BCDN; text: IA5String; rpmr: MR; timezone:TZONES)	
Structured Type:	RPDATA	
Derivation Path:		
Comments:	n->ms	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'001'B	
rpmr	rpmr	
rpOaddr	RpOrigAddr_03(rpoa_mt)	
rpDaddr	RpDestAddr_01	
rpusrdat	RpUsrData_11(tpoa, text, timezone)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpData_14(tpda: BCDN; rpda: BCDN; tpud: TPUD)	
Structured Type:	RPDATA	
Derivation Path:		
Comments:	ms->n	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'000'B	
rpmr	?	
rpOaddr	RpOrigAddr_02	
rpDaddr	RpDestAddr_03(rpda)	
rpusrdat	RpUsrData_12(tpda,tpud)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpAck_01(rpmr: MR)	
Structured Type:	RPACK	
Derivation Path:		
Comments:	ms->n	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'010'B	
rpmr	rpmr	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpAck_02(rpmr: MR)	
Structured Type:	RPACK	
Derivation Path:		
Comments:	RP_ACK n->ms	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'011'B	
rpmr	rpmr	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpError_01(rpmr: MR)	
Structured Type:	RPERR	
Derivation Path:		
Comments:	Protocol error, unspecified	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'100'B	
rpmr	rpmr	
rpcau	RpCause_01	
rpusrdat	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpError_02(rpmr: MR)	
Structured Type:	RPERR	
Derivation Path:		
Comments:	Memory Capacity Exceeded	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'100'B	
rpmr	rpmr	
rpcau	RpCause_02	
rpusrdat	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpCause_01	
Structured Type:	RPCAU	
Derivation Path:		
Comments:	ms->n, Protocol error, unspecified, cuase number 111.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
extb2	'0'B	
rpcau_class	'110'B	
rpcau_va	'1111'B	
rpcau_di	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpCause_02	
Structured Type:	RPCAU	
Derivation Path:		
Comments:	ms->n, Memory Capacity Exceeded, cause number 22.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
extb2	'0'B	
rpcau_class	'001'B	
rpcau_va	'0110'B	
rpcau_di	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpSMMA_01	
Structured Type:	RPSMMA	
Derivation Path:		
Comments:	ms->n	
Element Name	**Element Value**	**Comments**
sprb	'00000'B	
rpmti	'110'B	
rpmr	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpUsrData_01(tpoa1:BCDN; timezone:TZONES)	
Structured Type:	RPUSRDAT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((153 + OC_LengthOfBCDN(tpoa1)), 1)	
tpdeliver	TpDeliver_01(tpoa1, timezone)	
tpsubmit	-	
tpstatus_rpt	-	
tpcommand	-	
tpdlvr_sbmt_rpt	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpUsrData_02	
Structured Type:	RPUSRDAT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
tpdeliver	-	
tpsubmit	TpSubmit_01	
tpstatus_rpt	-	
tpcommand	-	
tpdlvr_sbmt_rpt	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpUsrData_03(tpoa1:BCDN; timezone:TZONES)	
Structured Type:	RPUSRDAT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((153+OC_LengthOfBCDN(tpoa1)), 1)	
tpdeliver	TpDeliver_02(tpoa1, timezone)	
tpsubmit	-	
tpstatus_rpt	-	
tpcommand	-	
tpdlvr_sbmt_rpt	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpUsrData_04(tpoa1:BCDN; timezone:TZONES)	
Structured Type:	RPUSRDAT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((153 + OC_LengthOfBCDN(tpoa1)), 1)	
tpdeliver	TpDeliver_03(tpoa1, timezone)	
tpsubmit	-	
tpstatus_rpt	-	
tpcommand	-	
tpdlvr_sbmt_rpt	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpUsrData_05(tpoa1:BCDN; timezone:TZONES)	
Structured Type:	RPUSRDAT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((153+OC_LengthOfBCDN(tpoa1)), 1)	
tpdeliver	TpDeliver_04(tpoa1, timezone)	
tpsubmit	-	
tpstatus_rpt	-	
tpcommand	-	
tpdlvr_sbmt_rpt	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpUsrData_06	
Structured Type:	RPUSRDAT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
tpdeliver	-	
tpsubmit	TpSubmit_02	
tpstatus_rpt	-	
tpcommand	-	
tpdlvr_sbmt_rpt	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpUsrData_07(tpda:BCDN; mr: MR; timezone:TZONES)	
Structured Type:	RPUSRDAT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((19 + OC_LengthOfBCDN(tpda)), 1)	
tpdeliver	-	
tpsubmit	-	
tpstatus_rpt	TpStatusReport_01(tpda, mr, timezone)	
tpcommand	-	
tpdlvr_sbmt_rpt	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpUsrData_09(tpcom:SMCMD)	
Structured Type:	RPUSRDAT	
Derivation Path:		
Comments:	SMS-COMMAND(Delete)	
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
tpdeliver	-	
tpsubmit	-	
tpstatus_rpt	-	
tpcommand	tpcom	
tpdlvr_sbmt_rpt	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpUsrData_10(tpoa1: BCDN; smtype: INTEGER; text: IA5String; timezone:TZONES)	
Structured Type:	RPUSRDAT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((153+OC_LengthOfBCDN(tpoa1)), 1)	
tpdeliver	TpDeliver_05(tpoa1, smtype, text, timezone)	
tpsubmit	-	
tpstatus_rpt	-	
tpcommand	-	
tpdlvr_sbmt_rpt	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpUsrData_11(tpoa1: BCDN; text: IA5String; timezone:TZONES)	
Structured Type:	RPUSRDAT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((153+OC_LengthOfBCDN(tpoa1)), 1)	
tpdeliver	TpDeliver_06(tpoa1, text, timezone)	
tpsubmit	-	
tpstatus_rpt	-	
tpcommand	-	
tpdlvr_sbmt_rpt	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpUsrData_12(tpda: BCDN; tpud: TPUD)	
Structured Type:	RPUSRDAT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
tpdeliver	-	
tpsubmit	TpSubmit_03(tpda,tpud)	
tpstatus_rpt	-	
tpcommand	-	
tpdlvr_sbmt_rpt	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TpDeliver_01(tpoa1:BCDN; timezone:TZONES)	
Structured Type:	SMDLVR	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
rp	'0'B	
udhi	'0'B	
sri	'0'B	
sprb2	'00'B	
mms	'0'B	
mti	'00'B	
oa	SmOrigAddr_01(tpoa1)	
pid	Tppid_01	
dcs	Tpdcs_01	
scts	OC_GetSCTimeStamp(timezone)	
udl	'A0'O	
ud	OC_ComputeSMContents(160)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TpDeliver_02(tpoa1:BCDN; timezone:TZONES)	
Structured Type:	SMDLVR	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
rp	'0'B	
udhi	'0'B	
sri	'0'B	
sprb2	'00'B	
mms	'0'B	
mti	'00'B	
oa	SmOrigAddr_01(tpoa1)	
pid	Tppid_01	
dcs	Tpdcs_02	
scts	OC_GetSCTimeStamp(timezone)	
udl	'A0'O	
ud	OC_ComputeSMContents(160)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TpDeliver_03(tpoa1:BCDN; timezone:TZONES)	
Structured Type:	SMDLVR	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
rp	'0'B	
udhi	'0'B	
sri	'0'B	
sprb2	'00'B	
mms	'0'B	
mti	'00'B	
oa	SmOrigAddr_01(tpoa1)	
pid	Tppid_01	
dcs	Tpdcs_03	
scts	OC_GetSCTimeStamp(timezone)	
udl	'A0'O	
ud	OC_ComputeSMContents(160)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TpDeliver_04(tpoa1:BCDN; timezone:TZONES)	
Structured Type:	SMDLVR	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
rp	'0'B	
udhi	'0'B	
sri	'0'B	
sprb2	'00'B	
mms	'0'B	
mti	'00'B	
oa	SmOrigAddr_01(tpoa1)	
pid	Tppid_01	
dcs	Tpdcs_04	
scts	OC_GetSCTimeStamp(timezone)	
udl	'A0'O	
ud	OC_ComputeSMContents(160)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TpDeliver_05(tpoa: BCDN; smtype: INTEGER; text: IA5String; timezone:TZONES)	
Structured Type:	SMDLVR	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
rp	'0'B	
udhi	'0'B	
sri	'0'B	
sprb2	'00'B	
mms	'1'B	
mti	'00'B	
oa	SmOrigAddr_02(tpoa)	
pid	Tppid_02(smtype)	
dcs	Tpdcs_01	
scts	OC_GetSCTimeStamp(timezone)	
udl	'A0'O	
ud	OC_ComputeSMContentsSpecText(160, text)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TpDeliver_06(tpoa: BCDN; text: IA5String; timezone:TZONES)	
Structured Type:	SMDLVR	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
rp	'1'B	
udhi	'0'B	
sri	'0'B	
sprb2	'00'B	
mms	'1'B	
mti	'00'B	
oa	SmOrigAddr_02(tpoa)	
pid	Tppid_01	
dcs	Tpdcs_01	
scts	OC_GetSCTimeStamp(timezone)	
udl	'A0'O	
ud	OC_ComputeSMContentsSpecText(160, text)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TpSubmit_01	
Structured Type:	SMSBMT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
rp	?	
udhi	?	
srr	?	
vpf	?	
rd	?	
mti	'01'B	
mr	?	
da	SmDestAddr_01	
pid	Tppid_01	
dcs	Tpdcs_01	
vp1	?	
vp7	-	
udl	?	
ud	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TpSubmit_02	
Structured Type:	SMSBMT	
Derivation Path:		
Comments:	status report requested	
Element Name	**Element Value**	**Comments**
rp	'0'B	
udhi	?	
srr	'1'B	
vpf	?	
rd	?	
mti	'01'B	
mr	?	
da	SmDestAddr_01	
pid	Tppid_01	
dcs	Tpdcs_01	
vp1	?	
vp7	-	
udl	?	
ud	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TpSubmit_03(tpda: BCDN; tpud: TPUD)	
Structured Type:	SMSBMT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
rp	?	
udhi	?	
srr	?	
vpf	'10'B	
rd	?	
mti	'01'B	
mr	?	
da	SmDestAddr_02(tpda)	
pid	Tppid_01	
dcs	Tpdcs_01	
vp1	'A7'O	
vp7	-	
udl	?	
ud	tpud	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TpStatusReport_01(tpda:BCDN; mr: MR; timezone:TZONES)	
Structured Type:	SMST_RPT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
sprb1	'00000'B	
mms	'1'B	
mti	'10'B	
mr	mr	
ra	SmDestAddr_02(tpda)	
scts	OC_GetSCTimeStamp(timezone)	
dt	OC_GetSCTimeStamp(timezone)	
st	TpStatus_01	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SmOrigAddr_01(tpoa1:BCDN)	
Structured Type:	TPA	
Derivation Path:		
Comments:	international number coded E.164 (GSM 11.10, 34.2.1.3, specific message contents). Used in SMS-DELIVER (n->ms)	
Element Name	**Element Value**	**Comments**
iel	OC_IntToOct((OC_LengthOfBCDN(tpoa1) * 2), 1)	
tonnpi	TonNpi_03	
digits	tpoa1	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SmOrigAddr_02(tpoa: BCDN)	
Structured Type:	TPA	
Derivation Path:		
Comments:	international number coded E.164 (GSM 11.10, 34.2.1.3, specific message contents). Used in SMS-DELIVER (n->ms)	
Element Name	**Element Value**	**Comments**
iel	OC_IntToOct((OC_LengthOfBCDN(tpoa) * 2), 1)	
tonnpi	TonNpi_03	
digits	tpoa	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SmDestAddr_01	
Structured Type:	TPA	
Derivation Path:		
Comments:	international number coded E.164 (GSM 11.10, 34.2.1.3, specific message contents). Used in SMS-SUBMIT (ms->n)	
Element Name	**Element Value**	**Comments**
iel	?	
tonnpi	TonNpi_04	
digits	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SmDestAddr_02(tpda: BCDN)	
Structured Type:	TPA	
Derivation Path:		
Comments:	international number coded E.164 (GSM 11.10, 34.2.1.3, specific message contents). Used in SMS-SUBMIT (ms->n)	
Element Name	**Element Value**	**Comments**
iel	OC_IntToOct((OC_LengthOfBCDN(tpda) * 2), 1)	
tonnpi	TonNpi_03	
digits	tpda	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpOrigAddr_01(rpoa_mt: BCDN)	
Structured Type:	CDPN	
Derivation Path:		
Comments:	Called party BCD number (CC information element) GSM 04.08, 10.5.4.7. Used in SMS-DELIVER (n->ms)	
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((OC_LengthOfBCDN(rpoa_mt) + 1), 1)	
tonnpi	TonNpi_03	
digits	rpoa_mt	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpOrigAddr_02	
Structured Type:	CDPN	
Derivation Path:		
Comments:	Called party BCD number (CC information element) GSM 04.08, 10.5.4.7. Used in SMS-SUBMIT (ms->n)	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'00'O	
tonnpi	-	
digits	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpOrigAddr_03(rpoa: BCDN)	
Structured Type:	CDPN	
Derivation Path:		
Comments:	Called party BCD number (CC information element) GSM 04.08, 10.5.4.7. Used in SMS-DELIVER (n->ms)	
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct((OC_LengthOfBCDN(rpoa) + 1), 1)	
tonnpi	TonNpi_03	
digits	rpoa	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpDestAddr_01	
Structured Type:	CDPN	
Derivation Path:		
Comments:	Called party BCD number (CC information element) GSM 04.08, 10.5.4.7. Used in SMS-DELIVER (n->ms)	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'00'O	
tonnpi	-	
digits	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpDestAddr_02	
Structured Type:	CDPN	
Derivation Path:		
Comments:	Called party BCD number (CC information element) GSM 04.08, 10.5.4.7. Used in SMS-SUBMIT (ms->n)	
Element Name	**Element Value**	**Comments**
iei	-	
iel	('02'O, '03'O, '04'O, '05'O, '06'O, '07'O, '08'O, '09'O, '0A'O, '0B'O)	
tonnpi	*	
digits	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RpDestAddr_03(rpda: BCDN)	
Structured Type:	CDPN	
Derivation Path:		
Comments:	Called party BCD number (CC information element) GSM 04.08, 10.5.4.7	
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
tonnpi	?	
digits	rpda	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SMSCBdata(b1, b2, b3, b4: OCTETSTRING)	
Structured Type:	SMSCBpack	
Derivation Path:		
Comments:	SMS cell broadcasting packing data	
Element Name	**Element Value**	**Comments**
block1	b1	
block2	b2	
block3	b3	
block4	b4	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Tppid_01	
Structured Type:	TPPID	
Derivation Path:		
Comments:	TP protocol identifier,GSM 03.40, 9.2.3.9 default value 0	
Element Name	**Element Value**	**Comments**
type	'00'B	
value	'000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Tppid_02(smtype: INTEGER)	
Structured Type:	TPPID	
Derivation Path:		
Comments:	TP protocol identifier,GSM 03.40, 9.2.3.9 default value 0	
Element Name	**Element Value**	**Comments**
type	'01'B	
value	INT_TO_BIT(smtype, 6)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Tpdcs_01	
Structured Type:	TPDCS	
Derivation Path:		
Comments:	SMS data coding scheme, GSM 03.38, 4, 5 default value is 0	
Element Name	**Element Value**	**Comments**
cg	'0000'B	
code	'0000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Tpdcs_02	
Structured Type:	TPDCS	
Derivation Path:		
Comments:	SMS data coding scheme, GSM 03.38, 4, 5 class 2	
Element Name	**Element Value**	**Comments**
cg	'1111'B	
code	'0010'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Tpdcs_03	
Structured Type:	TPDCS	
Derivation Path:		
Comments:	SMS data coding scheme, GSM 03.38, 4, 5 class 1	
Element Name	**Element Value**	**Comments**
cg	'1111'B	
code	'0001'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Tpdcs_04	
Structured Type:	TPDCS	
Derivation Path:		
Comments:	SMS data coding scheme, GSM 03.38, 4, 5 class 0	
Element Name	**Element Value**	**Comments**
cg	'1111'B	
code	'0000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Tpdcs_05	
Structured Type:	TPDCS	
Derivation Path:		
Comments:	SMS data coding scheme, GSM 03.38, 4, 5 default alphabet, english	
Element Name	**Element Value**	**Comments**
cg	'0000'B	
code	'0001'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TpStatus_01	
Structured Type:	TPST	
Derivation Path:		
Comments:	Short message received by the SME	
Element Name	**Element Value**	**Comments**
sprb1	'0'B	
value	'0000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TpCommand_01(tpmr: MR)	
Structured Type:	SMCMD	
Derivation Path:		
Comments:	Enquiry related to previously submitted short message	
Element Name	**Element Value**	**Comments**
sprb1	'00'B	
srr	'0'B	
sprb2	'000'B	
mti	'10'B	
mr	OC_IntToOct(OC_OctToInt(tpmr) + 1, 1)	
pid	Tppid_01	
ct	'00'O	
mn	?	
da	?	
cdl	?	
cd	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TpCommand_02(tpmr: MR)	
Structured Type:	SMCMD	
Derivation Path:		
Comments:	Delete previously submitted short message	
Element Name	**Element Value**	**Comments**
sprb1	'00'B	
srr	'0'B	
sprb2	'000'B	
mti	'10'B	
mr	OC_IntToOct(OC_OctToInt(tpmr) + 1, 1)	
pid	Tppid_01	
ct	'02'O	
mn	?	
da	?	
cdl	?	
cd	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	UuInfo_omit	
Structured Type:	UU	
Derivation Path:		
Comments:	Omitted UUInfo.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
uupd	-	
uui	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	UuInfo	
Structured Type:	UU	
Derivation Path:		
Comments:	Delete previously submitted short message	
Element Name	**Element Value**	**Comments**
iei	'01111110'B	
iel	OC_IntToOct(1+ LENGTH_OF(TSPX_UuInfo), 1)	
uupd	TSPX_Uupd	
uui	TSPX_UuInfo	
Detailed Comments:		

ASN1 type constraints

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CF all synchronous services

Constraint Value

```
activateSSComponents      activateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
12,
                                                                      ss_ForBS

{
ss_Code                           '20'H,
basicService       bearerService      '68'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFU all basic services

Constraint Value

```
activateSSComponents      activateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
12,
                                                                      ss_ForBS

{
ss_Code                           '21'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	BAOC all synchronous services

Constraint Value

```
activateSSComponents      activateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
12,
                                                                      ss_ForBS

{
ss_Code                          '92'H,
basicService          bearerService             '68'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	BIC-Roam

Constraint Value

```
activateSSComponents      activateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
12,
                                                                      ss_ForBS

{
ss_Code                          '9B'H  -- BIC-Roam
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_05
ASN.1 Type:	Component
Derivation Path:	
Comments:	BOIC

Constraint Value

```
activateSSComponents      activateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
12,
                                                                      ss_ForBS

{
ss_Code                          '93'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_06
ASN.1 Type:	Component
Derivation Path:	
Comments:	BAIC
Constraint Value	

activateSSComponents activateSS_InvokeComp
{
invokeID
?,
localValue
12,
ss_ForBS
{
ss_Code '9A'H
}
}	

Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	BldMptySS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	build multiparty request
Constraint Value	

buildMPTYComponents buildMPTY_InvokeComp
{
invokeID
?,
localValue
124
}	

Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	Comps_01(comp:Component)
ASN.1 Type:	Components
Derivation Path:	
Comments:	A received FIE may contain several components, but at least contains "comp".
Constraint Value	
SUPERSET({ comp})	
Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFC for speech

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '28'H,
basicService    teleservice    '10'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRc for all facsimile

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '2B'H,
basicService    teleservice    '60'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	deactivation for barring

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '90'H,
basicService   teleservice   '10'H
}
                                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	deactivation for barring of outging calls

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '91'H,
basicService   teleservice   '60'H
}
                                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_05
ASN.1 Type:	Component
Derivation Path:	
Comments:	deactivation for barring of incoming calls

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '99'H
}
                                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_06
ASN.1 Type:	Component
Derivation Path:	
Comments:	deactivation for BOICExHC

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '94'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	EraseSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFC for all facsimile

Constraint Value

```
eraseSSComponents      eraseSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
11,
                                                                      ss_ForBS

{
ss_Code       '28'H,
basicService   teleservice   '60'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	EraseSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRc for all basic services

Constraint Value

```
eraseSSComponents      eraseSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
11,
                                                                      ss_ForBS

{
ss_Code       '2B'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	EraseSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFU for speech

Constraint Value

```
eraseSSComponents     eraseSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
11,
                                                                      ss_ForBS

{
ss_Code      '21'H,
basicService   teleservice   '10'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	EraseSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRy for all facsimile

Constraint Value

```
eraseSSComponents     eraseSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
11,
                                                                      ss_ForBS

{
ss_Code      '2A'H,
basicService   teleservice   '60'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	FwdChAdvRslt_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
forwardChargeAdviceComponents     forwardChargeAdvice_ReturnResultComp
            {
                                                                      invokeID
0,
                                                                      result
*

               }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	GetPasswdRslt_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	
Constraint Value	
`getPasswordComponents getPassword_ReturnResultComp`	
`{`	
`invokeID`	
`0,`	
`result`	
`{`	
`localValue 18,`	
`currentPassword "1234"`	
`}`	
`}`	
Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	GetPasswdRslt_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	
Constraint Value	
`getPasswordComponents getPassword_ReturnResultComp`	
`{`	
`invokeID`	
`0,`	
`result`	
`{`	
`localValue 18,`	
`currentPassword "9876"`	
`}`	
`}`	
Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	GetPasswdRslt_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	
Constraint Value	
`getPasswordComponents getPassword_ReturnResultComp`	
`{`	
`invokeID`	
`0,`	
`result`	
`{`	
`localValue 18,`	
`currentPassword "9877"`	
`}`	
`}`	
Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFB for all basic services

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp       {
    invokeID                      ?,
    localValue                    14,
    ss_ForBS     {

ss_Code      '29'H
                }
                                                            }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRy for Speech

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
14,
                                                                  ss_ForBS

{
ss_Code      '2A'H, --CNFRy
basicService   teleservice   '10'H
}
                                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRc for all basic services

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
14,
                                                                  ss_ForBS

{
ss_Code      '2B'H
}
                                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFB for all facsimile

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
14,
                                                                      ss_ForBS

{
ss_Code      '29'H,
basicService   teleservice   '60'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_05
ASN.1 Type:	Component
Derivation Path:	
Comments:	BICRoam

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
14,
                                                                      ss_ForBS

{
ss_Code      '9B'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_06
ASN.1 Type:	Component
Derivation Path:	
Comments:	BOIC

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
14,
                                                                      ss_ForBS

{
ss_Code      '93'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_07
ASN.1 Type:	Component
Derivation Path:	
Comments:	BAIC

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
14,
                                                                      ss_ForBS

{

ss_Code      '9A'H

}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_08
ASN.1 Type:	Component
Derivation Path:	
Comments:	BOICExHC

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
14,
                                                                      ss_ForBS

{

ss_Code      '94'H

}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	NotificationSS_07(Invkid: OCTETSTRING)
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
unstructuredSSNotifyComponents     unstructuredSSNotify_ReturnResultComp
                                                                    {
                                                                      invokeID
OC_OctToInvokeIDType(Invkid),
                                                                      result
                                                                              {

localValue                    61
                                                                              }
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	NotificationSS_09(Invkid: OCTETSTRING)
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
unstructuredSSNotifyComponents     unstructuredSSNotify_ReturnErrorComp

                                                          errorCodes
                                                                    {
                                                                       invokeID
OC_OctToInvokeIDType(Invkid),
                                                                       errorCode
ussd_Busy
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegisterSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRy Speech

Constraint Value

```
registerSSComponents     registerSS_InvokeComp   {
     invokeID                    ?,
     localValue                 10,
registerSS_Arg  {

ss_Code                        '2A'H,

basicService    teleservice    '10'H,

forwardedToNumber         '8100342143'H,
                                       -- Unknown Number/ISDN telephony + '00431234'

noReplyConditionTime       5

}
                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegisterSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFU

Constraint Value

```
registerSSComponents     registerSS_InvokeComp      {
   invokeID                    ?,
   localValue                 10,
   registerSS_Arg  {

ss_Code                        '21'H,

basicService    teleservice    '60'H,

forwardedToNumber         '8100342143'H
                                       -- Unknown Number/ISDN telephony + '00431234'

                }
                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegisterSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFB

Constraint Value

```
registerSSComponents     registerSS_InvokeComp     {
       invokeID                    ?,
       localValue                 10,
       registerSS_Arg     {

ss_Code                      '29'H,

basicService    bearerService      '60'H,

forwardedToNumber          '8100342143'H
                                    -- Unknown Number/ISDN telephony + '00431234'
                            }
                                                  }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegisterSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	CF for all facsimile

Constraint Value

```
registerSSComponents     registerSS_InvokeComp     {
     invokeID                    ?,
     localValue                 10,
     registerSS_Arg      {

ss_Code                           '20'H,

basicService     teleservice      '60'H,

forwardedToNumber           '8100342143'H
                                       -- Unknown Number/ISDN telephony + '00431234'
                           }
                                                  }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegPasswdSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	All call restriction services.

Constraint Value

```
registerPasswordComponents     registerPassword_InvokeComp
                                                                                {
                                                                                  invokeID

?,
                                                                                  localValue

17,
                                                                                  ss_Code

'90'H
                                                                                   }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ProcessUSSData_01(ussdString: IA5String)
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
processUnstructuredSSDataComponents     processUnstructuredSSData_InvokeComp
                                                                          {
                                                                            invokeID
?,
                                                                            localValue
19,
                                                                            ss_UserData
ussdString
                                                                          }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ProcessUSSDReq_01(ussdString: IA5String)
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
processUnstructuredSSRequestComponents     processUnstructuredSSRequest_InvokeComp
                                                                          {
                                                                            invokeID
?,
                                                                            localValue
59,
                                                                            ussd_Arg

{
ussd_DataCodingScheme      'F0'O,
ussd_String            OC_CodingOfUssdString(ussdString)
}
                                                                           }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	USSDReq_04(Invkid :OCTETSTRING; ussdString: IA5String)
ASN.1 Type:	Component
Derivation Path:	
Comments:	ReturnResult for Unstructured SS request

Constraint Value

```
unstructuredSSRequestComponents     unstructuredSSRequest_ReturnResultComp
                                                                          {
                                                                            invokeID
OC_OctToInvokeIDType(Invkid),
                                                                            result
                                                                                     {
localValue                     60,
ussd_Res
{
ussd_DataCodingScheme      'F0'O,
ussd_String OC_CodingOfUssdString(ussdString)
}
                                                                                     }
                                                                           }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	USSDReq_05(Invkid :OCTETSTRING)
ASN.1 Type:	Component
Derivation Path:	
Comments:	Return Error for UnstructuredSS-Request with the error code USSD Busy
Constraint Value	
`unstructuredSSRequestComponents unstructuredSSRequest_ReturnErrorComp errorCodes { invokeID OC_OctToInvokeIDType(Invkid), errorCode ussd_Busy }`	
Detailed Comments:	

ASP constraint declarations

TTCN ASP constraint declarations

ASP Constraint Declaration		
Constraint Name:	Abort(ch: LOGICCH; pdu:ABRT_PDU)	
ASP Type:	DL_DatRqAbrt	
Derivation Path:		
Comments:	To send an Abort message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AlertRcv(pdu: ALERT_PDU)	
ASP Type:	DL_DatInAlert	
Derivation Path:		
Comments:	To receive an ALERTING message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AlertSnd(ch:LOGICCH; pdu: ALERT_PDU)	
ASP Type:	DL_DatRqAlert	
Derivation Path:		
Comments:	To send an ALERTING message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AssCmd(ch:LOGICCH; pdu: ASS_CMD_PDU)	
ASP Type:	DL_DatRqAssCmd	
Derivation Path:		
Comments:	To send an ASSIGNMENT COMMAND message which is assigned in send statement.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AssCmp(ch:LOGICCH; pdu:ASS_COM_PDU)	
ASP Type:	DL_DatInAssCom	
Derivation Path:		
Comments:	To match any received ASSIGNMENT COMPLETE message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AssFl_02(ch:LOGICCH)	
ASP Type:	DL_DatInAssfl	
Derivation Path:		
Comments:	protocol error undefined	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	AssgnFl_02	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AssFl_any_cau(ch: LOGICCH)	
ASP Type:	DL_DatInAssfl	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	AssgnFl_01	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AuthReq(ch: LOGICCH; pdu: AUTH_RQ_PDU)	
ASP Type:	DL_DatRqAuthRq	
Derivation Path:		
Comments:	To send an AUTHENTICATION REQUEST message with default values.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AuthReq_inv_01(ch: LOGICCH; rand:RAND)	
ASP Type:	DL_DatRqAuthRq	
Derivation Path:		
Comments:	To send an AUTHENTICATION REQUEST message containing arbitrary spare bits	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	AuthRequest_inv_01(rand)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AuthRejSnd(ch: LOGICCH; pdu:AUTH_REJ_PDU)	
ASP Type:	DL_DatRqAuthRej	
Derivation Path:		
Comments:	To send an AUTHENTICATION REJECT message with default values.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AuthRes(pdu: AUTH_RES_PDU)	
ASP Type:	DL_DatInAuthRes	
Derivation Path:		
Comments:	To match any received AUTHENTICATION RESPONSE message which contains any SRES.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CallCfm(pdu: CALL_CO_PDU)	
ASP Type:	DL_DatInCallCo	
Derivation Path:		
Comments:	To receive a CC CALL CONFIRMED message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CallProc(ch: LOGICCH; pdu: CALL_PROC_PDU)	
ASP Type:	DL_DatRqCallProc	
Derivation Path:		
Comments:	To send a CALL PROCEEDING message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CCStRcv(ch: LOGICCH; pdu:CCST_PDU)	
ASP Type:	DL_DatInCcst	
Derivation Path:		
Comments:	To match any received STATUS message with TI = Ti.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CCStSnd(ch: LOGICCH; pdu:CCST_PDU)	
ASP Type:	DL_DatRqCcst	
Derivation Path:		
Comments:	To send a CC STATUS message without mandatory cause IE and call state IE.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CCStESnd(ch: LOGICCH; pdu: CCST_ENQ_PDU)	
ASP Type:	DL_DatRqCcstEnq	
Derivation Path:		
Comments:	To send a STATUS ENQUIRY message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChmmoAckRcv(ch:LOGICCH; msg:CHMMO_ACK_PDU)	
ASP Type:	DL_DatInChmmoAck	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChmmoReqSnd(ch:LOGICCH; msg:CHMMO_PDU)	
ASP Type:	DL_DatRqChmmo	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChRel(ch: LOGICCH; pdu: CH_REL_PDU)	
ASP Type:	DL_DatRqChRel	
Derivation Path:		
Comments:	To send a CHANNEL RELEASE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChRel_inv(ch:LOGICCH)	
ASP Type:	DL_DatRqChRelErr	
Derivation Path:		
Comments:	To send a CHANNEL RELEASE message containing additional IE unknown in the RR protocol	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	ChRelease_inv_03	
Detailed Comments:	used in TC_26_5_6_3	

ASP Constraint Declaration		
Constraint Name:	ChReq(pdu: CH_RQ_PDU)	
ASP Type:	DL_RacInChRq	
Derivation Path:		
Comments:	To receive a primitive containing a CHANNEL REQUEST message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
fn	?	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChReq_01(ch:LOGICCH; pdu:CH_RQ_PDU)	
ASP Type:	DL_RacInChRq	
Derivation Path:		
Comments:	To receive a CHANNEL REQUEST message in cell B	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
fn	?	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ClassChgURcv(pdu:CLM_CHN_PDU)	
ASP Type:	DL_UdatInClmChn	
Derivation Path:		
Comments:	to match a received CLASSMARK CHANGE message containing classmark2 indicating original rf power class	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ClassChgDRcv(pdu:CLM_CHN_PDU)	
ASP Type:	DL_DatInClmChn	
Derivation Path:		
Comments:	to match a received CLASSMARK CHANGE message containing classmark2 indicating original rf power class and possible classmark3.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ClassMkEnq_01(ch: LOGICCH)	
ASP Type:	DL_DatRqClmEnq	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	ClassMarkEnq_01	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CmreReq_02(mi:MI; lai:LAI; cksn:B_3)	
ASP Type:	DL_EstInCmreRq	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
establish_mode	?	
msg	CMReEstReq_02(mi, lai, cksn)	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CMSerAcp(ch:LOGICCH; pdu: CMS_ACP_PDU)	
ASP Type:	DL_DatRqCmsAcp	
Derivation Path:		
Comments:	To send a CM service accept message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CMSerRej(ch:LOGICCH; pdu: CMS_REJ_PDU)	
ASP Type:	DL_DatRqCmsRej	
Derivation Path:		
Comments:	reject cause = "service or option not available, unspecified"	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CMSerReq(pdu: CMS_RQ_PDU)	
ASP Type:	DL_EstInCmsRq	
Derivation Path:		
Comments:	To receive a CM SERVICE REQUEST message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CMSerDatReq(pdu:CMS_RQ_PDU)	
ASP Type:	DL_DatInCmsRq	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ConnAckRcv(pdu: CONN_ACK_PDU)	
ASP Type:	DL_DatInConnAck	
Derivation Path:		
Comments:	To receive a CONNECT ACKNOWLEDGE message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ConnAckSnd(ch:LOGICCH; pdu: CONN_ACK_PDU)	
ASP Type:	DL_DatRqConnAck	
Derivation Path:		
Comments:	To send a CONNECT ACKNOWLEDGE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	Conn_invSnd(ch:LOGICCH; pdu:CONN_PDU_ERR)	
ASP Type:	DL_DatRqConnErr	
Derivation Path:		
Comments:	To send a CC CONNECT message containing a mandatory IE coded as comprehension required.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ConnRcv(pdu: CONN_PDU)	
ASP Type:	DL_DatInConn	
Derivation Path:		
Comments:	To receive a CC CONNECT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ConnSnd(ch: LOGICCH; pdu: CONN_PDU)	
ASP Type:	DL_DatRqConn	
Derivation Path:		
Comments:	To send a CC CONNECT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CphCmd(ch: LOGICCH; pdu: CPHM_CMD_PDU)	
ASP Type:	DL_DatRqCphmCmd	
Derivation Path:		
Comments:	To send a CIPHERING MODE COMMAND message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CphCmd_inv(ch:LOGICCH; pdu:CPHM_CMD_PDU_ERR)	
ASP Type:	DL_DatRqCphmCmdErr	
Derivation Path:		
Comments:	To send a CIPHERING MODE COMMAND message containing additional unknown IE	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	CphModeCmd_inv_03	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CphCom(pdu: CPHM_COM_PDU)	
ASP Type:	DL_DatInCphmCom	
Derivation Path:		
Comments:	To receive a CIPHERING MODE COMPLETE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	Disc_inv(ch: LOGICCH; pdu:DISC_PDU)	
ASP Type:	DL_DatRqDisc	
Derivation Path:		
Comments:	To send a DISCONNECT message containing the transaction ID not refer to the active call.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	Disc_inv_err(ch:LOGICCH; pdu:DISC_PDU_ERR)	
ASP Type:	DL_DatRqDiscErr	
Derivation Path:		
Comments:	To send a DISCONNECT message containing unknown IEI	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DiscRcv(ch:LOGICCH; pdu: DISC_PDU)	
ASP Type:	DL_DatInDisc	
Derivation Path:		
Comments:	To match a received DISCONNECT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DiscSnd(ch:LOGICCH; pdu: DISC_PDU)	
ASP Type:	DL_DatRqDisc	
Derivation Path:		
Comments:	To send a DISCONNECT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLEstCo(ch:LOGICCH)	
ASP Type:	DL_EstCo	
Derivation Path:		
Comments:	The ASP is used by the L2 to inform the L3 about the establishment of multiple frame link (L2 -> L3).	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap3	
logic_ch	ch	
establish_mode	C_Norm	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLEstRq(ch:LOGICCH)	
ASP Type:	DL_EstRq	
Derivation Path:		
Comments:	Request of a layer 2 connection establishment	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap3	
logic_ch	ch	
establish_mode	C_Norm	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLEstInd(ch :LOGICCH)	
ASP Type:	DL_EstIn	
Derivation Path:		
Comments:	Indication of a layer 2 connection establishment	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
establish_mode	?	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLEstInd_2(ch:LOGICCH)	
ASP Type:	DL_EstIn	
Derivation Path:		
Comments:	Indication of a layer 2 connection establishment for SAPI 3	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap3	
logic_ch	ch	
establish_mode	?	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLRelInd_01	
ASP Type:	DL_RelIn	
Derivation Path:		
Comments:	Layer 2 indication of the layer 2 connection has been released.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
release_mode	?	
outstanding_indicator	?	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLRelInd_02	
ASP Type:	DL_RelIn	
Derivation Path:		
Comments:	Layer 2 indication of the layer 2 connection has been released.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
release_mode	C_LocEndRel	
outstanding_indicator	?	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ESetupRcv(pdu: ESETUP_PDU)	
ASP Type:	DL_DatInESetup	
Derivation Path:		
Comments:	To receive an emergency call SETUP message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	FacilityRcv(fac:FAC_PDU)	
ASP Type:	DL_DatInFac	
Derivation Path:		
Comments:	To receive a FACILITY message containing build multiparty request	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	fac	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	FacilitySnd(ch:LOGICCH; fac:FAC_PDU)	
ASP Type:	DL_DatRqFac	
Derivation Path:		
Comments:	To send the FACILITY message passed by 'fac'	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	fac	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	FrqRedfSnd(ch:LOGICCH; pdu:FRQRE_PDU)	
ASP Type:	DL_DatRqFrqre	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvSnd(ch:LOGICCH; msg:HO_CMD_PDU)	
ASP Type:	DL_DatRqHoCmd	
Derivation Path:		
Comments:	To send a HANDOVER COMMAND indicating a finely synchronised intra cell handover.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvAcc_01	
ASP Type:	DL_RacInHoacc	
Derivation Path:		
Comments:	To received any HANDOVER ACCESS message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	HandOverAcc_01	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvAccRcv(ch: LOGICCH; msg:HOACC_PDU)	
ASP Type:	DL_RacInHoacc	
Derivation Path:		
Comments:	To received any HANDOVER ACCESS message on channel `ch`.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvFlRcv(ch:LOGICCH; msg :HOFL_PDU)	
ASP Type:	DL_DatInHofl	
Derivation Path:		
Comments:	To match a received HANDOVER FAILURE message containing any RR cause.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvCmpRcv(ch:LOGICCH; msg:HO_COM_PDU)	
ASP Type:	DL_DatInHoCom	
Derivation Path:		
Comments:	To receive any HANDOVER COMPLETE message on channel ch.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HoldRcv(pdu:HOLD_PDU)	
ASP Type:	DL_DatInHold	
Derivation Path:		
Comments:	To receive any HOLD message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HoldAckSnd(ch:LOGICCH; pdu:HOLD_ACK_PDU)	
ASP Type:	DL_DatRqHoldAck	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	IDReq(ch:LOGICCH; pdu: ID_RQ_PDU)	
ASP Type:	DL_DatRqIdRq	
Derivation Path:		
Comments:	To send IDENTITY REQUEST message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	IDRes(pdu: ID_RES_PDU)	
ASP Type:	DL_DatInIdRes	
Derivation Path:		
Comments:	To receive a IDENTITY RESPONSE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ImmAss(ch: LOGICCH; pdu: IMMASS_PDU)	
ASP Type:	DL_UdatRqImmass	
Derivation Path:		
Comments:	To send an IMMEDIATE ASSIGNMENT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ImmAssSp(ch:LOGICCH; pgg:PGG; msg:IMMASS_PDU)	
ASP Type:	DL_UdatRqImmass_sp	
Derivation Path:		
Comments:	To send an IMMEDIATE ASSIGNMENT message on the MS paging channel.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
pgg	pgg	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ImmAssX(ch:LOGICCH; pdu:IMMASSX_PDU)	
ASP Type:	DL_UdatRqImmassx	
Derivation Path:		
Comments:	To send an IMMEDIATE ASSIGNMENT EXTENDED message which assigns the SDCCH/4 channel for the MS	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:	The request reference (Rr, Fn) is used for the MS1, whilst (Rr_9, Fn_9) for MS2.	

ASP Constraint Declaration		
Constraint Name:	ImmAssXSp(ch:LOGICCH; pgg:PGG; msg:IMMASSX_PDU)	
ASP Type:	DL_UdatRqImmassx_sp	
Derivation Path:		
Comments:	To send an IMMEDIATE ASSIGNMENT EXTENDED message on the MS paging channel.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
pgg	pgg	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ImmAssRej(ch:LOGICCH; pdu:IMMASS_REJ_PDU)	
ASP Type:	DL_UdatRqImmassRej	
Derivation Path:		
Comments:	To send an IMMEDIATE ASSIGNMENT REJECT message containing normal paging mode and wait indication = 0 seconds.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ImmAssRejSp(ch:LOGICCH; pgg:PGG; msg:IMMASS_REJ_PDU)	
ASP Type:	DL_UdatRqImmassRej_sp	
Derivation Path:		
Comments:	To send an IMMEDIATE ASSIGNMENT REJECT message on the MS paging channel.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
pgg	pgg	
msg	msg	
Detailed Comments:	Only used in TC_26_6_2_4	

ASP Constraint Declaration		
Constraint Name:	ImsiDet(pdu:IMSID_IN_PDU)	
ASP Type:	DL_EstInImsidIn	
Derivation Path:		
Comments:	To match a received IMSI DETACH INDICATION message matching any MS classmark1 value and any mobile identity value	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	LocAcp(ch:LOGICCH; pdu:LUP_ACP_PDU)	
ASP Type:	DL_DatRqLupAcp	
Derivation Path:		
Comments:	To send a LOCATION UPDATING ACCEPT message with a new mobile identity TMSI and location area.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	LocAcp_inv(ch: LOGICCH; mi1, mi2:MI; mcc, mnc, lac:OCTETSTRING)	
ASP Type:	DL_DatRqLupAcpErr	
Derivation Path:		
Comments:	To send a LOCATION UPDATING ACCEPT message with duplicated mobile identities.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	LocUpdtAcp_inv_01(mi1, mi2, mcc, mnc, lac)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	LocUp(ch: LOGICCH; pdu:LUP_RQ_PDU)	
ASP Type:	DL_EstInLupRq	
Derivation Path:		
Comments:	To match a received LOCATION UPDATING REQUEST message containing location updating type = IMSI attach.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	LocRej(ch: LOGICCH; pdu:LUP_REJ_PDU)	
ASP Type:	DL_DatRqLupRej	
Derivation Path:		
Comments:	To send a LOCATION UPDATING REJECT message containing the reject cause IMSI unknown in HLR. Used var's: ch	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	MDLRelReq(ch :LOGICCH)	
ASP Type:	MDL_RelRq	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
release_mode	C_LocEndRel	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ModifySnd(ch:LOGICCH; msg:MODIFY_PDU)	
ASP Type:	DL_DatRqModify	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ModifyRcv(msg:MODIFY_PDU)	
ASP Type:	DL_DatInModify	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ModifyCmpSnd(ch:LOGICCH; msg:MODIFY_COM_PDU)	
ASP Type:	DL_DatRqModifyCom	
Derivation Path:		
Comments:	n -> ms	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ModifyRejRqSnd(ch:LOGICCH; msg:MODIFY_REJ_PDU)	
ASP Type:	DL_DatRqModifyRej	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ModifyRejRcv(ch:LOGICCH; msg:MODIFY_REJ_PDU)	
ASP Type:	DL_DatInModifyRej	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	MMSt(pdu:MMST_PDU)	
ASP Type:	DL_DatInMmst	
Derivation Path:		
Comments:	To receive a MM STATUS message containing reject cause value #97.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	MsrRept(pdu:MSR_RPT_PDU)	
ASP Type:	DL_UdatInMsrRpt	
Derivation Path:		
Comments:	To receive a measurement report without measurement results.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	NotifySnd(ch:LOGICCH; msg:NOTIFY_PDU)	
ASP Type:	DL_DatRqNotify	
Derivation Path:		
Comments:	To send a NOTIFY message to the MS. The message contains any valid notification indicator.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PgReq1(ch:LOGICCH; pgg:PGG; pg1_req_pdu: PG1_RQ_PDU)	
ASP Type:	DL_UdatRqPg1Rq	
Derivation Path:		
Comments:	To send a PAGING REQUEST TYPE 1 message to the paging group indicated by the parameter pgg which is derived from system parameters. The PAGING REQUEST TYPE1 message requests normal paging mode, any channel and for the MS identity TSPX_TMSI.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
pgg	pgg	
msg	pg1_req_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PgReq2(ch:LOGICCH; pgg:PGG; pg2_req_pdu: PG2_RQ_PDU)	
ASP Type:	DL_UdatRqPg2Rq	
Derivation Path:		
Comments:	To send a PAGING REQUEST TYPE 1 message to the paging group indicated by the parameter pgg which is derived from system parameters. The PAGING REQUEST TYPE1 message requests normal paging mode, any channel and for the MS identity TSPX_TMSI.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
pgg	pgg	
msg	pg2_req_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PgReq3(ch:LOGICCH; pgg:PGG; pg3_req_pdu: PG3_RQ_PDU)	
ASP Type:	DL_UdatRqPg3Rq	
Derivation Path:		
Comments:	To send a PAGING REQUEST TYPE 1 message to the paging group indicated by the parameter pgg which is derived from system parameters. The PAGING REQUEST TYPE1 message requests normal paging mode, any channel and for the MS identity TSPX_TMSI.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
pgg	pgg	
msg	pg3_req_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PagingRes(pdu: PG_RES_PDU)	
ASP Type:	DL_EstInPgRes	
Derivation Path:		
Comments:	To receive a PAGING RESPONSE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PgRes(ch:LOGICCH; pdu:PG_RES_PDU)	
ASP Type:	DL_EstInPgRes	
Derivation Path:		
Comments:	To match any received PAGING RESPONSE message on the channel `ch`.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PhyInfo_01(ch: LOGICCH; ta:TA)	
ASP Type:	DL_DatRqPhyinfo	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	Phyinfor_01(ta)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	Progress(ch:LOGICCH; pdu: PROG_PDU)	
ASP Type:	DL_DatRqProg	
Derivation Path:		
Comments:	To send a PROGRESS message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	Register_01(ch:LOGICCH; reg:REGISTER_PDU)	
ASP Type:	DL_DatInRegister	
Derivation Path:		
Comments:	To receive any Register message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	reg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	Register_03(reg:REGISTER_PDU)	
ASP Type:	DL_DatInRegister	
Derivation Path:		
Comments:	CFNRy	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	reg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	RegisterReq(ch: LOGICCH; reg:REGISTER_PDU)	
ASP Type:	DL_DatRqRegister	
Derivation Path:		
Comments:	To send a REGISTER message containing Invoke for UnstructuredSS-Notify	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	reg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ReleaseRcv(pdu: REL_PDU)	
ASP Type:	DL_DatInRel	
Derivation Path:		
Comments:	To receive a CC RELEASE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ReleaseSnd(ch: LOGICCH; pdu: REL_PDU)	
ASP Type:	DL_DatRqRel	
Derivation Path:		
Comments:	To send a RELEASE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	RelComRcv(pdu: REL_COM_PDU)	
ASP Type:	DL_DatInRelCmp	
Derivation Path:		
Comments:	To receive a RELEASE COMPLETE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	RelComSnd(ch:LOGICCH; pdu: REL_COM_PDU)	
ASP Type:	DL_DatRqRelCmp	
Derivation Path:		
Comments:	To send a RELEASE COMPLETE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	RrStatusRcv(ch:LOGICCH; pdu:RRST_PDU)	
ASP Type:	DL_DatInRrst	
Derivation Path:		
Comments:	To match a received RR STATUS message containing any RR cause.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SMSCBReq(ch:LOGICCH; pdu:SMSCB_PDU)	
ASP Type:	DL_UdatRqSMSCBData	
Derivation Path:		
Comments:	To send a SMSCB message, first block	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap3	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SetupRcv(pdu: SETUP_MO_PDU)	
ASP Type:	DL_DatInSetup	
Derivation Path:		
Comments:	To receive a SETUP message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SetupSnd(ch: LOGICCH; setup:SETUP_MT_PDU)	
ASP Type:	DL_DatRqSetup	
Derivation Path:		
Comments:	To send a SETUP message which is the input parameter of this ASP.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	setup	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	StartDTMFRcv(msg:START_DTMF_PDU)	
ASP Type:	DL_DatInStartDtmf	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	StartDTMFAckSnd(ch:LOGICCH; msg:START_DTMF_ACK_PDU)	
ASP Type:	DL_DatRqStartDtmfAck	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	StartDTMFRejSnd(ch: LOGICCH; msg:START_DTMF_REJ_PDU)	
ASP Type:	DL_DatRqStartDtmfRej	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	StopDTMFRcv(msg:STOP_DTMF_PDU)	
ASP Type:	DL_DatInStopDtmf	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	StopDTMFAckSnd(ch: LOGICCH; msg:STOP_DTMF_ACK_PDU)	
ASP Type:	DL_DatRqStopDtmfAck	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SyncInfo(ch:LOGICCH; bcc:BCC; ncc:NCC)	
ASP Type:	DL_UdatRqSchinfo	
Derivation Path:		
Comments:	To send SYNCHRONIZATION INFORMATION message with default parameters.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SyncInfor(bcc, ncc)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo1(ch:LOGICCH; cchd:CCHD; maxtx:B_2; txint:B_4; re:B_1)	
ASP Type:	DL_UdatRqSysinfo1	
Derivation Path:		
Comments:	To send SYSTEM INFORMATION TYPE 1 in cell A for RR testing of GSM900.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf1(cchd, maxtx, txint, re)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo1_nh(ch:LOGICCH; ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
ASP Type:	DL_UdatRqSysinfo1_nh	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf3(ci, mcc, mnc, lac, ccd, co, crh, mtmc, neci, maxtx, txint, re)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo2(ch: LOGICCH; bcchfl: NCD; maxtx:B_2; txint:B_4; re:B_1; nccp:NCCP)	
ASP Type:	DL_UdatRqSysinfo2	
Derivation Path:		
Comments:	To send the default SYSTEM INFORMATION TYPE 2 message containing default neighbour cells description in cell B for GSM900.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf2(bcchfl, maxtx, txint, re, nccp)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo2bis(ch: LOGICCH; bcchflegsm: NCD)	
ASP Type:	DL_UdatRqSysinfo2bis	
Derivation Path:		
Comments:	SYSTEM INFORMATION 2bis in cell A under EGSM with the ARFCN list = {988, 990, 1003}.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf2bis(bcchflegsm)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo3(ch:LOGICCH; ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
ASP Type:	DL_UdatRqSysinfo3	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf3(ci, mcc, mnc, lac, ccd, co, crh, mtmc, neci, maxtx, txint, re)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo3_inv_01(mcc, mnc, lac:OCTETSTRING)	
ASP Type:	DL_UdatRqSysinfo3	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION message containing new location area and rest octets which are not all '2B'O	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	C_BCCH_A_1	
msg	SysInf3_inv_01(mcc, mnc, lac)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo4(ch:LOGICCH; mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
ASP Type:	DL_UdatRqSysinfo4	
Derivation Path:		
Comments:	To send a default SYSTEM INFORMATION TYPE 4 message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf4(mcc, mnc, lac, crh, mtmc, neci, maxtx, txint, re)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo4_CBMS(ch:LOGICCH; mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; cbchd:CHD)	
ASP Type:	DL_UdatRqSysinfo4	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 4 message for SMSCB with the values or GSM 11.10, 34.3	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf4_CBMS(mcc, mnc, lac, crh, mtmc, neci, maxtx,txint, re, cbchd)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo4_inv_01(mcc, mnc, lac:OCTETSTRING)	
ASP Type:	DL_UdatRqSysinfo4	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION message containing rest octets which are not all '2B'O	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	C_BCCH_A_1	
msg	SysInf4_inv_01(mcc, mnc, lac)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo5(sacch: LOGICCH; sysinfo5_pdu:SYSINFO5_PDU)	
ASP Type:	DL_UdatRqSysinfo5	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 5 message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	sacch	
msg	sysinfo5_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo5bis(sacch: LOGICCH; sysinfo5bis_pdu:SYSINFO5bis_PDU)	
ASP Type:	DL_UdatRqSysinfo5bis	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 5bis message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	sacch	
msg	sysinfo5bis_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo6(sacch: LOGICCH; sysinfo6_pdu:SYSINFO6_PDU)	
ASP Type:	DL_UdatRqSysinfo6	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 6 message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	sacch	
msg	sysinfo6_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	TmsiReallocSnd(par:MI; ch:LOGICCH; mcc, mnc, lac:OCTETSTRING)	
ASP Type:	DL_DatRqTmsireCmd	
Derivation Path:		
Comments:	To send a TMSI REALLOCATION COMMAND message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	TmsiReallocCmd(par, mcc, mnc, lac)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	TmsiReallocCmp(ch:LOGICCH)	
ASP Type:	DL_DatInTmsireCom	
Derivation Path:		
Comments:	To receive a TMSI REALLOCATION COMPLETE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	TmsiReallocComp	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	UndefCC(ch: LOGICCH; pdu: CONN_PDU)	
ASP Type:	DL_DatRqUndefCC	
Derivation Path:		
Comments:	To send an undefined Layer 3 message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	UndefMM(ch:LOGICCH; pdu:ID_RES_PDU)	
ASP Type:	DL_DatRqUndefMM	
Derivation Path:		
Comments:	To send an undefined MM message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	UndefRR(ch: LOGICCH; pdu:PART_REL_PDU)	
ASP Type:	DL_DatRqUndefRR	
Derivation Path:		
Comments:	To send an undefined RR message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	Unknown(ch: LOGICCH; pdu: CCST_ENQ_PDU)	
ASP Type:	DL_DatRqUnknown	
Derivation Path:		
Comments:	To send an unknown CC message which is coded like a CC STATUS ENQUIRY.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DatInCpData(ch:LOGICCH; message:CP_DATA_PDU)	
ASP Type:	DL_DatInCpData	
Derivation Path:		
Comments:	The ASP is used to indicate the receipt of the SMS CP DATA message using acknowledged operation (L2 -> L3) for MT.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap3	
logic_ch	ch	
msg	message	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DatRqCpData(message:CP_DATA_PDU; ch: LOGICCH)	
ASP Type:	DL_DatRqCpData	
Derivation Path:		
Comments:	ASP to request the transmission of the SMS CP DATA message using acknowlodged operation (L3 -> L2) for MO.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap3	
logic_ch	ch	
msg	message	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DatInCpDataAck(message:CPDATA_ACK_PDU)	
ASP Type:	DL_DatInCpDataAck	
Derivation Path:		
Comments:	The ASP is used to indicate the receipt of the SMS CP DATA message using acknowledged operation (L2 -> L3) .	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap3	
logic_ch	?	
msg	message	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DatRqCpDataAck(message:CPDATA_ACK_PDU; ch: LOGICCH)	
ASP Type:	DL_DatRqCpDataAck	
Derivation Path:		
Comments:	The ASP is used to request the transmission of the SMS CP DATA message using acknowlodged operation (L3 -> L2).	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap3	
logic_ch	ch	
msg	message	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DatRqCpError(message:CPERR_PDU; ch: LOGICCH)	
ASP Type:	DL_DatRqCpError	
Derivation Path:		
Comments:	ASP to request the transmission of the SMS CP ERROR message using acknowledged operation (L3 -> L2) for MO.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap3	
logic_ch	ch	
msg	message	
Detailed Comments:		

PDU constraint declarations

TTCN PDU constraint declarations

PDU Constraint Declaration		
Constraint Name:	Alerting_01(Ti :TI)	
PDU Type:	ALERT_PDU	
Derivation Path:		
Comments:	An ALERTING message containing mandatory IE's only to be sent to the MS.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000001'B	
fie	-	
pi	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Alerting_04(Ti :TI; fie:FIE)	
PDU Type:	ALERT_PDU	
Derivation Path:		
Comments:	An ALERTING message containing facility IE to be sent to the MS.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000001'B	
fie	fie	
pi	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AlertingInd_01(ti:TI)	
PDU Type:	ALERT_PDU	
Derivation Path:		
Comments:	To match an received ALERTING message	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?000001'B	
fie	*	
pi	-	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AlertingInd_02(Ti:TI)	
PDU Type:	ALERT_PDU	
Derivation Path:		
Comments:	To match an received ALERTING message	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?000001'B	
fie	*	
pi	-	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_fh(slot:SN; tsc:TSC; chtype:CH_TDMA; par_pwlvl:INTEGER; maio:MAIO; hsn:HSN; frql:FRQL; Cchd:CCHD; chmod:CHMOD; ma:MA; Cphms:CPHMS)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message for hopping without starting time.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_fh(chtype, slot, tsc, maio, hsn)	
pcmd	Pcmd_19(INT_TO_BIT(par_pwlvl, 5))	
frql_at	frql	
cchd	Cchd	
ch1mod	chmod	
ch2d_at	-	
ch2mod	-	
ma_at	ma	
strt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	Cphms	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_nfh(type:CH_TDMA; slot:SN; tsc:TSC; par_pwlvl, arfcn:INTEGER; cchd:CCHD; chmod:CHMOD; strt:STRT; cphms:CPHMS)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message. The channel mode and type are specified as parameters.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_nfh(type, slot, tsc, arfcn)	
pcmd	Pcmd_19(INT_TO_BIT(par_pwlvl, 5))	
frql_at	-	
cchd	cchd	
ch1mod	chmod	
ch2d_at	-	
ch2mod	-	
ma_at	-	
strt	strt	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	cphms	
Detailed Comments:	used iin CC testing	

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_tchf_fh(slot:SN; tsc:TSC; par_pwlvl:INTEGER; maio:MAIO; hsn:HSN; frql:FRQL; cchd:CCHD; chmod:CHMOD; ma:MA; cphms:CPHMS)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_tchf_fh(slot, tsc, maio, hsn)	
pcmd	Pcmd_19(INT_TO_BIT(par_pwlvl, 5))	
frql_at	frql	
cchd	cchd	
ch1mod	chmod	
ch2d_at	-	
ch2mod	-	
ma_at	ma	
strt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	cphms	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_tchh_fh(subch:B_1; slot:SN; tsc:TSC; par_pwlvl:INTEGER; maio:MAIO; hsn:HSN; frql:FRQL; cchd:CCHD; chmod:CHMOD; ma:MA; cphms:CPHMS)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_tchh_fh(slot, tsc, subch, maio, hsn)	
pcmd	Pcmd_19(INT_TO_BIT(par_pwlvl, 5))	
frql_at	frql	
cchd	cchd	
ch1mod	chmod	
ch2d_at	-	
ch2mod	-	
ma_at	ma	
strt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	cphms	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_tchh_nfh(subch:B_1; slot:SN; tsc:TSC; par_pwlvl, arfcn:INTEGER; cchd:CCHD; chmod:CHMOD; strt:STRT)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_tchh_nfh(slot, tsc, subch, arfcn)	
pcmd	Pcmd_19(INT_TO_BIT(par_pwlvl, 5))	
frql_at	-	
cchd	cchd	
ch1mod	chmod	
ch2d_at	-	
ch2mod	-	
ma_at	-	
strt	strt	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_22_Ae1(ts_ccch: BITSTRING; par_chtype:CH_TDMA; par_flist:OCTETSTRING; par_flistl: LENGTH; n:INTEGER)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message assigning TCH/F non FH channel in cell A for EGSM.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_fh(par_chtype, ts_ccch, TSPX_TscDef, INT_TO_BIT((TSPX_MAIO MOD n),6), INT_TO_BIT(TSPX_HSN, 6))	
pcmd	Pcmd_19('01001'B)	
frql_at	Frql_20_egsm(par_flist, par_flistl)	
cchd	-	
ch1mod	-	
ch2d_at	-	
ch2mod	-	
ma_at	-	
strt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_22_Ae2(ts_ccch: BITSTRING; par_chtype:CH_TDMA; par_cchd: OCTETSTRING; par_ma:BITSTRING; n:INTEGER)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message assigning TCH/F non FH channel in cell A for EGSM.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_fh(par_chtype, ts_ccch, TSPX_TscDef, INT_TO_BIT((TSPX_MAIO MOD n),6), INT_TO_BIT(TSPX_HSN, 6))	
pcmd	Pcmd_19('01001'B)	
frql_at	-	
cchd	CellChDes_20_Be(par_cchd)	
ch1mod	-	
ch2d_at	-	
ch2mod	-	
ma_at	MoblAllc_20_Be1iei(par_ma)	
strt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_22_Ae3(ts_ccch: BITSTRING; par_chtype:CH_TDMA; par_cchd: OCTETSTRING; par_ma1:BITSTRING; par_ma2:BITSTRING; n:INTEGER)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message assigning TCH/F non FH channel in cell A for EGSM.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_fh(par_chtype, ts_ccch, TSPX_TscDef, INT_TO_BIT((TSPX_MAIO MOD n),6), INT_TO_BIT(TSPX_HSN, 6))	
pcmd	Pcmd_19('01001'B)	
frql_at	-	
cchd	CellChDes_20_Be(par_cchd)	
ch1mod	-	
ch2d_at	-	
ch2mod	-	
ma_at	MoblAllc_20_Be2iei(par_ma1, par_ma2)	
strt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_inv_01(slot:SN; tsc:TSC)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message containing invalid skip identifier	
Field Name	**Field Value**	**Comments**
ski	'0100'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_nfh('00001'B, slot, tsc, C_arfcnA)	
pcmd	Pcmd_19('01001'B)	
frql_at	-	
cchd	-	
ch1mod	-	
ch2d_at	-	
ch2mod	-	
ma_at	-	
strt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_r14(slot:SN; tsc:TSC; strt:STRT)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message defined by PIXIT	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_fh(TSPX_Chtp1, slot, tsc, INT_TO_BIT(TSPX_Maio2, 6), INT_TO_BIT(TSPX_Hsn2, 6))	
pcmd	Pcmd_19(INT_TO_BIT(TSPX_PwrlvlA, 5))	
frql_at	-	
cchd	-	
ch1mod	ChMod_r01	
ch2d_at	-	
ch2mod	-	
ma_at	MoblAllc_r06	
strt	strt	
frql_bt	-	
ch1d_bt	ChDescrp_fhiei(TSPX_Chtp1, slot, tsc, INT_TO_BIT(TSPX_Maio3, 6), INT_TO_BIT(TSPX_Hsn3, 6))	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	MoblAllc_r07	
cphms	-	
Detailed Comments:	used in TC_26_6_13_1 only.	

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_r15(slot:SN; tsc:TSC; slot2:SN; tsc2:TSC; strt:STRT)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message defined by PIXIT	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_fh(TSPX_Chtp2, slot, tsc, INT_TO_BIT(TSPX_Maio5, 6), INT_TO_BIT(TSPX_Hsn5, 6))	
pcmd	Pcmd_19(INT_TO_BIT(TSPX_PwrlvlB, 5))	
frql_at	-	
cchd	-	
ch1mod	ChMod_r02	
ch2d_at	-	
ch2mod	-	
ma_at	MoblAllc_r09	
strt	strt	
frql_bt	-	
ch1d_bt	ChDescrp_fhiei(TSPX_Chtp2, slot2, tsc2, INT_TO_BIT(TSPX_Maio3, 6), INT_TO_BIT(TSPX_Hsn3, 6))	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	MoblAllc_r07	
cphms	-	
Detailed Comments:	used in TC_26_6_13_2 only.	

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_r16(slot:SN; tsc:TSC; strt:STRT)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message defined by PIXIT	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_fh(TSPX_Chtp4, slot, tsc, INT_TO_BIT(TSPX_Maio8, 6), INT_TO_BIT(TSPX_Hsn8, 6))	
pcmd	Pcmd_19(INT_TO_BIT(TSPX_PwrlvlC, 5))	
frql_at	-	
cchd	-	
ch1mod	ChMod_r01	
ch2d_at	-	
ch2mod	-	
ma_at	MoblAllc_r12	
strt	strt	
frql_bt	-	
ch1d_bt	ChDescrp_fhiei(TSPX_Chtp4, slot, tsc, INT_TO_BIT(TSPX_Maio9, 6), INT_TO_BIT(TSPX_Hsn9, 6))	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	MoblAllc_r13	
cphms	-	
Detailed Comments:	used in TC_26_6_13_3 only.	

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_r17(slot:SN; tsc:TSC; strt:STRT)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message defined by PIXIT	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_fh(TSPX_Chtp6, slot, tsc, INT_TO_BIT(TSPX_Maio12, 6), INT_TO_BIT(TSPX_Hsn12, 6))	
pcmd	Pcmd_19(INT_TO_BIT(TSPX_PwrlvlD, 5))	
frql_at	-	
cchd	-	
ch1mod	ChMod_sign_iei	
ch2d_at	-	
ch2mod	-	
ma_at	MoblAllc_r16	
strt	strt	
frql_bt	-	
ch1d_bt	ChDescrp_fhiei(TSPX_Chtp6, slot, tsc, INT_TO_BIT(TSPX_Maio13, 6), INT_TO_BIT(TSPX_Hsn13, 6))	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	MoblAllc_r17	
cphms	-	
Detailed Comments:	used in TC_26_6_13_3 only.	

PDU Constraint Declaration		
Constraint Name:	AsgnCmp_02	
PDU Type:	ASS_COM_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMPLETE message containing any cause.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101001'B	
rrcau	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AssgnFl_01	
PDU Type:	ASSFL_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101111'B	
rrcau	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AssgnFl_02	
PDU Type:	ASSFL_PDU	
Derivation Path:	AssgnFl_01.	
Comments:	#6F	
Field Name	**Field Value**	**Comments**
rrcau	'01101111'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Abortmsg_01(par: REJCAU)	
PDU Type:	ABRT_PDU	
Derivation Path:		
Comments:	An ABORT message with reject cause which should be given as parameter.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00101001'B	
rejcau	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AuthRequest(cksn: B_3; rand:RAND)	
PDU Type:	AUTH_RQ_PDU	
Derivation Path:		
Comments:	An AUTHENTICATION REQUEST message containing default ciphering key sequence number and default challenge RAND from PIXIT.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00010010'B	
shoct	'0000'B	
cphksn	CphKeySN_07(cksn)	
rand	rand	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AuthRequest_inv_01(rand:RAND)	
PDU Type:	AUTH_RQ_PDU	
Derivation Path:		
Comments:	An invalid AUTHENTICATION REQUEST message containing arbitrary spare bits	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00010010'B	
shoct	'0101'B	
cphksn	CphKeySN_02	
rand	rand	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AuthReject_01	
PDU Type:	AUTH_REJ_PDU	
Derivation Path:		
Comments:	An AUTHENTICATION REJECT message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00010001'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AuthResponse	
PDU Type:	AUTH_RES_PDU	
Derivation Path:		
Comments:	An AUTHENTICATION RESPONSE message matching any SRES.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?010100'B	
sres	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallConfirm_01(ti:TI)	
PDU Type:	CALL_CO_PDU	
Derivation Path:		
Comments:	a CALL CONFIRMED message with TI parameterized	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?001000'B	
bcri	*	
bcap1	*	
bcap2	*	
cau	-	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallConfirm_02(Ti:TI)	
PDU Type:	CALL_CO_PDU	
Derivation Path:		
Comments:	a CALL CONFIRMED message with cause #17.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?001000'B	
bcri	*	
bcap1	*	
bcap2	*	
cau	Cause_17	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallConfirm_03(ti:TI)	
PDU Type:	CALL_CO_PDU	
Derivation Path:		
Comments:	a CALL CONFIRMED message with bcap1 presented but not checked	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?001000'B	
bcri	*	
bcap1	?	
bcap2	*	
cau	-	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallConfirm_04(ti:TI; Bcap1:BCAP)	
PDU Type:	CALL_CO_PDU	
Derivation Path:		
Comments:	a CALL CONFIRMED message with TI and bcap1 parameterized	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?001000'B	
bcri	-	
bcap1	Bcap1	
bcap2	-	
cau	-	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallConfirm_05(ti:TI; ri:RPI; Bcap1, Bcap2:BCAP)	
PDU Type:	CALL_CO_PDU	
Derivation Path:		
Comments:	a CALL CONFIRMED message with TI parameterized	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?001000'B	
bcri	ri	
bcap1	Bcap1	
bcap2	Bcap2	
cau	-	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallConfirm_20(ti:TI)	
PDU Type:	CALL_CO_PDU	
Derivation Path:		
Comments:	a CALL CONFIRMED message without bearer capability information elements	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?001000'B	
bcri	-	
bcap1	-	
bcap2	-	
cau	-	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallConfirm_Efr_20(ti:TI)	
PDU Type:	CALL_CO_PDU	
Derivation Path:		
Comments:	a CALL CONFIRMED message for EFR Mobiles	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?001000'B	
bcri	-	
bcap1	*	
bcap2	-	
cau	-	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallProced_01(Ti:TI)	
PDU Type:	CALL_PROC_PDU	
Derivation Path:		
Comments:	An CALL PROCEEDING message with mandatory IE's only.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000010'B	
bcri	-	
bcap1	-	
bcap2	-	
fie	-	
pi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallProced_03	
PDU Type:	CALL_PROC_PDU	
Derivation Path:		
Comments:	An CALL PROCEEDING message templet used for CallProcGen	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'00000010'B	
bcri	-	
bcap1	-	
bcap2	-	
fie	-	
pi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallProced_inv_02(Ti:TI)	
PDU Type:	CALL_PROC_PDU	
Derivation Path:		
Comments:	used as an invalid CC message.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000010'B	
bcri	'01011110'B	
bcap1	Bcap_02	
bcap2	-	
fie	-	
pi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CCStatus_01(Ti:TI)	
PDU Type:	CCST_PDU	
Derivation Path:		
Comments:	A CC STATUS message to match any received CC STATUS.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?111101'B	
cau	?	
cst	?	
acst	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CCStatus_02(ti:TI)	
PDU Type:	CCST_PDU	
Derivation Path:		
Comments:	A STATUS message containing cause value #97	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?111101'B	
cau	Cause_02	
cst	?	
acst	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CCStatus_03(Ti:TI)	
PDU Type:	CCST_PDU	
Derivation Path:	CCStatus_01.	
Comments:	CC STATUS message containing cause value #98	
Field Name	**Field Value**	**Comments**
ti	Ti	
cau	Cause_03	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CCStatus_04(Ti:TI)	
PDU Type:	CCST_PDU	
Derivation Path:	CCStatus_01.	
Comments:	A CC STATUS message containing cause value #96	
Field Name	**Field Value**	**Comments**
ti	Ti	
cau	Cause_04	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CCStatus_08(Ti:TI; st:INTEGER)	
PDU Type:	CCST_PDU	
Derivation Path:		
Comments:	A CC STATUS message to match a received CC STATUS containing CC state 'st', cause = #97	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?111101'B	
cau	Cause_02	
cst	CallState_01(st)	
acst	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CCStatus_14(Ti:TI; st:INTEGER)	
PDU Type:	CCST_PDU	
Derivation Path:		
Comments:	A CC STATUS message to match a received CC STATUS containing CC state = `st`, cause = #30	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?111101'B	
cau	Cause_18	
cst	CallState_01(st)	
acst	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CCStatus_inv_01(ti:TI)	
PDU Type:	CCST_PDU	
Derivation Path:		
Comments:	A CC STATUS message without mandatory cause IE and call state IE.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00111101'B	
cau	-	
cst	-	
acst	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CCStatusEq_01(Ti:TI)	
PDU Type:	CCST_ENQ_PDU	
Derivation Path:		
Comments:	A STATUS ENQUIRY message	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00110100'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChmomoAck_01(chmd:CHMOD; chd: CHD)	
PDU Type:	CHMMO_ACK_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010111'B	
chd	chd	
chmod	chmd	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChmomoAck_02(chmd:CHMOD; chd:CHD)	
PDU Type:	CHMMO_ACK_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010111'B	
chd	chd	
chmod	chmd	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChmomoAck_08(type:CH_TDMA; chmd:B_8; slot:SN; tsc :TSC; arfcn:INTEGER)	
PDU Type:	CHMMO_ACK_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010111'B	
chd	ChDescrp_nfh(type, slot, tsc, arfcn)	
chmod	ChMod(chmd)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChmomoReq_01(chmd:CHMOD; chd:CHD)	
PDU Type:	CHMMO_PDU	
Derivation Path:		
Comments:	the channel being modified is default full rate traffic channel.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010000'B	
chd	chd	
chmod	chmd	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChmomoReq_02(chmd:CHMOD; chd:CHD)	
PDU Type:	CHMMO_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010000'B	
chd	chd	
chmod	chmd	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChmomoReq_07(type: CH_TDMA; chmd: B_8; slot: SN; tsc: TSC; arfcn: INTEGER)	
PDU Type:	CHMMO_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010000'B	
chd	ChDescrp_nfh(type, slot, tsc, arfcn)	
chmod	ChMod(chmd)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRelease_01	
PDU Type:	CH_REL_PDU	
Derivation Path:		
Comments:	A CHANNEL RELEASE message with RR cause = normal event	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00001101'B	
rrcau	'00000000'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRelease_inv_01	
PDU Type:	CH_REL_PDU	
Derivation Path:	ChRelease_01.	
Comments:	An invalid CHANNEL RELEASE message without mandatory IE RR cause	
Field Name	**Field Value**	**Comments**
rrcau	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRelease_inv_02	
PDU Type:	CH_REL_PDU	
Derivation Path:	ChRelease_01.	
Comments:	A CHANNEL RELEASE message containing incorrect skip indicator 6.	
Field Name	**Field Value**	**Comments**
ski	'0110'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRelease_inv_03	
PDU Type:	CH_REL_PDU_ERR	
Derivation Path:		
Comments:	A CHANNEL RELEASE message containing additional IE unknown in the RR protocol	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00001101'B	
rrcau	'00000000'B	
add	'6205AA55EF6701'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_01	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	A CHANNEL REQUEST message containing establishment cause = answer to paging.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'100?????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_02	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match any received CHANNEL REQUEST message	
Field Name	**Field Value**	**Comments**
ecau_rrf	'????????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_03	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message containing establishment cause = '0001'B, "other procedures which can be completed with an SDCCH".	
Field Name	**Field Value**	**Comments**
ecau_rrf	'0001????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_04	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message which originates a call (establishment cause = '111'B), "Originating call and TCH/F is needed, or IMSI detach, SMS or SS procedures that can be completed with an SDCCH and NECI set to 0".	
Field Name	**Field Value**	**Comments**
ecau_rrf	'111?????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_05	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message containing establishment cause = '0100'B, "Originating speech call from dual-rate mobile station when TCH/H is sufficient and the network sets NECI bit to 1"	
Field Name	**Field Value**	**Comments**
ecau_rrf	'0100????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_06	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message which originates a half rate data call (establishment cause = '0101'B).	
Field Name	**Field Value**	**Comments**
ecau_rrf	'0101????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_07	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '0010'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'0010????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_08	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '0011'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'0011????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_09	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '000'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'000?????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_10	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	A CHANNEL REQUEST message with establishment cause = '110'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'110?????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_11	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '011010'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'011010??'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_12	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '100'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'100?????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_13	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '100'B or '0010'B or '0001'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	('100?????'B, '0010????'B, '0001????'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_14	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '100'B or '0011'B or '0001'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	('100?????'B, '0011????'B, '0001????'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_15	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '111'B or '0100'B or '0101'B --- initiate outgoing call or SDCCH other procedures.	
Field Name	**Field Value**	**Comments**
ecau_rrf	('111?????'B, '0100????'B, '0101????'B, '101?????'B, '0001????'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_16	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '101'B for emergency call.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'101?????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_17	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	A CHANNEL REQUEST message containing establishment cause = 100, 0010, 0011, 0001	
Field Name	**Field Value**	**Comments**
ecau_rrf	('100?????'B, '0010????'B, '0011????'B, '0001????'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_18	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '0000'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'0000????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_19	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '111'B or '101'B --- initiate outgoing call	
Field Name	**Field Value**	**Comments**
ecau_rrf	('111?????'B, '101?????'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ClassChange_01	
PDU Type:	CLM_CHN_PDU	
Derivation Path:		
Comments:	CLASSMARK CHANGE message containing classmark2 indicating original rf power class	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010110'B	
msclm	ClassMark2	
msclm_adi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ClassChange_02	
PDU Type:	CLM_CHN_PDU	
Derivation Path:	ClassChange_01.	
Comments:	CLASSMARK CHANGE message containing classmark2 indicating new rf power class due to addition of power amplification	
Field Name	**Field Value**	**Comments**
msclm	ClassMark2Amp	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ClassChange_03	
PDU Type:	CLM_CHN_PDU	
Derivation Path:	ClassChange_01.	
Comments:	CLASSMARK CHANGE message containing classmark2 indicating original rf power class and possible classmark3.	
Field Name	**Field Value**	**Comments**
msclm	ClassMark2	
msclm_adi	ClassMark3 IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ClassMarkEnq_01	
PDU Type:	CLM_ENQ_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010011'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMReEstReq_02(mi:MI; lai:LAI; cksn:B_3)	
PDU Type:	CMRE_RQ_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?101000'B	
shoct	'0000'B	
cphksn	CphKeySN_01(cksn)	
msclm	ClassMark2	
mi	mi	
lai	lai IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceAcp_01	
PDU Type:	CMS_ACP_PDU	
Derivation Path:		
Comments:	a CM SERVICE ACCEPT message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00100001'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceRej_01	
PDU Type:	CMS_REJ_PDU	
Derivation Path:		
Comments:	reject cause = 'service or option not available, unspecified"	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00100010'B	
mmcau	'20'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceRej_02	
PDU Type:	CMS_REJ_PDU	
Derivation Path:		
Comments:	reject cause = "IMEI not accepted"	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00100010'B	
mmcau	'05'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceRej_03	
PDU Type:	CMS_REJ_PDU	
Derivation Path:		
Comments:	reject cause = "Service Option not supported"	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00100010'B	
mmcau	'20'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceRej_04	
PDU Type:	CMS_REJ_PDU	
Derivation Path:		
Comments:	reject cause = 'network failure"	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00100010'B	
mmcau	'11'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceRej_30(par: REJCAU)	
PDU Type:	CMS_REJ_PDU	
Derivation Path:		
Comments:	reject cause = 'service or option not available, unsepcified"	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00100010'B	
mmcau	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_01	
PDU Type:	CMS_RQ_PDU	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?100100'B	
cphksn	?	
svtype	?	
msclm	?	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_02	
PDU Type:	CMS_RQ_PDU	
Derivation Path:	CMServiceReq_01.	
Comments:	To match a received CM SERVICE REQUEST message containing mobile station classmark 2 indicating new RF power capability.	
Field Name	**Field Value**	**Comments**
msclm	ClassMark2Amp	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_04	
PDU Type:	CMS_RQ_PDU	
Derivation Path:	CMServiceReq_01.	
Comments:	To match a received CM SERVICE REQUEST message containing CM service type = "Mobile originating call establishment or packet mode connection establishment" or " emergency call establishment".	
Field Name	**Field Value**	**Comments**
svtype	('0001'B, '0010'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_05(mi:MI)	
PDU Type:	CMS_RQ_PDU	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message for emergency call with IMEI.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?100100'B	
cphksn	?	
svtype	C_CMServiceTypeE	
msclm	?	
mi	mi	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_06(cksn:B_3)	
PDU Type:	CMS_RQ_PDU	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message for emergency call with TMSI and correct CKSN.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?100100'B	
cphksn	CphKeySN_01(cksn)	
svtype	C_CMServiceTypeE	
msclm	?	
mi	MiTmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_07	
PDU Type:	CMS_RQ_PDU	
Derivation Path:	CMServiceReq_01.	
Comments:	To match any received CM SERVICE REQUEST message for emergency call with IMEI and CKSN indicating "no key is available'.	
Field Name	**Field Value**	**Comments**
cphksn	CphKeySN_06	
svtype	C_CMServiceTypeE	
msclm	ClassMark2	
mi	MiImei_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_08	
PDU Type:	CMS_RQ_PDU	
Derivation Path:	CMServiceReq_01.	
Comments:	To match the received CM SERVICE REQUEST message indicating " supplementary service activation"	
Field Name	**Field Value**	**Comments**
svtype	'1000'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_09	
PDU Type:	CMS_RQ_PDU	
Derivation Path:	CMServiceReq_01.	
Comments:	To match the received CM SERVICE REQUEST message indicating " short message transfer"	
Field Name	**Field Value**	**Comments**
svtype	'0100'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_30(parexpected_mi: MI)	
PDU Type:	CMS_RQ_PDU	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?100100'B	
cphksn	?	
svtype	?	
msclm	?	
mi	parexpected_mi	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_31(parexpected_mi: MI; cksn: BITSTRING)	
PDU Type:	CMS_RQ_PDU	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?100100'B	
cphksn	CphKeySN_07(cksn)	
svtype	C_CMServiceTypeE	
msclm	?	
mi	parexpected_mi	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_32(parexpected_mi: MI; cksn: BITSTRING)	
PDU Type:	CMS_RQ_PDU	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?100100'B	
cphksn	CphKeySN_07(cksn)	
svtype	?	
msclm	?	
mi	parexpected_mi	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Connect_01(ti:TI)	
PDU Type:	CONN_PDU	
Derivation Path:		
Comments:	CC CONNECT message matching any received value	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?000111'B	
fie	*	
pi	-	
cnn	-	
cns	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Connect_02(Ti:TI)	
PDU Type:	CONN_PDU	
Derivation Path:		
Comments:	CC CONNECT message containing mandatory IE's only.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000111'B	
fie	-	
pi	-	
cnn	-	
cns	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Connect_05(Ti:TI; fie:FIE)	
PDU Type:	CONN_PDU	
Derivation Path:		
Comments:	CC CONNECT message containing faciliy IE.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000111'B	
fie	fie	
pi	-	
cnn	-	
cns	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Connect_inv_01(Ti:TI)	
PDU Type:	CONN_PDU_ERR	
Derivation Path:		
Comments:	An invalid CONNECT message containing an optional IE coded as comprehension required.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000111'B	
unknown	UnknownIE_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ConnectAck_01(ti:TI)	
PDU Type:	CONN_ACK_PDU	
Derivation Path:		
Comments:	A CONNECT ACKNOWLEDGE message n -> ms	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00001111'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ConnectAck_02(Ti:TI)	
PDU Type:	CONN_ACK_PDU	
Derivation Path:		
Comments:	A CONNECT ACKNOWLEDGE message	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?001111'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CphModeCmd_01	
PDU Type:	CPHM_CMD_PDU	
Derivation Path:		
Comments:	CIPHERING MODE COMMAND message, the ciphering algorithm is specified by PIXIT.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00110101'B	
cph_res	CiphRes_01	
cphms	CphMod_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CphModeCmd_02	
PDU Type:	CPHM_CMD_PDU	
Derivation Path:	CphModeCmd_01.	
Comments:	CIPHERING MODE COMMAND message with no ciphering.	
Field Name	**Field Value**	**Comments**
cphms	CphMod_02	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CphModeCmd_03	
PDU Type:	CPHM_CMD_PDU	
Derivation Path:	CphModeCmd_01.	
Comments:	CIPHERING MODE COMMAND message with no ciphering and IMEI included, the ciphering algorithm is specified by PIXIT.	
Field Name	**Field Value**	**Comments**
cph_res	CiphRes_02	
cphms	CphMod_02	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CphModeCmd_inv_01	
PDU Type:	CPHM_CMD_PDU	
Derivation Path:	CphModeCmd_01.	
Comments:	Invalid CIPHERING MODE COMMAND message with mandatory IE's missing	
Field Name	**Field Value**	**Comments**
cph_res	-	
cphms	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CphModeCmd_inv_02	
PDU Type:	CPHM_CMD_PDU	
Derivation Path:	CphModeCmd_01.	
Comments:	CIPHERING MODE COMMAND message containing incorrect skip identifier	
Field Name	**Field Value**	**Comments**
ski	'0011'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CphModeCmd_inv_03	
PDU Type:	CPHM_CMD_PDU_ERR	
Derivation Path:		
Comments:	CIPHERING MODE COMMAND message containing unknown IE	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00110101'B	
cph_res	CiphRes_01	
cphms	CphMod_03	
add	'92'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CphModeCmp_01	
PDU Type:	CPHM_COM_PDU	
Derivation Path:		
Comments:	To match any received CIPHERING MODE COMPLETE message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00110010'B	
mei	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CphModeCmp_02	
PDU Type:	CPHM_COM_PDU	
Derivation Path:	CphModeCmp_01.	
Comments:	To match a received CIPHERING MODE COMPLETE message without IMEI.	
Field Name	**Field Value**	**Comments**
mei	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CphModeCmp_03	
PDU Type:	CPHM_COM_PDU	
Derivation Path:		
Comments:	To match a received CIPHERING MODE COMPLETE message containing IMEI.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00110010'B	
mei	MiImeisv_01iei	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	DisconnS(Ti:TI; cau:CAU; progind:PI; uuinf:UU)	
PDU Type:	DISC_PDU	
Derivation Path:		
Comments:	A DISCONNECT message that contains cause = normal clearing and progress indicator = #8.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00100101'B	
cau	cau	
fie	-	
pi	progind	
uu	uuinf	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	DisconnR(Ti:TI; cau:CAU)	
PDU Type:	DISC_PDU	
Derivation Path:		
Comments:	A default DISCONNECT message matching parametrised value ms -> n.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?100101'B	
cau	cau	
fie	*	
pi	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Disconn_10(Ti:TI)	
PDU Type:	DISC_PDU	
Derivation Path:		
Comments:	To match a received DISCONNECT message containing transaction identifier = Ti , cause #68 and facility_IE45 for ForarwdChargeAdvice ReturnResult ms -> n.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?100101'B	
cau	Cause_Def	
fie	facilityIErcviei(FwdChAdvRslt_01)	
pi	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Disconn_inv_01(ti:TI)	
PDU Type:	DISC_PDU	
Derivation Path:		
Comments:	A DISCONNECT message containing cause value #16 and location = user, the transaction ID does not refer to the active call.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00100101'B	
cau	Cause_01	
fie	-	
pi	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Disconn_inv_02(ti:TI)	
PDU Type:	DISC_PDU	
Derivation Path:		
Comments:	A DISCONNECT message in which the mandatory IE cause is missing. (N -> MS)	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00100101'B	
cau	-	
fie	-	
pi	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Disconn_inv_03(ti :TI)	
PDU Type:	DISC_PDU_ERR	
Derivation Path:		
Comments:	An invalid DISCONNECT message which contains optional unknown IEI	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00100101'B	
cau	Cause_01	
unknown	UnknownIE_02	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Disconn_inv_04(ti :TI)	
PDU Type:	DISC_PDU	
Derivation Path:		
Comments:	An invalid DISCONNECT message containing arbitrary spare bits	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00100101'B	
cau	Cause_06	
fie	-	
pi	ProgInd_01	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FacilityPdu_05(Ti:TI)	
PDU Type:	FAC_PDU	
Derivation Path:		
Comments:	FACILITY message, call dependent, used in TC_11_3, containing any facility iformation element ms->n.	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'0?111010'B	
fie	?	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FacilityPdu_25(Ti:TI; fie:FIE)	
PDU Type:	FAC_PDU	
Derivation Path:		
Comments:	FACILITY message, call dependent, n -> ms	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'00111010'B	
fie	fie	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FacilityPdu_25_ci(Ti:TI; fie:FIE)	
PDU Type:	FAC_PDU	
Derivation Path:		
Comments:	FACILITY message, call independent, n -> ms	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'1011'B	
mt	'00111010'B	
fie	fie	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FacilityPdu_26(Ti:TI; fie:FIE)	
PDU Type:	FAC_PDU	
Derivation Path:		
Comments:	FACILITY message, call dependent ms -> n	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'0?111010'B	
fie	fie	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FacilityPdu_26_ci(Ti:TI; fie:FIE)	
PDU Type:	FAC_PDU	
Derivation Path:		
Comments:	FACILITY message, call independent ms -> n	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'1011'B	
mt	'0?111010'B	
fie	fie	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FreqRedef_01(chd:CHD; ma:MA; strt:STRT; cchd:CCHD)	
PDU Type:	FRQRE_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010100'B	
chd	chd	
ma	ma	
strt	strt	
cchd	cchd	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FreqRedef_20(ts_ccch: BITSTRING; par_chtype:CH_TDMA; par_cchd: OCTETSTRING; par_ma:BITSTRING; par_stime:STRT; n:INTEGER)	
PDU Type:	FRQRE_PDU	
Derivation Path:		
Comments:	Frequency Redefinition	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010100'B	
chd	ChDescrp_fh(par_chtype, ts_ccch, TSPX_TscDef, INT_TO_BIT((TSPX_MAIO MOD n), 6), INT_TO_BIT(TSPX_HSN, 6))	
ma	MoblAllc_20_Be1(par_ma)	
strt	par_stime	
cchd	CellChDes_20_Be(par_cchd)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FreqRedef_21(ts_ccch: BITSTRING; par_chtype:CH_TDMA; par_ma:BITSTRING; par_stime:STRT; n:INTEGER)	
PDU Type:	FRQRE_PDU	
Derivation Path:		
Comments:	Frequency Redefinition	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010100'B	
chd	ChDescrp_fh(par_chtype, ts_ccch, TSPX_TscDef, INT_TO_BIT((TSPX_MAIO MOD n), 6), INT_TO_BIT(TSPX_HSN, 6))	
ma	MoblAllc_20_Be1(par_ma)	
strt	par_stime	
cchd	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FreqRedef_22(ts_ccch: BITSTRING; par_chtype:CH_TDMA; par_cchd: OCTETSTRING; par_ma1:BITSTRING; par_ma2:BITSTRING; par_stime:STRT; n:INTEGER)	
PDU Type:	FRQRE_PDU	
Derivation Path:		
Comments:	Frequency Redefinition	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010100'B	
chd	ChDescrp_fh(par_chtype, ts_ccch, TSPX_TscDef, INT_TO_BIT((TSPX_MAIO MOD n), 6), INT_TO_BIT(TSPX_HSN, 6))	
ma	MoblAllc_20_Be2(par_ma1, par_ma2)	
strt	par_stime	
cchd	CellChDes_20_Be(par_cchd)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverAcc_01	
PDU Type:	HOACC_PDU	
Derivation Path:		
Comments:	To match any received HANDOVER ACCESS message	
Field Name	**Field Value**	**Comments**
horf	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverAcc_02(horef:HORF)	
PDU Type:	HOACC_PDU	
Derivation Path:		
Comments:	To match a received HANDOVER ACCESS message with handover reference ` horef`.	
Field Name	**Field Value**	**Comments**
horf	horef	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverCmd_nfh(type:CH_TDMA; slot:SN; tsc:TSC; ncc, bcc:B_3; sdccharfcn, tcharfcn:INTEGER; horf:HORF; pwlvl:B_5; synchi:SYNCHI; chmod:CHMOD; rtdif:TDIF; ta:TA; cphms:CPHMS)	
PDU Type:	HO_CMD_PDU	
Derivation Path:		
Comments:	An HANDOVER COMMAND indicating finely synchronised intra cell handover.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101011'B	
cd	CellDescrp(ncc, bcc, sdccharfcn)	
ch1d_at	ChDescrp_nfh(type, slot, tsc, tcharfcn)	
horf	horf	
pcmd	Pcmd_20(pwlvl)	
synchi	synchi	
frqsl_at	-	
frql_at	-	
cchd	-	
ch1mod	chmod	
ch2d_at	-	
ch2mod	-	
frqchs_at	-	
ma_at	-	
strt	-	
rtdif	rtdif	
ta	ta	
frqsl_bt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	cphms	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverCmd_fh(type:CH_TDMA; slot:SN; tsc:TSC; ncc, bcc:B_3; sdccharfcn, mod, hsn:INTEGER; horf:HORF; pwlvl:B_5; synchi:SYNCHI; frql1, frql2:FRQL; cchd:CCHD; chmod:CHMOD; frqchs:FRQCHS; ma:MA; rtdif:TDIF; ta:TA; cphms:CPHMS)	
PDU Type:	HO_CMD_PDU	
Derivation Path:		
Comments:	A derived HANDOVER COMMAND containing TCH/F_FH in non synchronized new CELL A.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101011'B	
cd	CellDescrp(ncc, bcc, sdccharfcn)	
ch1d_at	ChDescrp_fh(type, slot, tsc, INT_TO_BIT((TSPX_MAIO MOD mod), 6), INT_TO_BIT(hsn, 6))	
horf	horf	
pcmd	Pcmd_20(pwlvl)	
synchi	synchi	
frqsl_at	frql1	
frql_at	frql2	
cchd	cchd	
ch1mod	chmod	
ch2d_at	-	
ch2mod	-	
frqchs_at	frqchs	
ma_at	ma	
strt	-	
rtdif	rtdif	
ta	ta	
frqsl_bt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	cphms	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverCmd(ncc, bcc:B_3; slot:SN; tsc:TSC; cchd:CCHD; strt:STRT; arfcn: INTEGER; type:CH_TDMA; maio1, maio2, hsn1, hsn2:INTEGER; horf:HORF; pwlvl:INTEGER; synchi:SYNCHI; chmod:CHMOD; ma1, ma2:MA; cphms:CPHMS)	
PDU Type:	HO_CMD_PDU	
Derivation Path:		
Comments:	non synchronised, no ciphering for GSM	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101011'B	
cd	CellDescrp(ncc, bcc, arfcn)	
ch1d_at	ChDescrp_fh(type, slot, tsc, INT_TO_BIT(maio1, 6), INT_TO_BIT(hsn1, 6))	
horf	horf	
pcmd	Pcmd_19(INT_TO_BIT(pwlvl, 5))	
synchi	synchi	
frqsl_at	-	
frql_at	-	
cchd	cchd	
ch1mod	chmod	
ch2d_at	-	
ch2mod	-	
frqchs_at	-	
ma_at	ma1	
strt	strt	
rtdif	-	
ta	-	
frqsl_bt	-	
frql_bt	-	
ch1d_bt	ChDescrp_fhiei(type, slot, tsc, INT_TO_BIT(maio2, 6), INT_TO_BIT(hsn2, 6))	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	ma2	
cphms	cphms	
Detailed Comments:	used in TC_26_6_13_5 only.	

PDU Constraint Declaration		
Constraint Name:	HandOverCmd_inv_01(slot:SN; tsc:TSC; ncc, bcc:B_3; arfcn:INTEGER)	
PDU Type:	HO_CMD_PDU	
Derivation Path:		
Comments:	An invalid HANDOVER COMMAND containing incorrect skip indicator.	
Field Name	**Field Value**	**Comments**
ski	'0101'B	
rrpd	'0110'B	
mt	'00101011'B	
cd	CellDescrp(ncc, bcc, arfcn)	
ch1d_at	ChDescrp_nfh('00001'B, slot, tsc, arfcn)	
horf	'00000001'B	
pcmd	Pcmd_19('01000'B)	
synchi	-	
frqsl_at	-	
frql_at	-	
cchd	-	
ch1mod	-	
ch2d_at	-	
ch2mod	-	
frqchs_at	-	
ma_at	-	
strt	-	
rtdif	-	
ta	-	
frqsl_bt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverCmd_inv_02(slot:SN; tsc:TSC; ncc, bcc:B_3; arfcn:INTEGER)	
PDU Type:	HO_CMD_PDU	
Derivation Path:		
Comments:	HANDOVER COMMAND which contains, in the non-imperative part, an IE encoded as comprehension required.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101011'B	
cd	CellDescrp(ncc, bcc, arfcn)	
ch1d_at	ChDescrp_nfh('00001'B, slot, tsc, C_arfcnA)	
horf	'00000001'B	
pcmd	Pcmd_19('01000'B)	
synchi	Synchi_01	
frqsl_at	Frql_01	
frql_at	-	
cchd	-	
ch1mod	-	
ch2d_at	-	
ch2mod	-	
frqchs_at	-	
ma_at	-	
strt	-	
rtdif	-	
ta	-	
frqsl_bt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverCmp_01	
PDU Type:	HO_COM_PDU	
Derivation Path:		
Comments:	To match any received HANDOVER COMPLETE message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101100'B	
rrcau	?	
motdif	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverCmp_20	
PDU Type:	HO_COM_PDU	
Derivation Path:		
Comments:	A basic received constraint for HANDOVER COMPLETE message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101100'B	
rrcau	'00000000'B	
motdif	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOvFail_01	
PDU Type:	HOFL_PDU	
Derivation Path:		
Comments:	A HANDOVER FAILURE message matching any RR cause.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101000'B	
rrcau	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOvFail_02	
PDU Type:	HOFL_PDU	
Derivation Path:		
Comments:	A HANDOVER FAILURE message matching any abnormal release RR cause.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101000'B	
rrcau	('00000001'B, '00000010'B, '00000011'B, '00000100'B, '01101111'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Holdpdu_01(Ti:TI)	
PDU Type:	HOLD_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?011000'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HoldAckpdu_01(Ti:TI)	
PDU Type:	HOLD_ACK_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00011001'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDResponse_01	
PDU Type:	ID_RES_PDU	
Derivation Path:		
Comments:	An IDENTITY RESPONSE message which matches any mobile identity	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?011001'B	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDResponse_02	
PDU Type:	ID_RES_PDU	
Derivation Path:	IDResponse_01.	
Comments:	An IDENTITY RESPONSE message which matches TMSI of the MS under test	
Field Name	**Field Value**	**Comments**
mi	MiTmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDResponse_30(par:MI)	
PDU Type:	ID_RES_PDU	
Derivation Path:		
Comments:	An IDENTITY RESPONSE message which matches the given MI of the MS under test.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?011001'B	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDRequest_01(type:B_4)	
PDU Type:	ID_RQ_PDU	
Derivation Path:		
Comments:	An IDENTITY REQUEST message with specified identity type.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00011000'B	
shoct	'0000'B	
idtype	type	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDRequest_inv_01(skip :INTEGER)	
PDU Type:	ID_RQ_PDU	
Derivation Path:		
Comments:	An invalid IDENTITY REQUEST message with incorrect skip indicator.	
Field Name	**Field Value**	**Comments**
ski	INT_TO_BIT(skip, 4)	
mmpd	'0101'B	
mt	'00011000'B	
shoct	'0000'B	
idtype	'0001'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDRequest_inv_02	
PDU Type:	ID_RQ_PDU	
Derivation Path:		
Comments:	An invalid IDENTITY REQUEST message of which the identity type is coded as reserved value	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00011000'B	
shoct	'0000'B	
idtype	'1111'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDRequest_inv_03	
PDU Type:	ID_RQ_PDU	
Derivation Path:		
Comments:	An invalid IDENTITY REQUEST message containing arbitrary spare bits	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00011000'B	
shoct	'1010'B	
idtype	'1100'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_01Def(Rr: BITSTRING; Fn: FN; sub:B_2; slot:SN; tsc:TSC; par_arfcn:INTEGER; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/4 channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(sub)), 5), slot, tsc, par_arfcn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_02(Rr: BITSTRING; Fn: FN; ta:TA; chd:CHD; ma:MA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign a frequency hopping channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	chd	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	ma	
strt	-	
iaroct	IaRestOct_06	
Detailed Comments:	Used only in TC_26_6_6_1	

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_nfh(Rr: BITSTRING; Fn: FN; type: CH_TDMA; slot:SN; tsc:TSC; ta:TA; arfcn: INTEGER; pgm:B_2)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign TCH/F or TCH/H channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(pgm)	
chd	ChDescrp_nfh(type, slot, tsc, arfcn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_221(Rr:BITSTRING; Fn:FN; ts_ccch:SN; tsc:TSC; par_chtype:CH_TDMA; maio:MAIO; hsn:HSN; ta:TA; ma:MA; iel:LENGTH)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((11 + OC_OctToInt(iel)) * 4) + 1, 1)	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_fh(par_chtype, ts_ccch, tsc, maio, hsn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	ma	
strt	-	
iaroct	IaRestOct(iel)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_27(Rr: BITSTRING; Fn: FN;ts_ccch: BITSTRING; par_arfcn: INTEGER; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/8 NonFH channel in any cell.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_sdcch8_nfh(ts_ccch, TSPX_TscDef, TSPX_SDCCH8SubDef, par_arfcn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_E_01(Rr: BITSTRING; Fn: FN;ts_ccch: BITSTRING; tsc:TSC; par_arfcn: INTEGER; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/8 NonFH channel in any cell.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_sdcch8_nfh(ts_ccch, tsc, TSPX_SDCCH8SubA, par_arfcn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_E_02(Rr: BITSTRING; Fn: FN;ts_ccch: BITSTRING; tsc:TSC; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/8 FH channel in EGSM cases.	
Field Name	**Field Value**	**Comments**
l2_pl	'31'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_sdcch8_fh(ts_ccch, tsc, TSPX_SDCCH8SubB, INT_TO_BIT((TSPX_MAIO MOD 4), 6), INT_TO_BIT(TSPX_HSN, 6))	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_281e	
strt	-	
iaroct	IaRestOct_02	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_inv_01(Rr :BITSTRING; Fn:FN; slot:SN; tsc:TSC; arfcn:INTEGER; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An invalid IMMEDIATE ASSIGNMENT message with unknown skip indicator ('0001'B)	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0001'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubDef)), 5), slot, tsc, arfcn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_inv_04(Rr :BITSTRING; Fn:FN; slot:SN; tsc:TSC; ta:TA; arfcn:INTEGER)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An invalid IMMEDIATE ASSIGNMENT message containing arbitrary spare bits	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'1010'B	
pm	Pm_02	
chd	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubDef)), 5), slot, tsc, arfcn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_05	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r(chd:CHD; Rr:BITSTRING; Fn:FN; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	chd	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r02(Rr: BITSTRING; Fn: FN; sub:B_3; slot:SN; tsc:TSC; ta:TA; arfcn:INTEGER)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/8 channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_sdcch8_nfh(slot, tsc, sub, arfcn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r13(Rr: BITSTRING; Fn: FN; chd:CHD; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign TCH/F channel with a time advance of 30 bit periods.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	chd	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r14(Rr: BITSTRING; Fn: FN; chd:CHD; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign TCH/H channel with a time advance of 30 bit periods.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	chd	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r27(Rr: BITSTRING; Fn: FN; slot:SN; tsc:TSC; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/8 channel for TC_26_6_13_1.	
Field Name	**Field Value**	**Comments**
l2_pl	'39'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_sdcch8_fh(slot, tsc, TSPX_SDCCH8SubA, INT_TO_BIT(TSPX_Maio1, 6), INT_TO_BIT(TSPX_Hsn1, 6))	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_r05	
strt	-	
iaroct	IaRestOct_08	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r28(Rr: BITSTRING; Fn: FN; slot:SN; tsc:TSC; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/8 channel for TC_26_6_13_2 .	
Field Name	**Field Value**	**Comments**
l2_pl	'39'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_sdcch8_fh(slot, tsc, TSPX_SDCCH8SubB, INT_TO_BIT(TSPX_Maio4, 6), INT_TO_BIT(TSPX_Hsn4, 6))	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_r08	
strt	-	
iaroct	IaRestOct_08	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r29(Rr: BITSTRING; Fn: FN; slot:SN; tsc:TSC; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign the channel defined by PIXIT for TC_26_6_13_3.	
Field Name	**Field Value**	**Comments**
l2_pl	'39'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_fh(TSPX_Chtp3, slot, tsc, INT_TO_BIT(TSPX_Maio6, 6), INT_TO_BIT(TSPX_Hsn6, 6))	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_r10	
strt	-	
iaroct	IaRestOct_08	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r30(Rr: BITSTRING; Fn: FN; slot:SN; tsc:TSC; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message for TC_26_6_13_4.	
Field Name	**Field Value**	**Comments**
l2_pl	'39'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_fh(TSPX_Chtp5, slot, tsc, INT_TO_BIT(TSPX_Maio10, 6), INT_TO_BIT(TSPX_Hsn10, 6))	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_r14	
strt	-	
iaroct	IaRestOct_08	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r31(Rr: BITSTRING; Fn: FN; slot:SN; tsc:TSC; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/8 channel for TC_26_6_13_5.	
Field Name	**Field Value**	**Comments**
l2_pl	'39'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_sdcch8_fh(slot, tsc, TSPX_SDCCH8SubB, INT_TO_BIT(TSPX_Maio14, 6), INT_TO_BIT(TSPX_Hsn14, 6))	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_r18	
strt	-	
iaroct	IaRestOct_08	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r32(Rr: BITSTRING; Fn: FN; slot:SN; tsc:TSC; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/8 channel for TC_26_6_13_6.	
Field Name	**Field Value**	**Comments**
l2_pl	'39'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_sdcch8_fh(slot, tsc, TSPX_SDCCH8SubC, INT_TO_BIT(TSPX_Maio17, 6), INT_TO_BIT(TSPX_Hsn17, 6))	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_r21	
strt	-	
iaroct	IaRestOct_08	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r33(Rr: BITSTRING; Fn: FN; slot:SN; tsc:TSC; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign the channel defined by PIXIT for TC_26_6_13_7.	
Field Name	**Field Value**	**Comments**
l2_pl	'39'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_fh(TSPX_Chtp9, slot, tsc, INT_TO_BIT(TSPX_Maio20, 6), INT_TO_BIT(TSPX_Hsn20, 6))	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_r24	
strt	-	
iaroct	IaRestOct_08	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r34(Rr: BITSTRING; Fn: FN; slot:SN; tsc:TSC; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign the channel defined by PIXIT for TC_26_6_13_8.	
Field Name	**Field Value**	**Comments**
l2_pl	'39'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_fh(TSPX_Chtp11,slot, tsc, INT_TO_BIT(TSPX_Maio24, 6), INT_TO_BIT(TSPX_Hsn24, 6))	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_r28	
strt	-	
iaroct	IaRestOct_08	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r35(Rr: BITSTRING; Fn: FN; slot:SN; tsc:TSC; ta:TA; starttm:STRT)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign the channel defined by PIXIT for TC_26_6_13_9.	
Field Name	**Field Value**	**Comments**
l2_pl	'45'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_fh(TSPX_Chtp13, slot, tsc, INT_TO_BIT(TSPX_Maio28, 6), INT_TO_BIT(TSPX_Hsn28, 6))	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_r32	
strt	starttm	
iaroct	IaRestOct_03	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r36(Rr: BITSTRING; Fn: FN; slot:SN; tsc:TSC; ta:TA; starttm:STRT)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign the channel defined by PIXIT for TC_26_6_13_10.	
Field Name	**Field Value**	**Comments**
l2_pl	'45'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_fh(TSPX_Chtp14, slot, tsc, INT_TO_BIT(TSPX_Maio30, 6), INT_TO_BIT(TSPX_Hsn30, 6))	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_r33	
strt	starttm	
iaroct	IaRestOct_04	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_sdcch8(Rr: BITSTRING; Fn: FN; slot:SN; tsc:TSC; subch:B_3; arfcn:INTEGER; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH8 channel	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_sdcch8_nfh(slot, tsc, subch, arfcn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnX_01(Rr: BITSTRING; Fn: FN; sub1, sub2:B_3; slot:SN; tsc:TSC; ta:TA; arfcn:INTEGER)	
PDU Type:	IMMASSX_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT EXTENDED message to assign SDCCH/8 channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111001'B	
shoct	'0000'B	
pm	Pm(C_extended_paging)	
chd1	ChDescrp_sdcch8_nfh(slot, tsc, sub1, arfcn)	
rqr1	Rqr2(Rr, Fn)	
ta1	ta	
chd2	ChDescrp_sdcch8_nfh(slot, tsc, sub2, arfcn)	
rqr2	Rqr2(Rr, Fn)	
ta2	ta	
ma	MoblAllc_01	
strt	-	
iaxroct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnX_02(slot:SN; tsc:TSC; ta:TA; arfcn:INTEGER)	
PDU Type:	IMMASSX_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT EXTENDED message containing paging mode = paging reorganisation and Request References that do not pertain to MS under test.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111001'B	
shoct	'0000'B	
pm	Pm(C_reorg_paging)	
chd1	ChDescrp_sdcch8_nfh(slot, tsc, TSPX_SDCCH8SubA, arfcn)	
rqr1	Rqr3	
ta1	ta	
chd2	ChDescrp_sdcch8_nfh(slot, tsc, TSPX_SDCCH8SubB, C_arfcnA)	
rqr2	Rqr3	
ta2	ta	
ma	MoblAllc_01	
strt	-	
iaxroct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnX_03(slot:SN; tsc:TSC; ta:TA; arfcn:INTEGER)	
PDU Type:	IMMASSX_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT EXTENDED message containing paging mode = paging reorganisation and Request References that do not pertain to MS under test.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111001'B	
shoct	'0000'B	
pm	Pm(C_reorg_paging)	
chd1	ChDescrp_sdcch8_nfh(slot, tsc, TSPX_SDCCH8SubA, arfcn)	
rqr1	Rqr3	
ta1	ta	
chd2	ChDescrp_sdcch8_nfh(slot, tsc, TSPX_SDCCH8SubB, 30)	
rqr2	Rqr3	
ta2	ta	
ma	MoblAllc_01	
strt	-	
iaxroct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnX_04(slot:SN; tsc:TSC; ta:TA; arfcn:INTEGER)	
PDU Type:	IMMASSX_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT EXTENDED message containing paging mode = paging reorganisation and Request References that do not pertain to MS under test.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111001'B	
shoct	'0000'B	
pm	Pm(C_reorg_paging)	
chd1	ChDescrp_sdcch8_nfh(slot, tsc, TSPX_SDCCH8SubA, arfcn)	
rqr1	Rqr3	
ta1	ta	
chd2	ChDescrp_sdcch8_nfh(slot, tsc, TSPX_SDCCH8SubB, 30)	
rqr2	Rqr3	
ta2	ta	
ma	MoblAllc_01	
strt	-	
iaxroct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnX_r01(Rr: BITSTRING; Fn: FN; Rr_9: BITSTRING; Fn_9: FN; slot:SN; tsc:TSC; ta:TA; arfcn:INTEGER)	
PDU Type:	IMMASSX_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/4 channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111001'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd1	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubA)), 5), slot, tsc, arfcn)	
rqr1	Rqr2(Rr, Fn)	
ta1	ta	
chd2	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubB)), 5), slot, tsc, arfcn)	
rqr2	Rqr2(Rr_9, Fn_9)	
ta2	ta	
ma	MoblAllc_01	
strt	-	
iaxroct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnX_r02(Rr: BITSTRING; Fn: FN; Rr_9: BITSTRING; Fn_9: FN; slot:SN; tsc:TSC; ta:TA; arfcn:INTEGER)	
PDU Type:	IMMASSX_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/8 channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111001'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd1	ChDescrp_sdcch8_nfh(slot, tsc, TSPX_SDCCH8SubB, arfcn)	
rqr1	Rqr2(Rr, Fn)	
ta1	ta	
chd2	ChDescrp_sdcch8_nfh(slot, tsc, TSPX_SDCCH8SubC, arfcn)	
rqr2	Rqr2(Rr_9, Fn_9)	
ta2	ta	
ma	MoblAllc_01	
strt	-	
iaxroct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnX_r03(Rr: BITSTRING; Fn: FN; sub1, sub2:B_2; slot:SN; tsc:TSC; ta:TA; arfcn:INTEGER)	
PDU Type:	IMMASSX_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/4 channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111001'B	
shoct	'0000'B	
pm	Pm(C_extended_paging)	
chd1	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(sub1)), 5), slot, tsc, arfcn)	
rqr1	Rqr2(Rr, Fn)	
ta1	ta	
chd2	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(sub2)), 5), slot, tsc, arfcn)	
rqr2	Rqr2(Rr, Fn)	
ta2	ta	
ma	MoblAllc_01	
strt	-	
iaxroct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_01(Rr: BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT REJECT message containing normal paging mode.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111010'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
rqr1	Rqr2(Rr, Fn)	
wi1	'00'H	
rqr2	Rqr2(Rr, Fn)	
wi2	'00'H	
rqr3	Rqr2(Rr, Fn)	
wi3	'00'H	
rqr4	Rqr2(Rr, Fn)	
wi4	'00'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_02(Rr:BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT REJECT message in which only the third request reference addresses the MS under test.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111010'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
rqr1	Rqr1(Rr, Fn)	
wi1	'02'H	
rqr2	Rqr1(Rr, Fn)	
wi2	'02'H	
rqr3	Rqr2(Rr, Fn)	
wi3	'00'H	
rqr4	Rqr1(Rr, Fn)	
wi4	'02'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_03(Rr: BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:	ImmAsgnRej_01.	
Comments:	An IMMEDIATE ASSIGNMENT REJECT message containing normal paging mode and wait indication = 5 seconds.	
Field Name	**Field Value**	**Comments**
rqr1	Rqr2(Rr, Fn)	
wi1	'05'H	
rqr2	Rqr2(Rr, Fn)	
wi2	'05'H	
rqr3	Rqr2(Rr, Fn)	
wi3	'05'H	
rqr4	Rqr2(Rr, Fn)	
wi4	'05'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_04(Rr: BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:	ImmAsgnRej_01.	
Comments:	An IMMEDIATE ASSIGNMENT REJECT message containing normal paging mode and wait indication = 6 seconds.	
Field Name	**Field Value**	**Comments**
rqr1	Rqr2(Rr, Fn)	
wi1	'06'H	
rqr2	Rqr2(Rr, Fn)	
wi2	'06'H	
rqr3	Rqr2(Rr, Fn)	
wi3	'06'H	
rqr4	Rqr2(Rr, Fn)	
wi4	'06'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_inv_01(Rr: BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:	ImmAsgnRej_01.	
Comments:	An invalid IMMEDIATE ASSIGNMENT REJECT message with skip indicator = 2, reject time = 255 s.	
Field Name	**Field Value**	**Comments**
ski	'0010'B	
rqr1	Rqr2(Rr, Fn)	
wi1	'FF'H	
rqr2	Rqr2(Rr, Fn)	
wi2	'FF'H	
rqr3	Rqr2(Rr, Fn)	
wi3	'FF'H	
rqr4	Rqr2(Rr, Fn)	
wi4	'FF'H	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_inv_02(Rr: BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:	ImmAsgnRej_01.	
Comments:	An invalid IMMEDIATE ASSIGNMENT REJECT message containing arbitrary spare bits	
Field Name	**Field Value**	**Comments**
shoct	'1010'B	
pm	Pm(C_normal_paging)	
rqr1	Rqr2(Rr, Fn)	
rqr2	Rqr2(Rr, Fn)	
wi2	TSPX_T3122	
rqr3	Rqr2(Rr, Fn)	
wi3	TSPX_T3122	
rqr4	Rqr2(Rr, Fn)	
wi4	TSPX_T3122	
iarroct	'010101'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_r01(rqr1, rqr2, rqr3, rqr4: RQR; t1, t2: INTEGER)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT REJECT message in which only the second and the third request reference addresses the MS under test.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111010'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
rqr1	rqr1	
wi1	'00'H	
rqr2	rqr2	
wi2	INT_TO_HEX(t1, 2)	
rqr3	rqr3	
wi3	INT_TO_HEX(t2, 2)	
rqr4	rqr4	
wi4	'00'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_r02(Rr: BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT REJECT message containing normal paging mode.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111010'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
rqr1	Rqr2(Rr, Fn)	
wi1	'00'H	
rqr2	Rqr1(Rr, Fn)	
wi2	'00'H	
rqr3	Rqr1(Rr, Fn)	
wi3	'00'H	
rqr4	Rqr1(Rr, Fn)	
wi4	'00'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_r04	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT REJECT message containing paging mode = "extended paging"and wait indication = 0 seconds. The Request References do not pertain to MS under test.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111010'B	
shoct	'0000'B	
pm	Pm(C_extended_paging)	
rqr1	Rqr3	
wi1	'00'H	
rqr2	Rqr3	
wi2	'00'H	
rqr3	Rqr3	
wi3	'00'H	
rqr4	Rqr3	
wi4	'00'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImsiDetach_01	
PDU Type:	IMSID_IN_PDU	
Derivation Path:		
Comments:	IMSI DETACH INDICATION message matching any MS classmark1 value and any mobile identity value	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?000001'B	
msclm	?	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImsiDetach_30(par:MI)	
PDU Type:	IMSID_IN_PDU	
Derivation Path:		
Comments:	IMSI DETACH INDICATION message matching any MS classmark1 value and given IMSI.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?000001'B	
msclm	?	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtAcp(mcc, mnc, lac:OCTETSTRING)	
PDU Type:	LUP_ACP_PDU	
Derivation Path:		
Comments:	A LOCATION UPDATING ACCEPT message without mobile identity and LAI of PLMN2. LAC set in TCV_lac.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00000010'B	
lai	LocAreaId(mcc, mnc, lac)	
mi	-	
fop	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtAcp_01(newmi: MI; mcc, mnc, lac:OCTETSTRING)	
PDU Type:	LUP_ACP_PDU	
Derivation Path:		
Comments:	To assign a new TMSI with or without mobile identity.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00000010'B	
lai	LocAreaId(mcc, mnc, lac)	
mi	newmi	
fop	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtAcp_02(mcc, mnc, lac:OCTETSTRING)	
PDU Type:	LUP_ACP_PDU	
Derivation Path:		
Comments:	To assign a new TMSI.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00000010'B	
lai	LocAreaId(mcc, mnc, lac)	
mi	-	
fop	'10100001'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtAcp_inv_01(mi1, mi2:MI; mcc, mnc, lac:OCTETSTRING)	
PDU Type:	LUP_ACP_PDU_ERR	
Derivation Path:		
Comments:	Invalid LOCATION UPDATING ACCEPT message containing duplicated mobile identifier.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00000010'B	
lai	LocAreaId(mcc, mnc, lac)	
mi	mi1	
dupmi	mi2	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtReq_01(locup:B_2)	
PDU Type:	LUP_RQ_PDU	
Derivation Path:		
Comments:	A LOCATION UPDATING REQUEST message containing location updating type.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?001000'B	
cphksn	?	
lutype	LocUpType(locup)	
lai	?	
msclm	?	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtReq_04(locup:B_2; par:MI)	
PDU Type:	LUP_RQ_PDU	
Derivation Path:		
Comments:	LOCATION UPDATING REQUEST message containing TMSI.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?001000'B	
cphksn	?	
lutype	LocUpType(locup)	
lai	?	
msclm	?	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtReq_05	
PDU Type:	LUP_RQ_PDU	
Derivation Path:		
Comments:	to match any LOCATION UPDATING REQUEST message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?001000'B	
cphksn	?	
lutype	?	
lai	?	
msclm	?	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtReq_31(par:MI; mcc, mnc, lac:OCTETSTRING; locup:B_2; cksn: BITSTRING)	
PDU Type:	LUP_RQ_PDU	
Derivation Path:		
Comments:	A parameterised LOCATION UPDATING REQUEST message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?001000'B	
cphksn	CphKeySN_07(cksn)	
lutype	LocUpType(locup)	
lai	LocAreaId(mcc, mnc, lac)	
msclm	ClassMark1	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtRej_01(par:REJCAU)	
PDU Type:	LUP_REJ_PDU	
Derivation Path:		
Comments:	A LOCATION UPDATING REJECT message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00000100'B	
rejcau	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MMstatus_01	
PDU Type:	MMST_PDU	
Derivation Path:		
Comments:	A MM STATUS message containing reject cause value #97-- message type non-existent or not implemented	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?110001'B	
rejcau	'61'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MMstatus_02	
PDU Type:	MMST_PDU	
Derivation Path:	MMstatus_01.	
Comments:	cause value = #96-- invalid mandatory information	
Field Name	**Field Value**	**Comments**
rejcau	'60'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MMstatus_03	
PDU Type:	MMST_PDU	
Derivation Path:	MMstatus_01.	
Comments:	cause value = #98 -- message type not compatible with the protocol state	
Field Name	**Field Value**	**Comments**
rejcau	'62'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ModifyComp_02(Ti:TI; bc:BCAP)	
PDU Type:	MODIFY_COM_PDU	
Derivation Path:		
Comments:	n -> ms	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00011111'B	
bcap	bc	
llcmp	-	
hlcmp	-	
rcsd	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ModifyComp_03(Ti:TI; bc:BCAP)	
PDU Type:	MODIFY_COM_PDU	
Derivation Path:		
Comments:	n -> ms	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00011111'B	
bcap	bc	
llcmp	-	
hlcmp	-	
rcsd	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ModifyInd_01(Ti:TI; bc:BCAP)	
PDU Type:	MODIFY_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?010111'B	
bcap	bc	
llcmp	*	
hlcmp	*	
rcsd	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ModifyReq_01(Ti:TI; bc:BCAP)	
PDU Type:	MODIFY_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00010111'B	
bcap	bc	
llcmp	-	
hlcmp	-	
rcsd	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ModifyRj_01(Ti:TI; bc:BCAP)	
PDU Type:	MODIFY_REJ_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?010011'B	
bcap	bc	
cau	?	
llcmp	*	
hlcmp	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ModifyRjRq_01(Ti:TI; bc:BCAP)	
PDU Type:	MODIFY_REJ_PDU	
Derivation Path:		
Comments:	cause = bearer capability not presently available	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00010011'B	
bcap	bc	
cau	Cause_16	
llcmp	-	
hlcmp	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_01	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report without measurement results	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_02	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report match any received MSR_RPT_PDU.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_02	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_03	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing results for 6 strongest carriers.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_03	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_03e(par_measres: MSRR)	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing results for 6 strongest carriers.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	par_measres	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_04	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing 4 strongest carriers.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_04	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_04e	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing 4 strongest carriers.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_04e	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_05	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing 6 strongest carriers and DTX was used.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_05	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_06	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing 6 strongest carriers and DTX is not checked.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_06	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_07	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing 2 strongest carriers and DTX is not used.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_07	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	NotifiReq_01(Ti:TI)	
PDU Type:	NOTIFY_PDU	
Derivation Path:		
Comments:	Containing any valid notification indicator. n -> ms.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00111110'B	
nti	'10000000'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PagingRes_01	
PDU Type:	PG_RES_PDU	
Derivation Path:		
Comments:	To match any received PAGING RESPONSE message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00100111'B	
shoct	'0000'B	
cphksn	?	
msclm	?	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PagingRes_03(cksn:B_3)	
PDU Type:	PG_RES_PDU	
Derivation Path:		
Comments:	To match a received PAGING RESPONSE message with default TMSI, CKSN and classmark2.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00100111'B	
shoct	'0000'B	
cphksn	CphKeySN_01(cksn)	
msclm	ClassMark2	
mi	MiTmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PagingRes_r01	
PDU Type:	PG_RES_PDU	
Derivation Path:	PagingRes_01.	
Comments:	To match any received PAGING RESPONSE message, RR tests.	
Field Name	**Field Value**	**Comments**
mi	MiTmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PagingRes_r02	
PDU Type:	PG_RES_PDU	
Derivation Path:	PagingRes_01.	
Comments:	To match the received PAGING RESPONSE message containing IMSI of the IUT for RR tests.	
Field Name	**Field Value**	**Comments**
mi	MiImsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PagingRes_30(par:MI; cksn: BITSTRING)	
PDU Type:	PG_RES_PDU	
Derivation Path:		
Comments:	To match any received PAGING RESPONSE message. Used in MM test cases.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00100111'B	
shoct	'0000'B	
cphksn	CphKeySN_07(cksn)	
msclm	?	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_01	
PDU Type:	PG1_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE1 message requesting any channel with normal paging mode, the mobile identity is TMSI.	
Field Name	**Field Value**	**Comments**
l2_pl	'25'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100001'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	MiTmsi_01	
mi2	-	
p1roct	'2B2B2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_02	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	A PAGING REQUEST TYPE1 message to request SDCCH channel with normal paging mode for mobile identity MiTmsi_01.	
Field Name	**Field Value**	**Comments**
chn_m1_2	Chneed_02	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_03	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	A PAGING REQUEST TYPE1 message to request TCH/F channel with normal paging mode for mobile identity MiTmsi_01.	
Field Name	**Field Value**	**Comments**
chn_m1_2	Chneed_03	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_04	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	A PAGING REQUEST TYPE1 message to request TCH/H or TCH/F channel with normal paging mode for mobile identity MiTmsi_01.	
Field Name	**Field Value**	**Comments**
chn_m1_2	Chneed_04	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_05	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	PAGING REQUEST TYPE1 message with mobile identity 1 being IMSI of the MS.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((3 + OC_LengthOf(MiImsi_01)) * 4) + 1, 1)	
mi1	MiImsi_01	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O, 19 - OC_LengthOf(MiImsi_01))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_06	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	PAGING REQUEST TYPE1 message with mobile identity 1 being IMSI of the MS.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((3 + OC_LengthOf(MiImsi_r01)) * 4) + 1, 1)	
pm	Pm(C_extended_paging)	
mi1	MiImsi_r01	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O, 19 - OC_LengthOf(MiImsi_r01))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_07	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	PAGING REQUEST TYPE1 message with mobile identity 1 being IMSI of the MS.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((3 + OC_LengthOf(MiImsi_01)) * 4) + 1, 1)	
pm	Pm(C_reorg_paging)	
mi1	MiImsi_01	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O, 19 - OC_LengthOf(MiImsi_01))	
Detailed Comments:	Only used in TC_26_6_2_4 within Pg_Req1_08	

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_30(par_usedmi: MI)	
PDU Type:	PG1_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE1 message requesting any channel with normal paging mode, the mobile identity is paramerised. It is used only in MM test cases.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((3 + OC_LengthOf(par_usedmi)) * 4) + 1, 1)	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100001'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	par_usedmi	
mi2	-	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O, 19 - OC_LengthOf(par_usedmi))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1Norm	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	PAGING REQUEST TYPE1 message containing normal paging mode, used as paging filling message.	
Field Name	**Field Value**	**Comments**
l2_pl	'15'O	
mi1	MiImsi_02	
p1roct	'2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1Reorg	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	PAGING REQUEST TYPE1 message containing paging reorganisation mode, used as paging filling message.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((3 + OC_LengthOf(MiImsi_02)) * 4) + 1, 1)	
pm	Pm(C_reorg_paging)	
mi1	MiImsi_02	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O, 19 - OC_LengthOf(MiImsi_02))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_inv_01(skip:INTEGER)	
PDU Type:	PG1_RQ_PDU	
Derivation Path:		
Comments:	An invalid PAGING REQUEST TYPE 1 message with incorrect skip indicator.	
Field Name	**Field Value**	**Comments**
l2_pl	'25'O	
ski	INT_TO_BIT(skip, 4)	
rrpd	'0110'B	
mt	'00100001'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	MiTmsi_01	
mi2	-	
p1roct	'2B2B2B2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_inv_04	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	An invalid PAGING REQUEST TYPE1 message containing rest octets which are not all '2B'O	
Field Name	**Field Value**	**Comments**
p1roct	'09021234784BFFFF2B2B2B2B2B'O	
Detailed Comments:	'agressive' rest octet coding for 26.5.7.1.1. The first byte is coded in a manner that indicates comprehension required to a mobile that is incorrectly using the L2 pseudo length. The length of this IE is the octet after '09' and is set to 2. After the two bytes of content is the "next IEI". This is 78 which was the IEI allocated (but never used) in phase 1 to the RACH control parameters. The "RACH Control Parameters" are "set" to indicate that the cell barred and all access classes are prohibited.	

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_r01	
PDU Type:	PG1_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE1 message requesting any channel with normal paging mode, the first mobile identity is TMSI, the second is IMSI different from TSPX_IMSI.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100001'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	MiTmsi_01	
mi2	MiImsi_r01iei	
p1roct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_r02	
PDU Type:	PG1_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE1 message requesting any channel with normal paging mode, the first mobile identity is another TMSI, the second is IMSI of the IUT (TSPX_IMSI).	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((9 + OC_LengthOf(MiImsi_01iei)) * 4) + 1, 1)	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100001'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	MiTmsi_r01	
mi2	MiImsi_01iei	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B'O, 13 - OC_LengthOf(MiImsi_01iei))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_r03	
PDU Type:	PG1_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE1 message requesting any channel with normal paging mode, the first mobile identity is an another TMSI differing from different from MiTmsi_01 and _02, the second one is TMSI of the IUT, MiTmsi_01.	
Field Name	**Field Value**	**Comments**
l2_pl	'41'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100001'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	MiTmsi_r01	
mi2	MiTmsi_01iei	
p1roct	'2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_r04	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	An PAGING REQUEST TYPE1 message requesting any channel with normal paging mode, the mobile identity is TMSI, but the identity type is set to 'no identity'.	
Field Name	**Field Value**	**Comments**
mi1	MiTmsi_r02	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_01	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the mobile identity is TMSI of the IUT.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_01	
mi2	Tmsi_r01	
mi3	-	
p2roct	'2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_02	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with extended paging mode and not addressing the MS under test.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_extended_paging)	
mi1	Tmsi_r01	
mi2	Tmsi_r03	
mi3	-	
p2roct	'2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_03	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with paging reorganisation mode, the mobile identity is TMSI of the IUT.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_reorg_paging)	
mi1	Tmsi_01	
mi2	Tmsi_r01	
mi3	-	
p2roct	'2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_04	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with same as before paging mode, the mobile identity is TMSI of the IUT.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_sab_paging)	
mi1	Tmsi_01	
mi2	Tmsi_r01	
mi3	-	
p2roct	'2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_r01	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the 1st mobile identity is TMSI of the IUT. The 2nd one addresses an another MS.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_01	
mi2	Tmsi_r01	
mi3	-	
p2roct	'2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_r02	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the 2nd mobile identity is TMSI of the IUT. The 1st one addresses an another MS.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_r01	
mi2	Tmsi_01	
mi3	-	
p2roct	'2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_r03	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the 3 mobile identity is TMSI of the IUT. The 1st and 2nd one address an another MS.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_r03	
mi2	Tmsi_r01	
mi3	MiTmsi_01iei	
p2roct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_r04	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the 1st and 2nd mobile identity addresses an another MS. The 3rd one is IMSI of the IUT. .	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((11 + OC_LengthOf(MiImsi_01iei)) * 4) + 1, 1)	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_r03	
mi2	Tmsi_r01	
mi3	MiImsi_01iei	
p2roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B'O, 11 - OC_LengthOf(MiImsi_01iei))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_r05	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the 3 mobile identity is TMSI of the IUT with the type of no id. The 1st and 2nd one address an another MS.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_r03	
mi2	Tmsi_r01	
mi3	MiTmsi_r02iei	
p2roct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_r06	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the 1st and 2nd mobile identity addresses an another MS. The 3rd one is IMSI of the IUT. .	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((11 + OC_LengthOf(MiImsi_01iei)) * 4) + 1, 1)	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_extended_paging)	
mi1	Tmsi_r03	
mi2	Tmsi_r01	
mi3	MiImsi_01iei	
p2roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B'O, 11 - OC_LengthOf(MiImsi_01iei))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_01	
PDU Type:	PG3_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE3 message requesting any channel with extended paging mode and not adressing the MS under test.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100100'B	
chn_m1_2	Chneed_01	
pm	Pm(C_extended_paging)	
mi1	Tmsi_r05	
mi2	Tmsi_r01	
mi3	Tmsi_r03	
mi4	Tmsi_r04	
p3roct	'2B2B2B'O	
Detailed Comments:	Only used in 26_6_2_2 within Pg_Req3_01.	

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_02	
PDU Type:	PG3_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE3 message containing paging mode = "same as before".	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100100'B	
chn_m1_2	Chneed_01	
pm	Pm(C_sab_paging)	
mi1	Tmsi_r05	
mi2	Tmsi_r01	
mi3	Tmsi_r03	
mi4	Tmsi_r04	
p3roct	'2B2B2B'O	
Detailed Comments:	Only used in TC_26_6_2_4 within Pg_Req3_02	

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_03	
PDU Type:	PG3_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE3 message containing paging mode = "normal paging".	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100100'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_r05	
mi2	Tmsi_r01	
mi3	Tmsi_r03	
mi4	Tmsi_r04	
p3roct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_r01	
PDU Type:	PG3_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE3 message requesting any channel with normal paging mode, the 1st mobile identity is TMSI of the IUT. The 2nd, 3rd and 4th one address another MSs.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100100'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_01	
mi2	Tmsi_r01	
mi3	Tmsi_r03	
mi4	Tmsi_r04	
p3roct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_r02	
PDU Type:	PG3_RQ_PDU	
Derivation Path:	PgReqTp3_r01.	
Comments:	An PAGING REQUEST TYPE3 message requesting any channel with normal paging mode, the 2nd mobile identity is TMSI of the IUT. The 1st, 3rd and 4th one address another MSs.	
Field Name	**Field Value**	**Comments**
mi1	Tmsi_r01	
mi2	Tmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_r03	
PDU Type:	PG3_RQ_PDU	
Derivation Path:	PgReqTp3_r01.	
Comments:	An PAGING REQUEST TYPE3 message requesting any channel with normal paging mode, the 3rd mobile identity is TMSI of the IUT. The 1st, 2nd and 4th one address another MSs.	
Field Name	**Field Value**	**Comments**
mi1	Tmsi_r03	
mi3	Tmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_r04	
PDU Type:	PG3_RQ_PDU	
Derivation Path:	PgReqTp3_r01.	
Comments:	An PAGING REQUEST TYPE3 message requesting any channel with normal paging mode, the 4th mobile identity is TMSI of the IUT. The 1st, 2nd and 3rd one address another MSs.	
Field Name	**Field Value**	**Comments**
mi1	Tmsi_r04	
mi4	Tmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_r05	
PDU Type:	PG3_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE3 message requesting any channel with normal paging mode, the 1st mobile identity is TMSI of the IUT. The 2nd, 3rd and 4th one address another MSs.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100100'B	
chn_m1_2	Chneed_01	
pm	Pm(C_extended_paging)	
mi1	Tmsi_01	
mi2	Tmsi_r01	
mi3	Tmsi_r03	
mi4	Tmsi_r04	
p3roct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Phyinfor_01(ta:TA)	
PDU Type:	PHYINFO_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101101'B	
ta	ta	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Progress_01(Ti:TI)	
PDU Type:	PROG_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000011'B	
pi	ProgInd_02	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Progress_02(Ti:TI)	
PDU Type:	PROG_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000011'B	
pi	ProgInd_03	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	RegisterPdu_01	
PDU Type:	REGISTER_PDU	
Derivation Path:		
Comments:	To match any received REGISTER message	
Field Name	**Field Value**	**Comments**
ti	?	
sspd	('1011'B, '0011'B)	
mt	'0?111011'B	
fie	?	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	RegisterPdu_03(fie:FIE)	
PDU Type:	REGISTER_PDU	
Derivation Path:		
Comments:	To match a received REGISTER message invoking registration of CFRNy for Speech	
Field Name	**Field Value**	**Comments**
ti	?	
sspd	'1011'B	
mt	'0?111011'B	
fie	fie	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	RegisterPdu_34(Ti:TI; fie:FIE)	
PDU Type:	REGISTER_PDU	
Derivation Path:		
Comments:	To send a REGISTER message containing Invoke for UnstructuredSS-Notify	
Field Name	**Field Value**	**Comments**
ti	Ti	
sspd	'1011'B	
mt	'00111011'B	
fie	fie	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Release_01(ti:TI)	
PDU Type:	REL_PDU	
Derivation Path:		
Comments:	A CC RELEASE message containing cause #96, used to match received CC RELEASE message.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?101101'B	
cau	Cause_04iei	
cau2	*	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Release_02	
PDU Type:	REL_PDU	
Derivation Path:		
Comments:	A CC RELEASE message to match any received CC RELEASE message.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?101101'B	
cau	*	
cau2	*	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Release_03(Ti:TI)	
PDU Type:	REL_PDU	
Derivation Path:		
Comments:	A CC RELEASE message containing cause #31 to be sent to the MS.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00101101'B	
cau	Cause_11	
cau2	-	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Release_05(Ti:TI; Cau:CAU)	
PDU Type:	REL_PDU	
Derivation Path:		
Comments:	A CC RELEASE message.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?101101'B	
cau	Cau	
cau2	Cause_14 IF_PRESENT	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Release_06(Ti:TI)	
PDU Type:	REL_PDU	
Derivation Path:		
Comments:	A CC RELEASE message containing cause #16 and second cause #102 to be sent to the MS.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00101101'B	
cau	Cause_01iei	
cau2	Cause_23	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Release_08(Ti:TI)	
PDU Type:	REL_PDU	
Derivation Path:		
Comments:	A CC RELEASE message with mandatory IE's only used in structured procedures test .	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00101101'B	
cau	-	
cau2	-	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Release_10(Ti:TI)	
PDU Type:	REL_PDU	
Derivation Path:		
Comments:	A CC RELEASE message used to match any received CC RELEASE message with a controllable TI.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?101101'B	
cau	*	
cau2	*	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmpRcv(Ti :TI; cau:CAU)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	A RELEASE COMPLETE message containing cause value = #81. ms -> n	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'0?101010'B	
cau	cau	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_02(ti:TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	A RELEASE COMPLETE message containing cause value #16. n -> ms	
Field Name	**Field Value**	**Comments**
ti	ti	
cc_sspd	'0011'B	
mt	'00101010'B	
cau	Cause_26	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_03(Ti:TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	To match any received RELEASE message.	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'0?101010'B	
cau	*	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_04(Ti:TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	A RELEASE COMPLETE message containing cause value #1. n -> ms	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'00101010'B	
cau	Cause_10	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_07(Ti:TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	A RELEASE COMPLETE message containing cause value = #88. ms -> n	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'0?101010'B	
cau	Cause_27	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_08(Ti :TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	A RELEASE COMPLETE message containing mandatory IEs only. n -> ms	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'00101010'B	
cau	-	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_09(Ti :TI; fie:FIE)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	A RELEASE COMPLETE message containing facility IE. n -> ms	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'1011'B	
mt	'00101010'B	
cau	-	
fie	fie	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_10(Ti:TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	A RELEASE COMPLETE message containing cause value = #17. ms -> n	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'0?101010'B	
cau	Cause_17	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_25(Ti:TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	To match a received RELEASE message which may or may not contain facility IE	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'0?101010'B	
cau	*	
fie	facilityIErcviei(FwdChAdvRslt_01) IF_PRESENT	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_42(Ti:TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	To send a RELEASE COMPLETE message containing cause 'facility rejected' and without FIE	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'1011'B	
mt	'00101010'B	
cau	Cause_28	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_47(Ti:TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	To send a RELEASE COMPLETE message without cause and without FIE	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'1011'B	
mt	'00101010'B	
cau	-	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_49(Ti:TI; Invkid:OCTETSTRING)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	To receive a RELEASE COMPLETE message containing Return Error for UnstructuredSS-Notify with the error code USSD Busy	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'1011'B	
mt	'0?101010'B	
cau	*	
fie	facilityIErcviei(NotificationSS_09(Invkid))	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_50(Ti:TI; Invkid:OCTETSTRING)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	To receive a RELEASE COMPLETE message containing Return Error for UnstructuredSS-Request with the error code USSD Busy	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'1011'B	
mt	'0?101010'B	
cau	*	
fie	facilityIErcviei(USSDReq_05(Invkid))	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_52(ti:TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	A RELEASE COMPLETE message containing cause value #81 and TI = '1110'B. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	ti	
cc_sspd	'0011'B	
mt	'0?101010'B	
cau	Cause_22	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_inv_02(ti:TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	An invalid RELEASE COMPLETE message containing unknown optional IEI	
Field Name	**Field Value**	**Comments**
ti	ti	
cc_sspd	'0011'B	
mt	'00101010'B	
cau	Cause_08	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseReq_inv_01(Ti:TI)	
PDU Type:	REL_PDU	
Derivation Path:		
Comments:	An invalid RELEASE message containing unknown optional IE.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00101101'B	
cau	Cause_09	
cau2	-	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	RRStatus_01	
PDU Type:	RRST_PDU	
Derivation Path:		
Comments:	A RR STATUS message containing any RR cause.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010010'B	
rrcau	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	RRStatus_02	
PDU Type:	RRST_PDU	
Derivation Path:	RRStatus_01.	
Comments:	RR STATUS message containing cause value #96--invalid mandatory information	
Field Name	**Field Value**	**Comments**
rrcau	'01100000'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	RRStatus_03	
PDU Type:	RRST_PDU	
Derivation Path:	RRStatus_01.	
Comments:	A RR STATUS message containing cause value #97--message type non-existent or not implemented	
Field Name	**Field Value**	**Comments**
rrcau	'01100001'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_01(ti:TI)	
PDU Type:	SETUP_MT_PDU	
Derivation Path:		
Comments:	A SETUP message containing speech bearer capability. n -> ms.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00000101'B	
bcri	-	
bcap1	Bcap_Speech_MT	
bcap2	-	
fie	-	
pi	-	
sig	-	
cgpn	-	
cgps	-	
cdpn	-	
cdps	-	
llcri	-	
llcmp1	-	
llcmp2	-	
hlcri	-	
hlcmp1	-	
hlcmp2	-	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_02(bcap: BCAP)	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	A SETUP message containing speech bearer capability. ms -> n	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	-	
bcap1	bcap	
bcap2	-	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	*	
llcmp1	*	
llcmp2	*	
hlcri	*	
hlcmp1	*	
hlcmp2	*	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_03(ti:TI)	
PDU Type:	SETUP_MT_PDU	
Derivation Path:		
Comments:	A SETUP message with mandatory IE's only. This is default for BIBO testing.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00000101'B	
bcri	-	
bcap1	-	
bcap2	-	
fie	-	
pi	-	
sig	-	
cgpn	-	
cgps	-	
cdpn	-	
cdps	-	
llcri	-	
llcmp1	-	
llcmp2	-	
hlcri	-	
hlcmp1	-	
hlcmp2	-	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_04(bcap: BCAP)	
PDU Type:	ESETUP_PDU	
Derivation Path:		
Comments:	A Emergency call SETUP message with/without speech bearer capability. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?001110'B	
bcap	bcap IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_10_UDI(bcap: BCAP; llc1:LLCMP; hlc1:HLCMP)	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	A SETUP message containing bearer capability supported by the mobile station, used for Unrestricted digital information transfer capability. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	-	
bcap1	bcap	
bcap2	-	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	-	
llcmp1	llc1	
llcmp2	-	
hlcri	-	
hlcmp1	hlc1 IF_PRESENT	
hlcmp2	-	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	CallCntrlCap IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_10_3100(bcap: BCAP; llc1:LLCMP; hlc1:HLCMP)	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	A SETUP message containing bearer capability supported by the mobile station. used for 3.1 kHz audio information transfer capability. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	-	
bcap1	bcap	
bcap2	-	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	-	
llcmp1	llc1 IF_PRESENT	
llcmp2	-	
hlcri	-	
hlcmp1	hlc1 IF_PRESENT	
hlcmp2	-	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	CallCntrlCap IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_11_3100(bcri1: RPI; bcap, bcap2nd: BCAP; llc1, llc2:LLCMP; llcri1, hlcri1:RPI; hlc1, hlc2:HLCMP)	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	A SETUP message containing two bearer capabilities supported by the mobile station, used for 3.1 kHz audio information transfer capability. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	bcri1	
bcap1	bcap	
bcap2	bcap2nd	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	llcri1 IF_PRESENT	
llcmp1	llc1 IF_PRESENT	
llcmp2	llc2 IF_PRESENT	
hlcri	hlcri1 IF_PRESENT	
hlcmp1	hlc1 IF_PRESENT	
hlcmp2	hlc2 IF_PRESENT	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	CallCntrlCap IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_20(par: BCAP; ti:TI)	
PDU Type:	SETUP_MT_PDU	
Derivation Path:		
Comments:	A SETUP message containing bearer capability supported by the mobile station.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00000101'B	
bcri	-	
bcap1	par	
bcap2	-	
fie	-	
pi	-	
sig	-	
cgpn	-	
cgps	-	
cdpn	-	
cdps	-	
llcri	-	
llcmp1	-	
llcmp2	-	
hlcri	-	
hlcmp1	-	
hlcmp2	-	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_21(par1:RPI; par2, par3 :BCAP; ti:TI)	
PDU Type:	SETUP_MT_PDU	
Derivation Path:		
Comments:	A SETUP message containing two bearer capabilities supported by the mobile station.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00000101'B	
bcri	par1	
bcap1	par2	
bcap2	par3	
fie	-	
pi	-	
sig	-	
cgpn	-	
cgps	-	
cdpn	-	
cdps	-	
llcri	-	
llcmp1	-	
llcmp2	-	
hlcri	-	
hlcmp1	-	
hlcmp2	-	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_inv_01(ti:TI)	
PDU Type:	SETUP_MT_PDU	
Derivation Path:		
Comments:	A SETUP message with ti_f value = 1 as an invalid message.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00000101'B	
bcri	-	
bcap1	-	
bcap2	-	
fie	-	
pi	-	
sig	-	
cgpn	-	
cgps	-	
cdpn	-	
cdps	-	
llcri	-	
llcmp1	-	
llcmp2	-	
hlcri	-	
hlcmp1	-	
hlcmp2	-	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_inv_02(ti:TI)	
PDU Type:	SETUP_MT_PDU	
Derivation Path:		
Comments:	An invalid SETUP message with arbitrary spare bits	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00000101'B	
bcri	-	
bcap1	-	
bcap2	-	
fie	-	
pi	-	
sig	Signal_01	
cgpn	Cgpn_01	
cgps	Cgps_01	
cdpn	Cdpn_01	
cdps	Cdps_01	
llcri	-	
llcmp1	-	
llcmp2	-	
hlcri	-	
hlcmp1	-	
hlcmp2	-	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SMSCB_01(sernum: SERIAL_NUMBER)	
PDU Type:	SMSCB_PDU	
Derivation Path:		
Comments:	To send a SMSCB message, first block	
Field Name	**Field Value**	**Comments**
blocktype	Blocktype_01('0000'B, '0'B)	
serial_number	sernum	
message_id	'0000'O	
dcs	Tpdcs_05	
page_param	'00010001'B	
message_contents	OC_CodeSMSCBMessage(1, 16)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SMSCB_02(seqnum, lb: BITSTRING; firstoct: INTEGER; lastoct: INTEGER)	
PDU Type:	SMSCB_PDU	
Derivation Path:		
Comments:	To send a SMSCB message, second to fourth block (depending on sequence number)	
Field Name	**Field Value**	**Comments**
blocktype	Blocktype_01(seqnum, lb)	
serial_number	-	
message_id	-	
dcs	-	
page_param	-	
message_contents	OC_CodeSMSCBMessage(firstoct, lastoct)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SetupInd_01	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	To match any received SETUP message	
Field Name	**Field Value**	**Comments**
ti	TI_09	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	*	
bcap1	?	
bcap2	*	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	*	
llcmp1	*	
llcmp2	*	
hlcri	*	
hlcmp1	*	
hlcmp2	*	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	CallCntrlCap IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	StartDtmf_01(Ti:TI; character:IA5String)	
PDU Type:	START_DTMF_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?110101'B	
kpf	KeyPad_01(character)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	StartDtmf_02(Ti:TI)	
PDU Type:	START_DTMF_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?110101'B	
kpf	KeyPad_02	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	StartDtmfAck_01(Ti:TI; character:IA5String)	
PDU Type:	START_DTMF_ACK_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00110110'B	
kpf	KeyPad_01(character)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	StartDtmfRej_01(Ti:TI)	
PDU Type:	START_DTMF_REJ_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00110111'B	
cau	Cause_15	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	StopDtmf_01(Ti:TI)	
PDU Type:	STOP_DTMF_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?110001'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	StopDtmfAck_01(Ti:TI)	
PDU Type:	STOP_DTMF_ACK_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00110010'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SyncInfor(bcc:BCC; ncc:NCC)	
PDU Type:	SCHINFO_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ncc	ncc	
bcc	bcc	
t1	?	
t2	?	
t3_	?	
Detailed Comments:	1. The values of t1, t2 and t3_ shall be correctly set by the L2 simulation module when this message is sent to air on the synchronization channel. It is assumed that there is a set of timebase counters in the L2 module, these counters run continuously and keep the correct timeslot number SN and TDMA frame number FN as long as the test system is poweron , the T1, T2, T3' (values for t1, t2 and t3_) can be derived from these counters.	

PDU Constraint Declaration		
Constraint Name:	SysInf1(cchd:CCHD; maxtx:B_2; txint:B_4; re:B_1)	
PDU Type:	SYSINFO1_PDU	
Derivation Path:		
Comments:	Default parameters for cell A in RR testing of GSM 900	
Field Name	**Field Value**	**Comments**
l2_pl	'55'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011001'B	
cchd	cchd	
rachcp	RachCntrlPara(maxtx, txint, re)	
si1roct	'2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf2(bcchfl: NCD; maxtx:B_2; txint:B_4; re:B_1; nccp:NCCP)	
PDU Type:	SYSINFO2_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION TYPE2 containing default neighbour cells description	
Field Name	**Field Value**	**Comments**
l2_pl	'59'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011010'B	
bcchfl	bcchfl	
nccp	nccp	
rachcp	RachCntrlPara(maxtx, txint, re)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf2bis(bcchflegsm: NCD)	
PDU Type:	SYSINFO2bis_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION 2bis in cell A under EGSM with the ARFCN list = {988, 990, 1003}.	
Field Name	**Field Value**	**Comments**
l2_pl	'55'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00000010'B	
xbcchfl	bcchflegsm	
rachcp	RachCntrlPara_r01	
si2bisroct	'00'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf3(ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
PDU Type:	SYSINFO3_PDU	
Derivation Path:		
Comments:	CCCH combined or not with SDCCH	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011011'B	
ci	ci	
lai	LocAreaId(mcc, mnc, lac)	
ccd	ccd	
co	co	
csp	CellSelPara(crh, mtmc, neci)	
rachcp	RachCntrlPara(maxtx, txint, re)	
si3roct	SI3RestOctetDef	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf3_inv_01(mcc, mnc, lac:OCTETSTRING)	
PDU Type:	SYSINFO3_PDU	
Derivation Path:		
Comments:	An invalid SYSTEM INFORMATION TYPE 3 message containing rest octets which are not all '2B'O	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011011'B	
ci	C_ci_cellA	
lai	LocAreaId(mcc, mnc, lac)	
ccd	CntrlChDscrp_inv	
co	CellOpt_04	
csp	CellSelPara_01	
rachcp	RachCntrlPara_01	
si3roct	SI3RestOctet_inv	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf4(mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
PDU Type:	SYSINFO4_PDU	
Derivation Path:		
Comments:	A SYSTEM INFORMATION TYPE 4 message containing default values	
Field Name	**Field Value**	**Comments**
l2_pl	'31'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011100'B	
lai	LocAreaId(mcc, mnc, lac)	
csp	CellSelPara(crh, mtmc, neci)	
rachcp	RachCntrlPara(maxtx, txint, re)	
cbchd	-	
cbchma	-	
si4roct	SI4RestOctetDef	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf4_CBMS(mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; cbchd:CHD)	
PDU Type:	SYSINFO4_PDU	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 4 message for SMSCB with the values or GSM 11.10, 34.3	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011100'B	
lai	LocAreaId(mcc, mnc, lac)	
csp	CellSelPara(crh, mtmc, neci)	
rachcp	RachCntrlPara(maxtx, txint, re)	
cbchd	cbchd	
cbchma	-	
si4roct	SI4RestOctet_CBMS	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf4_inv_01(mcc, mnc, lac:OCTETSTRING)	
PDU Type:	SYSINFO4_PDU	
Derivation Path:		
Comments:	An invalid SYSTEM INFORMATION message containing rest octets which are not all '2B'O	
Field Name	**Field Value**	**Comments**
l2_pl	'31'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011100'B	
lai	LocAreaId(mcc, mnc, lac)	
csp	CellSelPara_01	
rachcp	RachCntrlPara_01	
cbchd	-	
cbchma	-	
si4roct	SI4RestOctet_inv	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf5(bcchfl: NCD)	
PDU Type:	SYSINFO5_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION TYPE 5 containing default neighbour cells description for GSM900	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00011101'B	
bcchfl	bcchfl	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf5bis(bcchfl: NCD)	
PDU Type:	SYSINFO5bis_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION TYPE 5bis containing partial neighbour cells description with 1 frequency for DCS1800.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00000101'B	
xbcchfl	bcchfl	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf6(ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; nccp:NCCP)	
PDU Type:	SYSINFO6_PDU	
Derivation Path:		
Comments:	A SYSTEM INFORMATION TYPE 6 message containing default parameters for RR testing.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00011110'B	
ci	ci	
lai	LocAreaId(mcc, mnc, lac)	
co	co	
nccp	nccp	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	TmsiReallocCmd(par:MI; mcc, mnc, lac: OCTETSTRING)	
PDU Type:	TMSIRE_CMD_PDU	
Derivation Path:		
Comments:	TMSI REALLOCATION COMMAND message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00011010'B	
lai	LocAreaId(mcc, mnc, lac)	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	TmsiReallocComp	
PDU Type:	TMSIRE_COM_PDU	
Derivation Path:		
Comments:	TMSI REALLOCATION COMPLETE message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?011011'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	UndefCC_02(Ti:TI)	
PDU Type:	CONN_PDU	
Derivation Path:		
Comments:	This is an undefined CC message	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00111111'B	
fie	-	
pi	-	
cnn	-	
cns	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	UndefMM_01	
PDU Type:	ID_RES_PDU	
Derivation Path:		
Comments:	This is an undefined MM message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00100101'B	
mi	Mi_05	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	UndefRR_01	
PDU Type:	PART_REL_PDU	
Derivation Path:		
Comments:	An undefined RR message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00100101'B	
chd	'02E090'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	UnknownMsg_01(ti:TI)	
PDU Type:	CCST_ENQ_PDU	
Derivation Path:		
Comments:	CC STATUS ENQUIRY alike unknown message	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0000'B	
mt	'00110100'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	UnknownMsg_02(ti:TI)	
PDU Type:	CCST_ENQ_PDU	
Derivation Path:		
Comments:	CC STATUS ENQUIRY alike unknown message with arbitrary transaction ID.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0000'B	
mt	'00110100'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CpDataAckPdu_01(ti:TI)	
PDU Type:	CPDATA_ACK_PDU	
Derivation Path:		
Comments:	CP DATA ACKNOWLEDGE ms -> n	
Field Name	**Field Value**	**Comments**
ti	ti	
smspd	'1001'B	
mt	'00000100'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CpDataAckPdu_02(ti :TI)	
PDU Type:	CPDATA_ACK_PDU	
Derivation Path:		
Comments:	CP DATA ACKNOWLEDGE n -> ms	
Field Name	**Field Value**	**Comments**
ti	ti	
smspd	'1001'B	
mt	'00000100'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CpDataAckPdu_03(ti:TI)	
PDU Type:	CPDATA_ACK_PDU	
Derivation Path:		
Comments:	CP DATA ACKNOWLEDGE n -> ms, MO-SMS	
Field Name	**Field Value**	**Comments**
ti	ti	
smspd	'1001'B	
mt	'00000100'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CpErrPdu_01(ti:TI)	
PDU Type:	CPERR_PDU	
Derivation Path:		
Comments:	CP ERROR n -> ms GSM 04.11	
Field Name	**Field Value**	**Comments**
ti	ti	
smspd	'1001'B	
mt	'00010000'B	
cp_cause	'11'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CpDataPdu_01(ti:TI; cpdat:CPDATA)	
PDU Type:	CP_DATA_PDU	
Derivation Path:		
Comments:	CP DATA n -> ms, RP DATA, TP-DCS set to 0 GSM 04.11	
Field Name	**Field Value**	**Comments**
ti	ti	
smspd	'1001'B	
mt	'00000001'B	
CPdata	cpdat	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CpDataPdu_02(ti:TI; cpdat:CPDATA)	
PDU Type:	CP_DATA_PDU	
Derivation Path:		
Comments:	CP DATA ms -> n, RP Acknowledge GSM 04.11	
Field Name	**Field Value**	**Comments**
ti	ti	
smspd	'1001'B	
mt	'00000001'B	
CPdata	cpdat	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CpDataPdu_03(cpdat:CPDATA)	
PDU Type:	CP_DATA_PDU	
Derivation Path:		
Comments:	CP DATA ms -> n, RP data, TP-DCS set to 0, no status report requested GSM 04.11	
Field Name	**Field Value**	**Comments**
ti	TI_09	
smspd	'1001'B	
mt	'00000001'B	
CPdata	cpdat	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CpDataPdu_04	
PDU Type:	CP_DATA_PDU	
Derivation Path:		
Comments:	CP DATA n -> ms, RP Acknowledge GSM 04.11	
Field Name	**Field Value**	**Comments**
ti	TI_09	
smspd	'1001'B	
mt	'00000001'B	
CPdata	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CpDataPdu_any	
PDU Type:	CP_DATA_PDU	
Derivation Path:		
Comments:	CP DATA ms -> n GSM 04.11	
Field Name	**Field Value**	**Comments**
ti	?	
smspd	'1001'B	
mt	'00000001'B	
CPdata	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CpDataPdu_16(tpoa1: BCDN; rpoa_mt: BCDN; smtype: INTEGER; text: IA5String; ti:TI; rpmr: MR; timezone:TZONES)	
PDU Type:	CP_DATA_PDU	
Derivation Path:		
Comments:	CP DATA n -> ms, RP DATA, TP-DCS set to 0 GSM 04.11	
Field Name	**Field Value**	**Comments**
ti	ti	
smspd	'1001'B	
mt	'00000001'B	
CPdata	CpData_15(tpoa1, rpoa_mt, smtype, text, rpmr, timezone)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CpDataPdu_17(tpoa1: BCDN; rpoa_mt: BCDN; text: IA5String; ti:TI; rpmr: MR; timezone:TZONES)	
PDU Type:	CP_DATA_PDU	
Derivation Path:		
Comments:	CP DATA n -> ms, RP DATA, TP-DCS set to 0 GSM 04.11	
Field Name	**Field Value**	**Comments**
ti	ti	
smspd	'1001'B	
mt	'00000001'B	
CPdata	CpData_16(tpoa1, rpoa_mt, text, rpmr, timezone)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CpDataPdu_18(tpda, rpda:BCDN; tpud:TPUD)	
PDU Type:	CP_DATA_PDU	
Derivation Path:		
Comments:	CP DATA ms -> n, RP data, TP-DCS set to 0, no status report requested GSM 04.11	
Field Name	**Field Value**	**Comments**
ti	TI_09	
smspd	'1001'B	
mt	'00000001'B	
CPdata	CpData_17(tpda, rpda,tpud)	
Detailed Comments:		

Dynamic Part

Test Cases

Test Group General

Test Case Dynamic Behaviour					
Test Case Name:		TC_11_1_1			
Group:		GSM_L3_MS_v4220/General/			
Purpose:		1. To verify that the MS, for the case of the Multinumbering scheme or ISDN, accepts a SETUP message, where the Information Elements for Bearer Capability are compatible with the Bearer Services / Teleservices declared as supported by the MS, by sending a CALL CONFIRMED message. This is verified for all Mobile Terminated Bearer Services / Teleservices declared as supported by the MS. 2. To verify that the MS in the "Null" state, U0, when receiving a SETUP message containing incompatible Information Elements for Bearer Capability will respond with a RELEASE COMPLETE message. This is verified for all Mobile Terminated Bearer Services / Teleservices not declared as supported by the MS.			
Default:		OtherEventsFail			
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		START T_guard(1800)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_SpecialCase := C_Yes)			
4		+testTS11			
5		+testBs2x			
6		+testBS3x			
7		+testBS61			
8		+testBS81			
9		+testTS62			
10		[NOT TSPC_Serv_TS62 OR TSPC_Serv_TS61]			
11		+testTS61			
12		[TSPC_Serv_TS62 AND NOT TSPC_Serv_TS61]			
		testTS11			
13		+test(C_Telephony, TSPC_Serv_TS11)			
		testTS61			
14		+test(C_AltSpchFax, TSPC_Serv_TS61)			
		testTS62			
15		+test(C_AutoFax, TSPC_Serv_TS62)			
		testBs2x			
16		+test(C_Async300, TSPC_Serv_BS21)			
17		+test(C_Async1200, TSPC_Serv_BS22)			
18		+test(C_Async2400, TSPC_Serv_BS24)			
19		+test(C_Async4800, TSPC_Serv_BS25)			

20		+test(C_Async9600, TSPC_Serv_BS26)			
		testBS3x			
21		+test(C_Sync1200, TSPC_Serv_BS31)			
22		+test(C_Sync2400, TSPC_Serv_BS32)			
23		+test(C_Sync4800, TSPC_Serv_BS33)			
24		+test(C_Sync9600, TSPC_Serv_BS34)			
		testBS61			
25		+test(C_AltSpchData, TSPC_Serv_BS61)			
		testBS81			
26		+test(C_SpchData, TSPC_Serv_BS81)			
		test(srv:SERVICES; supported:BOOLEAN)			
27		+BasicServiceMT(srv, C_Full)			
28		+SetupAndCheck(supported)			
29		[TCV_2ndtest AND supported]			
30		(TCV_Setup_mt := TCV_Setup_mt1)			
31		+local_IFsetup(srv)			
32		+SetupAndCheck(TRUE)			
33		[NOT TCV_2ndtest OR NOT supported]			
		SetupAndCheck(supported:BOOLEAN)			
34		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
35		+Authentication(TCV_ch, TSPX_CKSNDef, TSPX_RANDDef)			
36		+Ciphering_on(TCV_ch)			
37		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		
38		+check(supported)			
39		+PostMainLinkRel(TCV_ch)			
40		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			3.
		check(supported:BOOLEAN)			
41		[supported]			1.
42	L_0001	L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))	(P)	
43		[NOT supported]			2.
44	L_0002	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_07(TI_01))	(P)	
		local_IFsetup(srv:SERVICES)			
45		[(srv = C_AltSpchFax) OR (srv = C_AutoFax)]			
46		(TCV_Null := OO_IFsetup_TS6x_MT(srv, TCV_ur2))			
47		[(srv = C_Async300) OR (srv = C_Async1200) OR (srv = C_Async2400) OR (srv = C_Async4800) OR (srv = C_Async9600)]			
48		(TCV_Null := OO_IFsetup_BS2x_MT(srv, TCV_itc))			
49		[(srv = C_Sync1200) OR (srv = C_Sync2400) OR (srv = C_Sync4800) OR (srv = C_Sync9600)]			
50		(TCV_Null := OO_IFsetup_BS3x_MT(srv, TCV_itc, TCV_sacp2))			
51		[(srv = C_AltSpchData) OR (srv = C_SpchData)]			
52		(TCV_Null := OO_IFsetup_BS61orBS81_MT(srv, TCV_ur2, TCV_sa2))			

Detailed Comments:
1. To test the supported basic service.
2. To test the non-supported basic service.
3. To set the channel back to non-ciphering mode for next test execution.

Test Case Dynamic Behaviour

Test Case Name: TC_11_1_2

Group: GSM_L3_MS_v4220/General/

Purpose: 1. To verify that the MS generates a SETUP message which includes a single or multiple Bearer Capability and a single LLC, according to the actual configuration on the MS.

This is verified for all Mobile Originated Bearer Services / Teleservices described in GSM 07.01 and declared as supported by the MS.

2. To verify that the MS includes a correctly encoded Repeat Indicator if it includes multiple Bearer Capabilities in the SETUP message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(6000)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+testTS11			
4		+testTS12			
5		+testTS61			
6		+testTS62			
7		+testBS21			
8		+testBS22			
9		+testBS23			
10		+testBS24			
11		+testBS25			
12		+testBS26			
13		+continue			
		continue			
14		+testBS31			
15		+testBS32			
16		+testBS33			
17		+testBS34			
18		+testBS41			
19		+testBS42			
20		+testBS43			
21		+testBS44			
22		+testBS45			
23		+testBS46			
24		+testBS51			
25		+testBS52			
26		+testBS53			
27		+testBS61			
28		+testBS81			
		testTS11			
29		[TSPC_Serv_TS11]			
30		+check(C_Telephony)			TS11
31		[NOT TSPC_Serv_TS11]			
		testTS12			
32		[TSPC_Serv_TS12]			
33		+check(C_EmgCall)			TS12
34		[NOT TSPC_Serv_TS12]			

		testTS61			
35		[TSPC_Serv_TS61]			
36		+check(C_AltSpchFax)			TS61
37		[NOT TSPC_Serv_TS61]			
		testTS62			
38		[TSPC_Serv_TS62]			
39		+check(C_AutoFax)			TS62
40		[NOT TSPC_Serv_TS62]			
		testBS21			
41		[TSPC_Serv_BS21]			
42		+check(C_Async300)			BS21
43		[NOT TSPC_Serv_BS21]			
		testBS22			
44		[TSPC_Serv_BS22]			
45		+check(C_Async1200)			BS22
46		[NOT TSPC_Serv_BS22]			
		testBS23			
47		[TSPC_Serv_BS23]			
48		+check(C_Async120075)			BS23
49		[NOT TSPC_Serv_BS23]			
		testBS24			
50		[TSPC_Serv_BS24]			
51		+check(C_Async2400)			BS24
52		[NOT TSPC_Serv_BS24]			
		testBS25			
53		[TSPC_Serv_BS25]			
54		+check(C_Async4800)			BS25
55		[NOT TSPC_Serv_BS25]			
		testBS26			
56		[TSPC_Serv_BS26]			
57		+check(C_Async9600)			BS26
58		[NOT TSPC_Serv_BS26]			
		testBS31			
59		[TSPC_Serv_BS31]			
60		+check(C_Sync1200)			BS31
61		[NOT TSPC_Serv_BS31]			
		testBS32			
62		[TSPC_Serv_BS32]			
63		+check(C_Sync2400)			BS32
64		[NOT TSPC_Serv_BS32]			
		testBS33			
65		[TSPC_Serv_BS33]			
66		+check(C_Sync4800)			BS33
67		[NOT TSPC_Serv_BS33]			
		testBS34			
68		[TSPC_Serv_BS34]			
69		+check(C_Sync9600)			BS34
70		[NOT TSPC_Serv_BS34]			
		testBS41			
71		[TSPC_Serv_BS41]			
72		+check(C_PAD300)			BS41
73		[NOT TSPC_Serv_BS41]			

		testBS42			
74		[TSPC_Serv_BS42]			
75		+check(C_PAD1200)			BS42
76		[NOT TSPC_Serv_BS42]			
		testBS43			
77		[TSPC_Serv_BS43]			
78		+check(C_PAD120075)			BS43
79		[NOT TSPC_Serv_BS43]			
		testBS44			
80		[TSPC_Serv_BS44]			
81		+check(C_PAD2400)			BS44
82		[NOT TSPC_Serv_BS44]			
		testBS45			
83		[TSPC_Serv_BS45]			
84		+check(C_PAD4800)			BS45
85		[NOT TSPC_Serv_BS45]			
		testBS46			
86		[TSPC_Serv_BS46]			
87		+check(C_PAD9600)			BS46
88		[NOT TSPC_Serv_BS46]			
		testBS51			
89		[TSPC_Serv_BS51]			
90		+check(C_Packet2400)			BS51
91		[NOT TSPC_Serv_BS51]			
		testBS52			
92		[TSPC_Serv_BS52]			
93		+check(C_Packet4800)			BS52
94		[NOT TSPC_Serv_BS52]			
		testBS53			
95		[TSPC_Serv_BS53]			
96		+check(C_Packet9600)			BS53
97		[NOT TSPC_Serv_BS53]			
		testBS61			
98		[TSPC_Serv_BS61]			
99		+check(C_AltSpchData)			BS61
100		[NOT TSPC_Serv_BS61]			
		testBS81			
101		[TSPC_Serv_BS81]			
102		+check(C_SpchData)			BS81
103		[NOT TSPC_Serv_BS81]			
		check(srv:SERVICES)			
104		+BasicServiceMO(srv, C_Full)			
105		+InitCall(srv)			
106		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
107		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
108		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
109		L?DL_EstInCmsRq	CMSerReq(		

			CMServiceReq_01)		
110		ACTIVATE(OtherEventsFail)			Restore Normal default
111		+Authentication(TCV_ch, TCV_cksn, TSPX_RANDDef)			
112		+Ciphering_on(TCV_ch)			
113		+ltree_SetupChk(srv)			
114		+PostMainLinkRel(TCV_ch)			
115		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
		ltree_SetupChk(srv1:SERVICES)			
116		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
117	L_0003	[(srv1 = C_Telephony) OR (srv1 = C_EmgCall)]		(P)	no further check for TS11, TS12
118	L_0004	[(srv1 = C_Packet2400) OR (srv1 = C_Packet4800) OR (srv1 = C_Packet9600)]		(P)	no further check for BS5x
119		[(srv1 = C_Async300) OR (srv1 = C_Async1200) OR (srv1 = C_Async120075) OR (srv1 = C_Async2400) OR (srv1 = C_Async4800) OR (srv1 = C_Async9600)]			
120		+local_BS2x_more_chk(srv1)			further check for BS2x
121		[(srv1 = C_Sync1200) OR (srv1 = C_Sync2400) OR (srv1 = C_Sync4800) OR (srv1 = C_Sync9600)]			
122		+local_BS3x_more_chk(srv1)			further check for BS3x
123		[(srv1 = C_PAD300) OR (srv1 = C_PAD1200) OR (srv1 = C_PAD120075) OR (srv1 = C_PAD2400) OR (srv1 = C_PAD4800) OR (srv1 = C_PAD9600)]			
124		+local_BS4x_more_chk(srv1)			further check for BS4x
125		[srv1 = C_AltSpchData]			
126		+local_BS61orBS81_more_chk(srv1, TSPX_BS_61_A_ur1_T_NT)			further check for BS61
127		[srv1 = C_SpchData]			
128		+local_BS61orBS81_more_chk(srv1, TSPX_BS_81_A_ur1_T_NT)			further check for BS81
129	L_0005	[(srv1 = C_AltSpchFax) OR (srv1 = C_AutoFax)]		(P)	no further check for TS61, TS62
		local_BS4x_more_chk(srv2:SERVICES)			
130		[srv2 = C_PAD300]			
131		+local_ce_strc_ir_chk(srv2, TSPX_BS_41_ce, TSPX_BS_41_T_NT)			
132		[srv2 = C_PAD1200]			
133		+local_ce_strc_ir_chk(srv2, TSPX_BS_42_ce, TSPX_BS_42_T_NT)			
134		[srv2 = C_PAD120075]			
135		+local_ce_strc_ir_chk(srv2, TSPX_BS_43_ce, TSPX_BS_43_T_NT)			
136		[srv2 = C_PAD2400]			
137		+local_ce_strc_ir_chk(srv2, TSPX_BS_44_ce, TSPX_BS_44_T_NT)			
138		[srv2 = C_PAD4800]			
139		+local_ce_strc_ir_chk(srv2, TSPX_BS_45_ce, TSPX_BS_45_T_NT)			
140		[srv2 = C_PAD9600]			
141		+local_ce_strc_ir_chk(srv2, TSPX_BS_46_ce, TSPX_BS_46_T_NT)			
		local_BS2x_more_chk(srv3:SERVICES)			
142		[srv3 = C_Async300]			
143		+local_ce_strc_ir_chk(srv3, TSPX_BS_21_ce, TSPX_BS_21_T_NT)			
144		[srv3 = C_Async1200]			
145		+local_ce_strc_ir_chk(srv3, TSPX_BS_22_ce, TSPX_BS_22_T_NT)			
146		[srv3 = C_Async120075]			
147		+local_ce_strc_ir_chk(srv3, TSPX_BS_23_ce, TSPX_BS_23_T_NT)			

148		[srv3 = C_Async2400]			
149		+local_ce_strc_ir_chk(srv3, TSPX_BS_24_ce, TSPX_BS_24_T_NT)			
150		[srv3 = C_Async4800]			
151		+local_ce_strc_ir_chk(srv3, TSPX_BS_25_ce, TSPX_BS_25_T_NT)			
152		[srv3 = C_Async9600]			
153		+local_ce_strc_ir_chk(srv3, TSPX_BS_26_ce, TSPX_BS_26_T_NT)			
		local_BS3x_more_chk(srv4:SERVICES)			
154	L_0006	[srv4 = C_Sync1200]		(P)	no further check
155		[srv4 = C_Sync2400]			
156	L_0007	[(NOT TSPX_BS_32_X32_T_NT) OR(TSPX_BS_32_itc1 <> C_3100Hz) OR(TSPX_BS_32_sacp1 <> C_X32) OR(TSPX_BS_32_X32_ce <> C_transparent)]		(P)	no further check
157		[TSPX_BS_32_X32_T_NT AND (TSPX_BS_32_itc1 = C_3100Hz) AND (TSPX_BS_32_sacp1 = C_X32) AND (TSPX_BS_32_X32_ce = C_transparent)]			
158		+local_ce_strc_ir_chk(srv4, TSPX_BS_32_X32_ce, TSPX_BS_32_X32_T_NT)			
159		[srv4 = C_Sync4800]			
160	L_0008	[(NOT TSPX_BS_33_X32_T_NT) OR(TSPX_BS_33_itc1 <> C_3100Hz) OR(TSPX_BS_33_sacp1 <> C_X32) OR(TSPX_BS_33_X32_ce <> C_transparent)]		(P)	no further check
161		[TSPX_BS_33_X32_T_NT AND (TSPX_BS_33_itc1 = C_3100Hz) AND (TSPX_BS_33_sacp1 = C_X32) AND (TSPX_BS_33_X32_ce = C_transparent)]			
162		+local_ce_strc_ir_chk(srv4, TSPX_BS_33_X32_ce, TSPX_BS_33_X32_T_NT)			
163		[srv4 = C_Sync9600]			
164	L_0009	[(NOT TSPX_BS_34_X32_T_NT) OR(TSPX_BS_34_itc1 <> C_3100Hz) OR(TSPX_BS_34_sacp1 <> C_X32) OR(TSPX_BS_34_X32_ce <> C_transparent)]		(P)	no further check
165		[TSPX_BS_34_X32_T_NT AND (TSPX_BS_34_itc1 = C_3100Hz) AND (TSPX_BS_34_sacp1 = C_X32) AND (TSPX_BS_34_X32_ce = C_transparent)]			
166		+local_ce_strc_ir_chk(srv4, TSPX_BS_34_X32_ce, TSPX_BS_34_X32_T_NT)			
		local_ce_strc_ir_chk(srv5:SERVICES; ce:B_2; both:BOOLEAN)			
167	L_0010	[(NOT both) OR (both AND(TCV_ce = C_notransparent))]		(P)	no further check needed
168		[both AND (TCV_ce = C_transparent)]			
169		[(srv5=C_AltSpchFax) OR (srv5=C_AltSpchData) OR (srv5=C_SpchData)]			
170		(TCV_ce := TCV_Setup_mo1.bcap2.ce, TCV_ir := TCV_Setup_mo1.bcap2.ir, TCV_strc := TCV_Setup_mo1.bcap2.strc)			
171		+local_temp_tree			
172		[C_Yes]			
173		(TCV_ce := TCV_Setup_mo1.bcap1.ce, TCV_ir := TCV_Setup_mo1.bcap1.ir, TCV_strc := TCV_Setup_mo1.bcap1.strc)			
174		+local_temp_tree			
		local_temp_tree			
175	L_0011	[TCV_ce = C_notransparent]		(F)	
176		[TCV_ce = C_transparent]			
177	L_0012	[TCV_strc <> C_Unstructured]		(F)	

178	L_0013	[TCV_strc = C_Unstructured]		(P)	
179		[(TCV_ce <> C_transparent) AND(TCV_ce <> C_notransparent)]			
180	L_0014	[(TCV_ir = C_ir_16kbs) AND (TCV_strc = C_SDUintegrity)]		(P)	
181	L_0015	[(TCV_ir <> C_ir_16kbs) OR(TCV_strc <> C_SDUintegrity)]		(F)	
		local_BS61orBS81_more_chk(srv6: SERVICES; both1:BOOLEAN)			
182	L_0016	[TCV_sa <> C_Asynchronous]		(P)	no further check needed
183		[TCV_sa =C_Asynchronous]			
184		+local_ce_strc_ir_chk(srv6, TCV_ce, both1)			

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_11_2

Group: GSM_L3_MS_v4220/General/

Purpose: 1. To verify that the MS, for the case of the Single Numbering Scheme, accepts a SETUP message, where the Information Elements for Brearer Capability and Lower and Higher Layer Compatibility are not present by sending a CALL CONFIRMED message which includes the single or multiple Bearer Capabilities, according to the actual configuration on the MS.

This is verified for one Mobile Terminated Bearer Services / Teleservices described in GSM 07.01 and declared as supported by the MS.

2. To verify that the MS includes a correctly encoded Repeat Indicator if it includes multiple Bearer Capabilities in the CALL CONFIRMED message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+CallCfmGen(TSPX_MTBscSvcD, TSPX_MTChRateD)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+RRmtcallprepare(TimingAdv(0), TSPX_RANDDef)			
5		L!DL_DatRqSetup	SetupSnd(TCV_ch, Setup_03(TI_02))		3.
6	L_0017	L?DL_DatInCallCo	CallCfm(TCV_CallCfm)	(P)	4.
7		+PostMainLinkRel(TCV_ch)			
8		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			

Detailed Comments:

1. To generate a CALL COMKFIRMED based on the required basic service and IXIT parameters, and ask operator to configurate the MS for the required basic service.
2. To setup physical channel for BCCH,CCCH and SDCCH4 channels.
3. To send a SETUP_PDU without bearer capabilities and lower and higher layer compatibilities.
4. To check whether the received CALL COMFIRM PDU is correct.

Test Case Dynamic Behaviour

Test Case Name: TC_11_3

Group: GSM_L3_MS_v4220/General/

Purpose: 1. To verify that an MS claiming to not support AOCC and in the outgoing call / U4 call delivered state, on receipt of a CONNECT message containing AOCC information acknowledges the CONNECT message but ignores and does not acknowledge the AOCC information sent within the CONNECT.

2. To verify that an MS claiming to not support AOCC and in the outgoing call / U4 call delivered state, on receipt of a FACILITY message containing AOCC information ignores and does not acknowledge the AOCC information contained within the FACILITY.

3. To verify that an MS claiming to not support AOCC and in the incoming call / U9 call confirmed state, on receipt of a FACILITY message containing AOCC information ignores and does not acknowledge the AOCC information contained within the FACILITY.

4. To verify that an MS claiming to not support AOCC and in the U10 call active state, on receipt of a FACILITY message containing AOCC information ignores and does not acknowledge the AOCC information contained within the FACILITY.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+BasicServiceMO(TSPX_MOBscSvcB, TSPX_MOChRateB)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+test1			2.
6		+test2			3.
7		+test3			4.
8		+test4			5.
		test1			
9		+ltree_estMOcall_U4			
10		L!DL_DatRqConn START T_dly(15000)	ConnSnd(TCV_chTch, Connect_05(TCV_TI, facilityIEtsndiei(FwdChAdvSS_01)))		
11		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
12	L_0018	L?DL_DatInFac CANCEL T_dly	FacilityRcv(FacilityPdu_05(TCV_TI0))	(F)	
13		+release(TCV_chTch)			
14	L_0019	?TIMEOUT T_dly		(P)	
15		+release(TCV_chTch)			
16	L_0020	L?DL_DatInFac CANCEL T_dly	FacilityRcv(FacilityPdu_05(TCV_TI0))	(F)	
17		+release(TCV_chTch)			
18	L_002	?TIMEOUT T_dly		(P)	

	1				
19		+release(TCV_chTch)			
		test2			
20		+ltree_estMOcall_U4			
21		L!DL_DatRqFac START T_dly(15000)	FacilitySnd(TCV_chTch, FacilityPdu_25(TCV_TI, facilityIEtsnd(FwdChAdvSS_01)))		
22	L_0022	L?DL_DatInFac CANCEL T_dly	FacilityRcv(FacilityPdu_05(TCV_TI0))	(F)	
23		+release(TCV_chTch)			
24	L_0023	?TIMEOUT T_dly		(P)	
25		+release(TCV_chTch)			
		test3			
26		+BasicServiceMT(TSPX_MTBscSvcA, TSPX_MTChRateA)			
27		+RRmtcallprepare(TimingAdv(0), TSPX_RANDDef)			
28		L!DL_DatRqSetup (TCV_TI.ti_f := '0'B, TCV_TI.ti_v := '000'B)	SetupSnd(TCV_ch, TCV_Setup_mt)		
29		L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))		
30		L!DL_DatRqFac START T_dly(15000)	FacilitySnd(TCV_ch, FacilityPdu_25(TI_02, facilityIEtsnd(FwdChAdvSS_01)))		
31	L_0024	L?DL_DatInFac CANCEL T_dly	FacilityRcv(FacilityPdu_05(TI_01))	(F)	
32		+release(TCV_ch)			
33	L_0025	?TIMEOUT T_dly		(P)	
34		+release(TCV_ch)			
		test4			
35		+BasicServiceMO(TSPX_MOBscSvcB, TSPX_MOChRateB)			
36		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
37		L!DL_DatRqFac START T_dly(15000)	FacilitySnd(TCV_chTch, FacilityPdu_25(TCV_TI, facilityIEtsnd(FwdChAdvSS_01)))		
38	L_0026	L?DL_DatInFac CANCEL T_dly	FacilityRcv(FacilityPdu_05(TCV_TI0))	(F)	
39		+PostMainLinkRel(TCV_chTch)			
40	L_0027	?TIMEOUT T_dly		(P)	
41		+PostMainLinkRel(TCV_chTch)			
		release(ch: LOGICCH)			
42		+PostMainLinkRel(ch)			
43		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
		ltree_estMOcall_U4			
44		+InitCall(TCV_Service)			
45		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
46		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.

47		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
48		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
49		ACTIVATE(OtherEventsFail)			Restore Normal default
50		+Authentication(TCV_ch, TSPX_CKSNDef, TSPX_RANDDef)			
51		+Ciphering_on(TCV_ch)			
52		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
53		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
54		L!DL_DatRqAlert	AlertSnd(TCV_ch, Alerting_01(TCV_TI))		
55		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
56		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			

Detailed Comments:

1. To setup physical channels for BCCH, CCCH, SDCCH4 and traffic channels.
2. To verify non-supported AOCC information carried by CONNECT message in the case of MO call state U4.
3. To verify non-supported AOCC information carried by FACILITY message in the case of MO call state U4.
4. To verify non-supported AOCC information carried by FACILITY message in the case of MT call state U9.
5. To verify non-supported AOCC information carried by FACILITY message in the case of MO call state U10.

Test Case Dynamic Behaviour

Test Case Name: TC_11_4

Group: GSM_L3_MS_v4220/General/

Purpose: To verify that an MS claiming to not support the Call Hold supplementary service and in the U10 call active state, reacts in the following manner when the appropriate call hold MMI command is entered:
- MS fails to put the first call on hold
- MS fails to place the second call.
- Optionally provides some indication to the user of an error.

Default: OtherEvents_01

Comments: other irrelevant mesages wil be discarded by the default tree.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcC, TSPX_MOChRateC)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
6		+InitCall(TCV_Service)			3.
7		START T_dly(3000)			
8		ACTIVATE(DTMFsig, OtherEvents_01)			Accept DTMF signalling
9	L_0028	L?DL_DatInHold CANCEL T_dly	HoldRcv(Holdpdu_01(TCV_TI0))	(F)	4.
10		+PostMainLinkRel(TCV_chTch)			
11	L_0029	?TIMEOUT T_dly		(P)	5.
12		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To setup physical channels for BCCH, CCCH, SDCCH4 and traffic channels.
2. To bring the MS into U10 state of mobile originating call.
3. To enter hold MMI command.
4. The MS sends out HOLD mesage, fail.
5. Within 3 seconds there is no HOLD message, pass.

Test Case Dynamic Behaviour

Test Case Name: TC_11_5

Group: GSM_L3_MS_v4220/General/

Purpose: To verify that an MS claiming to not support the MultiParty supplementary service and in the U10 call active state with one call and another call on hold, reacts in the following manner when the appropriate MultiParty MMI command is entered:

- Fails to combine the three parties in a MultiParty call.
- Optionally provides some indication to the user of an error.

Default: OtherEvents_01

Comments: other irrelevant mesages wil be discarded by the default tree.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcD, TSPX_MOChRateD)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
6		+InitCall(TCV_Service)			7.
7		ACTIVATE(DTMFsig, OtherEvents_01)			Accept DTMF signalling
8		L?DL_DatInHold	HoldRcv(Holdpdu_01(TCV_TI0))		
9		ACTIVATE(OtherEvents_01)			Restore Normal Default
10		L!DL_DatRqHoldAck	HoldAckSnd(TCV_chTch, HoldAckpdu_01(TCV_TI))		
11		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_01)		
12		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
13		+SetupRcvMo2(SetupInd_01)			
14		+continue			
		continue			
15		L!DL_DatRqCallProc	CallProc(TCV_chTch, TCV_CallProc)		
16		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI2))		
17		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI2))		
18		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(		3.

			TCV_TI1))		
19		(TCV_Null := OO_MptyCall())			4.
20		ACTIVATE(DTMFsig, OtherEvents_01)			Accept DTMF signalling
21		START T_dly(3000)			
22	L_0030	L?DL_DatInFac CANCEL T_dly	FacilityRcv(FacilityPdu_26(TCV_TI0, facilityIErcv(BldMptySS_01)))	(F)	5.
23		+PostMainLinkRel(TCV_chTch)			
24	L_0031	L?DL_DatInFac CANCEL T_dly	FacilityRcv(FacilityPdu_26(TCV_TI1, facilityIErcv(BldMptySS_01)))	(F)	5.
25		+PostMainLinkRel(TCV_chTch)			
26	L_0032	?TIMEOUT T_dly		(P)	6.
27		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup physical channels for BCCH, CCCH, SDCCH4 and traffic channels.
2. To bring the MS into U10 state of mobile originating call.
3. The first call is on hold and second call is in active state.
4. To enter MultiParty MMI command.
5. The MS sends out FACILITY message, fail.
6. Within 3 seconds there is no FACILITY message, pass.
7. To initiate the second call.

Test Case Dynamic Behaviour

Test Case Name: TC_11_6

Group: GSM_L3_MS_v4220/General/

Purpose: 1. To verify that an MS claiming to not support FDN and that has a SIM with FDN allocated and activated inserted in it refuses an attempt to make an outgoing call made by the user.

2. To verify that an MS claiming to not support FDN and that has a SIM with FDN allocated and activated inserted in it does not answer to paging.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcE, TSPX_MOChRateE)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		(TCV_Null := OO_SIM3Ins())			2.
5		+InitCall(TCV_Service)			3.
6		START T_dly(3000)			
7	L_0033	L?DL_RacInChRq CANCEL T_dly	ChReq(ChRequest_02)	(F)	
8		START T_dly(C_T_Wait)			4.
9		?TIMEOUT T_dly			
10		+continue			
11	L_0034	?TIMEOUT T_dly		(P)	
12		+continue			
		continue			
13		L!DL_UdatRqPg1Rq (DL_UdatRqPg1Rq.pgg := TCV_Pgg)	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		5.
14		START T_dly(3000)			
15	L_0035	L?DL_RacInChRq CANCEL T_dly	ChReq(ChRequest_02)	(F)	
16	L_0036	?TIMEOUT T_dly		(P)	6.

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4.
2. To insert the SIM with FDN allocated and activated, the power on the MS.
3. To attemp an outgoing CM connection.
4. To wait the MS back to idle.
5. To page the MS.
6. No CHANNEL REQUEST, pass

Test Group InitialTest

Test Case Dynamic Behaviour					
Test Case Name:	TC_26_2_1_1				
Group:	GSM_L3_MS_v4220/InitialTest/				
Purpose:	1) To verify that the MS answers to a PAGING message by sending a CHANNEL REQUEST message within 0.7 seconds after reception of the PAGING message. 2) To verify that the MS does not always use the same delay between reception of paging message and sending of the CHANNEL REQUEST message. If an MS uses a fixed delay, there is a high probability that different MSs of the same product series use the same delay. There would then be a high risk of collision.				
Default:	OtherEventsFail_01				
Comments:	The default tree OtherEventsFail_01 throws away any possible retransmitted channel request messages.				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		START T_guard(1800)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, TSPX_CcchConf1, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+localtree			
		localtree			
4		(TCV_Cnt :=0)			
5	body	REPEAT localtree1 UNTIL [TCV_Cnt = 200]			
6		(TCV_Res := OC_SaveAndProc1(TCV_Fk, C_PROC, TCV_Cnt, TCV_Upd))			1.
7	L_0037	[NOT TCV_Res]		(F)	2.
8	L_0038	[TCV_Res]		(P)	
		localtree1			
9		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
10		L?DL_RacInChRq(TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) START T_dly(C_T_Wait)	ChReq(ChRequest_17)		
11		(TCV_Fn1 := OM_ReturnFn(TCV_PgCh), TCV_Fk := OC_RachSlots(TCV_Fn1, TCV_Fn, TCV_Upd, 0))			4.
12	L_0039	[TCV_Fk < -9990]		(F)	7.
13		(TCV_Cnt := 200)			
14		CANCEL T_dly			
15		[TCV_Fk >= 0]			
16		(TCV_Res := OC_SaveAndProc1(TCV_Fk, C_SAVE, TCV_Cnt, TCV_Upd))			5.
17	L_0040	[NOT TCV_Res]		(F)	6.
18		(TCV_Cnt := 200)			
19		CANCEL T_dly			
20	L_0041	[TCV_Res]		(P)	
21		L!DL_UdatRqImmassRej (TCV_Cnt := TCV_Cnt+1)	ImmAssRej(TCV_agch,		

22		?TIMEOUT T_dly	ImmAsgnRej_01(TCV_Rr, TCV_Fn))		

Detailed Comments:

1. To analyse the delay of channel request msg.
2. The channel request messages are not spread equally.
3. To get the frame number on which paging request was sent.
4. To calculate the rach slots between paging request and channel request.
5. To record the delay of channel request msg.
6. MS is too slow to answer a paging msg.
7. Rach TDMA frame mapping is not correct.

Test Case Dynamic Behaviour

Test Case Name: TC_26_2_1_2

Group: GSM_L3_MS_v4220/InitialTest/

Purpose:

1) To verify that the MS spreads retransmission of a CHANNEL REQUEST message with equal probability on Tx-Integer time slots and correctly applies the fixed delay when the following conditions apply:
   - the CCCH is combined or not combined with SDCCHs;
   - the maximum number of retransmissions is equal to one of the following values: 1, 2, 4, 7, according to the value of TSPX_MaxRetrans.
   - Tx-Integer is put to any of the allowed values among those which are greater or equal to 6, according to the value of TSPX_Txint.

2) To verify that the MS retransmits exactly Max_Retrans times a CHANNEL REQUEST message if the network never responds to the CHANNEL REQUEST message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_K := ((230 + TSPX_MaxRetrans-1) / TSPX_MaxRetrans), TCV_T := 10 +((C_T_Wait/ 1000) + 1), TCV_T := (TCV_T * TCV_K) * 2)			1.
2		START T_guard(TCV_T)			
3		(TCV_Cnt1 := 0)			
4		REPEAT ltree_main UNTIL [TCV_Cnt1 = 2]			
		ltree_main			
5		[TCV_Cnt1 = 0]			
6		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, TSPX_Txint, TSPX_MaxRetrans, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
7		(TCV_S := OC_LookupS(TSPX_Txint, TCV_Upd))			2.
8		+localtree(TCV_Upd)			
9		(TCV_Cnt1 := TCV_Cnt1 + 1)			
10		[TCV_Cnt1 = 1]			
11		(TCV_slot := C_S0, TCV_tsc := C_BCC, TCV_Upd := C_NotCombined)			
12		+NonCombinedBCCH_A(C_E_default, C_MaxPwrLvlG, C_MaxPwrLvlD, C_arfcnA, C_arfcnAd, C_Immass, C_S0, C_BCC, TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
13		+SysInfoSending_nfh(C_SCH_A, C_BCCH_A_1, TSPX_Txint, TSPX_MaxRetrans, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_NCCP_2)			
14		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
15		+Varinit_fix(C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd,			

		TCV_Ccd0A, TSPX_IMSI)			
16		START T_dly(C_T_Wait)			
17		?TIMEOUT T_dly			
18		(TCV_S := OC_LookupS(TSPX_Txint, TCV_Upd))			2.
19		+localtree(TCV_Upd)			
20		(TCV_Cnt1 := TCV_Cnt1 + 1)			
		localtree(mode: BOOLEAN)			
21		(TCV_kcnt := 0, TCV_M :=0)			
22		REPEAT localtree1(mode) UNTIL [TCV_kcnt =TCV_K]			
23		(TCV_Res :=OC_InRang(TSPX_Txint, TSPX_MaxRetrans, TCV_M))			
24	L_0042	[TCV_Res]		(P)	
25	L_0043	[NOT TCV_Res]		(F)	
		localtree1(mode: BOOLEAN)			
26		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		3.
27		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn1 := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
28		(TCV_Cnt :=0)			
29		REPEAT localtree2(mode) UNTIL [TCV_Cnt = TSPX_MaxRetrans]			
30		+localtree3			
31		?TIMEOUT T_dly			
32		(TCV_kcnt := TCV_kcnt +1)			
		localtree2(mode: BOOLEAN)			
33		CANCEL T_dly			
34		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) START T_dly(C_T_Wait)	ChReq(ChRequest_17)		
35		(TCV_Fk := OC_RachSlots(TCV_Fn1, TCV_Fn, mode, 1), TCV_Fn1 := TCV_Fn, TCV_Cnt := TCV_Cnt+1)			4.
36	L_0044	[(TCV_Fk >= TCV_S) AND (TCV_Fk <= (TCV_S+TSPX_Txint- 1))]		(P)	5.
37		[TCV_Fk >= (TCV_S +(TSPX_Txint + 1) / 2)]			
38		(TCV_M := TCV_M + 1)			
39		[TCV_Fk <(TCV_S +(TSPX_Txint + 1) / 2)]			
40	L_0045	[(TCV_Fk < TCV_S) OR(TCV_Fk >(TCV_S+TSPX_Txint - 1))]		(F)	6.
		localtree3			
41	L_0046	[TCV_kcnt = TCV_K]		(P)	
42		[TCV_kcnt < TCV_K]			
43		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_02(TCV_Rr, TCV_Fn))		

Detailed Comments:

1. To setup the timeout value for guard timer, each excuton of one sequence= 10s.
2. To generate the required parameters.
3. To start the measuring.
4. To get the number of the CCCH RACH slots between the moment where the last CHANNEL REQUEST received and the reception of the new CHANNEL REQUEST.
5. The f(i, k) is in the set {S, S+1, ...S+T-1}.
6. The f(i, k) is not in the set {S, S+1, ...S+T-1}, fail.

Test Case Dynamic Behaviour

Test Case Name: TC_26_2_1_3

Group: GSM_L3_MS_v4220/InitialTest/

Purpose: To verify that an MS produces different random references for a CHANNEL REQUEST. If a MS always produces the same random reference, it makes possible that different MSs of the same product series produce the same random reference.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(360)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+localtree			
		localtree			
4		(TCV_Cnt := 0)			
5	body	REPEAT localtree1 UNTIL [TCV_Cnt = 7]			
6		(TCV_Res := OC_SaveAndProc3(TCV_Rr, C_PROC, TCV_Cnt))			1.
7	L_0047	[TCV_Res]		P	
8	L_0048	[NOT TCV_Res]		F	2.
		localtree1			
9		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
10		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf)	ChReq(ChRequest_17)		
11		(TCV_Null := OC_SaveAndProc3(TCV_Rr, C_SAVE,TCV_Cnt))			3.
12		L?DL_RacInChRq	ChReq(ChRequest_17)		4.
13		START T_dly(C_T_Wait)			5.
14		(TCV_Cnt := TCV_Cnt+1)			
15		?TIMEOUT T_dly			

Detailed Comments:

1. To analyse the distribution of the random references.
2. The random references are not randomly distributed.
3. To record the random reference.
4. The MS retransmits the channel request once more.
5. To wait long enough to guarantee that the MS is in service.

Test Case Dynamic Behaviour

Test Case Name: TC_26_2_2

Group: GSM_L3_MS_v4220/InitialTest/

Purpose: 1) To verify that the MS correctly performs IMSI detach/attach procedures when it is required by the network and upon deactivation/activation or SIM removal/insertion and does not perform these procedures when not required.

2) To verify that the mobile station acknowledges a re-allocated TMSI during IMSI attach.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3	body	+procedure1			
4		+procedure2			
5		+SetATT(C_TxInt_5, C_Max_1, C_ATT_1, C_NECI_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1)			1.
6		+procedures3			
7		+procedures4			
		procedure1			
8		+ImsiDetachNoReaction(5000, C_SIMIn)			2.
9		+ImsiAttachNoReaction(30000, C_SIMIn)			4.
		procedure2			
10		[TSPC_SIMRmv]			
11		(TCV_Null :=OO_SIMRmv())			6.
12		+NoReaction(5000)			3.
13		(TCV_Null := OO_SIMIns())			7.
14		+NoReaction(30000)			5.
15		[NOT TSPC_SIMRmv]			
		procedures3			
16		+MM_PwrOrSimOff(C_SIMIn)			
17		[(TSPC_DetachOnPwrDn) OR(TSPC_SwitchOnOff)]			
18		+imsidetach			
19		+MM_PwrOrSimOn(C_SIMIn)			
20		+locup(MiTmsi_02iei)			
21		[(NOT TSPC_DetachOnPwrDn) AND(NOT TSPC_SwitchOnOff)]			
22		+MM_PwrOrSimOn(C_SIMIn)			
23		+locup(MiTmsi_02iei)			
		procedures4			
24		[(TSPC_SIMRmv) AND(TSPC_DetachOnSIMRmv)]			
25		(TCV_Null :=OO_SIMRmv())			
26		+imsidetach			
27		(TCV_Null := OO_SIMIns())			
28		+locup(MiTmsi_01iei)			
29		[(NOT TSPC_SIMRmv) AND(TSPC_DetachOnPwrDn)]			

30		+MM_PwrOrSimOff(C_SIMIn)			
31		+imsidetach			
32		+MM_PwrOrSimOn(C_SIMIn)			
33		+locup(MiTmsi_01iei)			
34		[((NOT TSPC_SIMRmv) AND(NOT TSPC_DetachOnPwrDn)) OR((NOT TSPC_DetachOnSIMRmv) AND(TSPC_SIMRmv)) OR((NOT TSPC_DetachOnSIMRmv) AND(NOT TSPC_DetachOnPwrDn))]			
35		+MM_PwrOrSimOff(C_SIMIn)			
36		+MM_PwrOrSimOn(C_SIMIn)			
37		+locup(MiTmsi_01iei)			
		activitychk(t: INTEGER)			
38		START T_dly(t)			
39	L_0049	?TIMEOUT T_dly		(P)	
		locup(newmi:MI)			
40		+channelrequest			
41		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(C_imsi_attach))		9.
42		ACTIVATE(OtherEventsFail)			Restore Normal default
43		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(newmi, C_MCC_1, C_PLMN_1, TCV_lac))		10.
44	L_0050	L?DL_DatInTmsireCom	TmsiReallocCmp(TCV_ch)	(P)	
45		+channelrelease			
		imsidetach			
46		+channelrequest			
47	L_0051	L?DL_EstInImsidIn	ImsiDet(ImsiDetach_01)	(P)	8.
48		ACTIVATE(OtherEventsFail)			Restore Normal default
49		+channelrelease			
		channelrequest			
50		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
51		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
52		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
		channelrelease			
53		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
54		L?DL_RelIn	DLRelInd_01		

Detailed Comments:

1. To set ATT = 1 for procedures3 and procedures4.
2. To switch off or power down the MS.
3. The test system checks for 5 seconds that MS shall not initiate the IMSI detach procedure.
4. To switch on or power up the MS.
5. The test system checks for 30 seconds that MS shall not initiate the IMSI attach procedure.
6. If possible to remove the SIM.
7. To insert the SIM to the MS under test.
8. The MS initiate IMSI detach procedure.
9. The location updating type shall be IMSI attach
10. To allocate a new TMSI.

Test Case Dynamic Behaviour

Test Case Name: TC_26_2_3
Group: GSM_L3_MS_v4220/InitialTest/
Purpose: To verify that V(SD) is correctly set to 0 at the beginning of the establishment of the first RR connection and to verify that the MS handles correctly this variable in the special case of IDENTITY REQUEST messages, which are MM messages.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		TMSI paging ...
4		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
5		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
6		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
7		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
8		ACTIVATE(OtherEventsFail)			Restore Normal default
9		L!DL_DatRqIdRq	IDReq(TCV_ch, IDRequest_01('0001'B))		
10		L?DL_DatInIdRes (TCV_Mt := DL_DatInIdRes.msg.mt)	IDRes(IDResponse_01)		
11		+check1			1.
12		(TCV_Cnt := 0)			
13		REPEAT localtree UNTIL [TCV_Cnt = 5]			
14		+PostLinkRelEnd(TCV_ch)			3.
		localtree			
15		L!DL_DatRqIdRq	IDReq(TCV_ch, IDRequest_01('0001'B))		
16		L?DL_DatInIdRes (TCV_Mt := DL_DatInIdRes.msg.mt)	IDRes(IDResponse_01)		
17		+check2			
18		L!DL_DatRqIdRq	IDReq(TCV_ch, IDRequest_01('0001'B))		
19		L?DL_DatInIdRes (TCV_Mt := DL_DatInIdRes.msg.mt)	IDRes(IDResponse_01)		
20	L_0052	[OC_Bit7(TCV_Mt) = '0'B]		(P)	1.
21		(TCV_Cnt := TCV_Cnt+1)			
22	L_0053	[OC_Bit7(TCV_Mt) = '1'B]		(F)	
23		(TCV_Cnt := TCV_Cnt+1)			

		check1			
24	L_0054	[OC_Bit7(TCV_Mt) = '1'B]		(F)	
25	L_0055	[OC_Bit7(TCV_Mt) = '0'B]		(P)	
		check2			
26	L_0056	[OC_Bit7(TCV_Mt) = '0'B]		(F)	
27	L_0057	[OC_Bit7(TCV_Mt) = '1'B]		(P)	2.

Detailed Comments:
1. The N(SD) shall be 0.
2. The N(SD) shall be 1.
3. The test system waits the disconnection of the main signalling link.

Test Case Dynamic Behaviour

Test Case Name: TC_26_2_4_1

Group: GSM_L3_MS_v4220/InitialTest/

Purpose: To verify that the establishment cause sent by the MS in the Max-Retrans+1 CHANNEL REQUEST messages is consistent with the requested service, with the capabilities of the MS and with the indications of the network in the following case:

If the MS supports a service on a traffic channel:

when the NECI bit is set to 0 and call re-establishment is attempted and the call was established on TCH/H if the MS supports a service on half rate channel or on TCH/F otherwise.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+local_HalfRateSvcOtherwiseFullRate			
3		+BasicServiceMO(TCV_Service1, TCV_ChRate1)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
6	body	+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
7		(TCV_Null := OM_Deactivate(TCV_chTch, TCV_sacchTch), TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			2.
8		+localtree			
		localtree			
9		(TCV_Cnt :=0)			
10		REPEAT localtree1 UNTIL [TCV_Cnt=7]			
11	L_0058	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_10)	(P)	3.
12		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
		localtree1			
13		L?DL_RacInChRq	ChReq(ChRequest_10)		3.
14		(TCV_Cnt := TCV_Cnt +1)			
		local_HalfRateSvcOtherwiseFullRate			
15		(TCV_ChRate1 := C_Half)			
16		[TSPX_MOChRateA = C_Half]			
17		(TCV_Service1 := TSPX_MOBscSvcA)			
18		[TSPX_MOChRateB = C_Half]			
19		(TCV_Service1 := TSPX_MOBscSvcB)			
20		[TSPX_MOChRateC = C_Half]			

21		(TCV_Service1 := TSPX_MOBscSvcC)			
22		[TSPX_MOChRateD = C_Half]			
23		(TCV_Service1 := TSPX_MOBscSvcD)			
24		[TSPX_MOChRateE = C_Half]			
25		(TCV_Service1 := TSPX_MOBscSvcE)			
26		[TSPX_MOChRateF = C_Half]			
27		(TCV_Service1 := TSPX_MOBscSvcF)			
28		[TSPX_MOChRateG = C_Half]			
29		(TCV_Service1 := TSPX_MOBscSvcG)			
30		[TSPX_MOChRateH = C_Half]			
31		(TCV_Service1 := TSPX_MOBscSvcH)			
32		[TSPX_MOChRateI = C_Half]			
33		(TCV_Service1 := TSPX_MOBscSvcI)			
34		[TSPX_MOChRateJ = C_Half]			
35		(TCV_Service1 := TSPX_MOBscSvcJ)			
36		[C_Yes]			otherwise
37		(TCV_Service1 := TSPX_MOBscSvcA, TCV_ChRate1 := TSPX_MOChRateA)			

Detailed Comments:

1. To set up a call. The generic call setup procedure is used.
2. To stop transmission on the channel SACCH.
3. The establishment cause shall be '110'B, otherwise the test case fails in the default tree OtherEventsFail.

Test Case Dynamic Behaviour

Test Case Name: TC_26_2_4_2

Group: GSM_L3_MS_v4220/InitialTest/

Purpose: To verify that the establishment cause sent by the MS in the Max-Retrans+1 CHANNEL REQUEST messages is consistent with the requested service, with the capabilities of the MS and with the indications of the network in the following case:

If the MS supports a service on half rate channel:

when the NECI bit is set to 1 and call re-establishment is attempted and the call was established on TCH/H.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+local_PickupHalfRateSvc			
3		+BasicServiceMO(TCV_Service1, C_Half)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_1, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
5		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
6		+ltree_body			
		ltree_body			
7		+Est_MO_Call(TimingAdv(0), C_Half, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
8		(TCV_Null := OM_Deactivate(TCV_chTch, TCV_sacchTch), TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			3.
9		(TCV_Cnt :=0)			
10		REPEAT localtree UNTIL [TCV_Cnt=7]			
11	L_0059	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_11)	(P)	4.
12		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
		localtree			
13		L?DL_RacInChRq	ChReq(ChRequest_11)		4.
14		(TCV_Cnt := TCV_Cnt +1)			
		local_PickupHalfRateSvc			
15		[TSPX_MOChRateA = C_Half]			
16		(TCV_Service1 := TSPX_MOBscSvcA)			
17		[TSPX_MOChRateB = C_Half]			
18		(TCV_Service1 := TSPX_MOBscSvcB)			
19		[TSPX_MOChRateC = C_Half]			
20		(TCV_Service1 := TSPX_MOBscSvcC)			
21		[TSPX_MOChRateD = C_Half]			

22		(TCV_Service1 := TSPX_MOBscSvcD)			
23		[TSPX_MOChRateE = C_Half]			
24		(TCV_Service1 := TSPX_MOBscSvcE)			
25		[TSPX_MOChRateF = C_Half]			
26		(TCV_Service1 := TSPX_MOBscSvcF)			
27		[TSPX_MOChRateG = C_Half]			
28		(TCV_Service1 := TSPX_MOBscSvcG)			
29		[TSPX_MOChRateH = C_Half]			
30		(TCV_Service1 := TSPX_MOBscSvcH)			
31		[TSPX_MOChRateI = C_Half]			
32		(TCV_Service1 := TSPX_MOBscSvcI)			
33		[TSPX_MOChRateJ = C_Half]			
34		(TCV_Service1 := TSPX_MOBscSvcJ)			
35	L_0060	[C_Yes]		I	no half rate service

Detailed Comments:

1. The NECI = 1.
2. To set up a call on the TCH/H by generic call setup procedure.
3. To stop transmission on the SACCH.
4. The establishment cause shall be '011010'B, otherwise the test case fails in the default tree.

Test Case Dynamic Behaviour

Test Case Name: TC_26_2_4_3

Group: GSM_L3_MS_v4220/InitialTest/

Purpose: To verify that the establishment cause sent by the MS in the Max-Retrans+1 CHANNEL REQUEST messages is consistent with the requested service, with the capabilities of the MS and with the indications of the network in the following case:

If the MS supports speech:

1. when the NECI bit is set to 0 and a speech call is attempted.
2. when the NECI bit is set to 1 and a speech call is attempted.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(C_Telephony, TSPX_Telephony_Rate)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4	body	+neci0Behaviour			
5		+SetNECI(C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1)			2.
6		+neci1Behaviour			
		neci0Behaviour			
7		+InitCall(TCV_Service)			3.
8		(TCV_Cnt :=0)			
9		REPEAT localtree1 UNTIL [TCV_Cnt=7]			
10	L_0061	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	4.
11		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
		neci1Behaviour			
12		START T_dly(30000)			5.
13		?TIMEOUT T_dly			
14		+InitCall(TCV_Service)			3.
15		(TCV_Cnt := 0)			
16		REPEAT localtree2 UNTIL [TCV_Cnt=7]			
17		[NOT TSPC_DualRate]			
18	L_0062	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	6.
19		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
20		[TSPC_DualRate]			
21	L_0063	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_05)	(P)	7.
22		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		

		localtree1			
23		L?DL_RacInChRq	ChReq(ChRequest_04)		4.
24		(TCV_Cnt := TCV_Cnt +1)			
		localtree2			
25		[TSPC_DualRate]			
26		L?DL_RacInChRq	ChReq(ChRequest_05)		7.
27		(TCV_Cnt := TCV_Cnt +1)			
28		[NOT TSPC_DualRate]			
29		L?DL_RacInChRq	ChReq(ChRequest_04)		6.
30		(TCV_Cnt := TCV_Cnt +1)			

Detailed Comments:

1. The Max_retrans = 7.
2. To set NECI = 1.
3. To attempt a speech call.
4. The establishment cause shall be '111'B, otherwise the test case fails in the default tree OtherEventsFail.
5. The test system waits for 30 seconds.
6. The establishment cause shall be '111'B if the MS does not support half rate speech.
7. The establishment cause shall be '0100'B if the MS supports half rate speech.

Test Case Dynamic Behaviour

Test Case Name: TC_26_2_4_4

Group: GSM_L3_MS_v4220/InitialTest/

Purpose: To verify that the establishment cause sent by the MS in the Max-Retrans+1 CHANNEL REQUEST messages is consistent with the requested service, with the capabilities of the MS and with the indications of the network in the following case:

If the MS supports a data service:

1. when the NECI bit is set to 0 and a data call is attempted.
2. when the NECI bit is set to 1 and a data call is attempted for a service supported on half rate channel (if the MS does not support any data call on half rate channel any data service is used).

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+local_PickupDataService			
3		+BasicServiceMO(TCV_Service1, TCV_ChRate1)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
5	body	+neci0Behaviour			
6		+SetNECI(C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1)			2.
7		START T_dly(30000)			3.
8		?TIMEOUT T_dly			
9		+neci1Behaviour			
		neci0Behaviour			
10		+InitCall(TCV_Service)			4.
11		(TCV_Cnt :=0)			
12		REPEAT localtree1 UNTIL [TCV_Cnt=7]			
13	L_0064	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	5.
14		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
		neci1Behaviour			
15		+InitCall(TCV_Service)			6.
16		(TCV_Cnt := 0)			
17		+local_rcv_ChRq			
18		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
		local_rcv_ChRq			
19		[TSPC_DualRate]			
20		REPEAT localtree2 UNTIL [TCV_Cnt=7]			
21	L_0065	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_06)	(P)	7.
22		[NOT TSPC_DualRate]			
23		REPEAT localtree1 UNTIL [TCV_Cnt=7]			

24	L_0066	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	5.
		localtree1			
25		L?DL_RacInChRq	ChReq(ChRequest_04)		5.
26		(TCV_Cnt := TCV_Cnt +1)			
		localtree2			
27		L?DL_RacInChRq	ChReq(ChRequest_06)		7.
28		(TCV_Cnt := TCV_Cnt +1)			
		local_PickupDataService			
29		[(TSPX_MOBscSvcA <> C_Telephony) AND(TSPX_MOBscSvcA <> C_EmgCall)]			
30		(TCV_Service1 := TSPX_MOBscSvcA, TCV_ChRate1 := TSPX_MOChRateA)			
31		[(TSPX_MOBscSvcB <> C_Telephony) AND(TSPX_MOBscSvcB <> C_EmgCall)]			
32		(TCV_Service1 := TSPX_MOBscSvcB, TCV_ChRate1 := TSPX_MOChRateB)			
33		[(TSPX_MOBscSvcC <> C_Telephony) AND(TSPX_MOBscSvcC <> C_EmgCall)]			
34		(TCV_Service1 := TSPX_MOBscSvcC, TCV_ChRate1 := TSPX_MOChRateC)			
35		[(TSPX_MOBscSvcD <> C_Telephony) AND(TSPX_MOBscSvcD <> C_EmgCall)]			
36		(TCV_Service1 := TSPX_MOBscSvcD, TCV_ChRate1 := TSPX_MOChRateD)			
37		[(TSPX_MOBscSvcE <> C_Telephony) AND(TSPX_MOBscSvcE <> C_EmgCall)]			
38		(TCV_Service1 := TSPX_MOBscSvcE, TCV_ChRate1 := TSPX_MOChRateE)			
39		[(TSPX_MOBscSvcF <> C_Telephony) AND(TSPX_MOBscSvcF <> C_EmgCall)]			
40		(TCV_Service1 := TSPX_MOBscSvcF, TCV_ChRate1 := TSPX_MOChRateF)			
41		[(TSPX_MOBscSvcG <> C_Telephony) AND(TSPX_MOBscSvcG <> C_EmgCall)]			
42		(TCV_Service1 := TSPX_MOBscSvcG, TCV_ChRate1 := TSPX_MOChRateG)			
43		[(TSPX_MOBscSvcH <> C_Telephony) AND(TSPX_MOBscSvcH <> C_EmgCall)]			
44		(TCV_Service1 := TSPX_MOBscSvcH, TCV_ChRate1 := TSPX_MOChRateH)			
45		[(TSPX_MOBscSvcI <> C_Telephony) AND(TSPX_MOBscSvcI <> C_EmgCall)]			
46		(TCV_Service1 := TSPX_MOBscSvcI, TCV_ChRate1 := TSPX_MOChRateI)			
47		[(TSPX_MOBscSvcJ <> C_Telephony) AND(TSPX_MOBscSvcJ <> C_EmgCall)]			
48		(TCV_Service1 := TSPX_MOBscSvcJ, TCV_ChRate1 := TSPX_MOChRateJ)			
49	L_0067	[C_Yes]		I	no data service

Detailed Comments:

1. The Max_retrans =7.
2. To set NECI = 1.
3. The test system waits for 30 seconds.
4. To attempt a data call.
5. The establishment cause shall be '111'B, otherwise the test case fails in the default tree OtherEventsFail.
6. To attempt a data call if the MS supports half rate data service.
7. The establishment cause shall be '0101'B, otherwise the test case fails in the default tree OtherEventsFail.
8. To attempt any data call if the MS does not support half rate data service.

Test Case Dynamic Behaviour

Test Case Name: TC_26_2_4_5

Group: GSM_L3_MS_v4220/InitialTest/

Purpose: To verify that the establishment cause sent by the MS in the Max-Retrans+1 CHANNEL REQUEST messages is consistent with the requested service, with the capabilities of the MS and with the indications of the network in the following case:

1. when the NECI bit is set to 0 and the MS is paged with the paging indication set to "any channel".
2. when the NECI bit is set to 0 and the MS is paged with the paging indication set to "SDCCH".
3. when the NECI bit is set to 0 and the MS is paged with he paging indication set to "TCH/F".
4. when the NECI bit is set to 0 and the MS is paged with the paging indication set to "TCH/H or TCH/F".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3	body	+pagingAnyChannel			
4		+pagingSDCCH			
5		+pagingTCHF			
6		+pagingTCHHorTCHF			
		pagingAnyChannel			
7		L!DL_UdatRqPg1Rq (TCV_Cnt := 0)	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		1.
8		REPEAT localtree1 UNTIL [TCV_Cnt = 7]			
9	L_0068	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) START T_dly(C_T_Wait)	ChReq(ChRequest_12)	(P)	2.
10		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
		pagingSDCCH			
11		?TIMEOUT T_dly			
12		L!DL_UdatRqPg1Rq (TCV_Cnt := 0)	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_02)		4.
13		REPEAT localtree2 UNTIL [TCV_Cnt = 7]			
14	L_0069	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) START T_dly(C_T_Wait)	ChReq(ChRequest_03)	(P)	5.
15		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
		pagingTCHF			
16		?TIMEOUT T_dly			
17		L!DL_UdatRqPg1Rq (TCV_Cnt := 0)	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_03)		6.
18		REPEAT localtree3 UNTIL [TCV_Cnt = 7]			

19	L_0070	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) START T_dly(C_T_Wait)	ChReq(ChRequest_13)	(P)	
20		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
		pagingTCHHorTCHF			
21		?TIMEOUT T_dly			
22		L!DL_UdatRqPg1Rq (TCV_Cnt := 0)	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_04)		10.
23		REPEAT localtree4 UNTIL [TCV_Cnt = 7]			
24	L_0071	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_14)	(P)	
25		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
		localtree1			
26	L_0072	L?DL_RacInChRq	ChReq(ChRequest_12)	(P)	2.
27		(TCV_Cnt := TCV_Cnt +1)			
		localtree2			
28	L_0073	L?DL_RacInChRq	ChReq(ChRequest_03)	(P)	5.
29		(TCV_Cnt := TCV_Cnt +1)			
		localtree3			
30		[TCV_CC AND(NOT TSPC_DualRate)]			
31	L_0074	L?DL_RacInChRq	ChReq(ChRequest_12)	(P)	7.
32		(TCV_Cnt := TCV_Cnt +1)			
33		[TSPC_DualRate]			
34	L_0075	L?DL_RacInChRq	ChReq(ChRequest_07)	(P)	8.
35		(TCV_Cnt := TCV_Cnt +1)			
36		[TSPC_SDCCHOnly]			
37	L_0076	L?DL_RacInChRq	ChReq(ChRequest_03)	(P)	9.
38		(TCV_Cnt := TCV_Cnt +1)			
		localtree4			
39		[TCV_CC AND(NOT TSPC_DualRate)]			
40	L_0077	L?DL_RacInChRq	ChReq(ChRequest_12)	(P)	11.
41		(TCV_Cnt := TCV_Cnt +1)			
42		[TSPC_DualRate]			
43	L_0078	L?DL_RacInChRq	ChReq(ChRequest_08)	(P)	12.
44		(TCV_Cnt := TCV_Cnt +1)			
45		[TSPC_SDCCHOnly]			
46	L_0079	L?DL_RacInChRq	ChReq(ChRequest_03)	(P)	13.
47		(TCV_Cnt := TCV_Cnt +1)			

Detailed Comments:

1. To send a PAGING REQUEST TYPE1 message with paging indication = any channel.
2. The establishment cause shall be '100'B.
3. The test system waits for 5 seconds.
4. To send a PAGING REQUEST TYPE1 message with paging indication = SDCCH.
5. The establishment cause shall be '0001'B.
6. To send a PAGING REQUEST TYPE1 message with paging indication = TCH/F.
7. The establishment cause shall be '100'B, if the MS capability is full rate only.
8. The establishment cause shall be '0010'B, if the MS capability is dual rate.
9. The establishment cause shall be '0001'B, if the MS capability is SDCCH only.
10. To send a PAGING REQUEST TYPE1 message with paging indication = TCH/H or TCH/F

11. The establishment cause shall be '100'B, if the MS capability is full rate only.
12. The establishment cause shall be '0011'B, if the MS capability is dual rate.
13. The establishment cause shall be '0001'B, if the MS capability is SDCCH only.

Test Case Dynamic Behaviour

Test Case Name: TC_26_2_4_6

Group: GSM_L3_MS_v4220/InitialTest/

Purpose: To verify that the establishment cause sent by the MS in the Max-Retrans+1 CHANNEL REQUEST messages is consistent with the requested service, with the capabilities of the MS and with the indications of the network in the following case:

1. when the NECI bit is set to 0 and IMSI attach is attempted.
2. when the NECI bit is set to 0 and normal location updating is attempted.
3. when the NECI bit is set to 0 and periodic location updating is attempted.
4. when the NECI bit is set to 0 and IMSI detach is attempted.
5. when the NECI bit is set to 1 and IMSI attach is attempted.
6. when the NECI bit is set to 1 and normal location updating is attempted.
7. when the NECI bit is set to 1 and periodic location updating is attempted.
8. when the NECI bit is set to 1 and IMSI detach is attempted.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(1200)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_Cnt1:=0, TCV_lac := C_LAC_1)			
4		REPEAT ltree_main UNTIL [TCV_Cnt1 = 2]			
		ltree_main			
5		+SysInfoSending_nfh(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_7, TCV_Cnt1, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, TCV_lac, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_NCCP_2)			1.
6		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
7		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			
8	body	+MM_PwrOrSimOff(C_SIMIn)			
9		+SysInfoSending_nfh(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_7, TCV_Cnt1, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, TCV_lac, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_NCCP_2)			ATT =1
10		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
11		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			
12		+MM_PwrOrSimOn(C_SIMIn)			2.
13		+imsiAttachCheck			
14		+ChgLac			
15		+ltree_continue			

		ltree_continue			
16		+SysInfoSending_nfh(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_7, TCV_Cnt1, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, TCV_lac, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_NCCP_2)			3.
17		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
18		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			
19		+UpdatingCheck(33000, C_normal_updating)			7. 9.
20		+UpdatingCheck(420000, C_periodic_updating)			10. 11.
21		[TSPC_SwitchOnOff]			
22		+imsiDetachCheck			
23		(TCV_Cnt1 := TCV_Cnt1 + 1)			
24		[NOT TSPC_SwitchOnOff]			
25		(TCV_Cnt1 := TCV_Cnt1 + 1)			
		imsiAttachCheck			
26		(TCV_Cnt := 0)			
27		REPEAT localtree1 UNTIL [TCV_Cnt = 8]			
28		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
29	L_0080	L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(C_imsi_attach))	(P)	5.
30		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1, C_PLMN_1, TCV_lac))		6.
31		+PostMainLinkRel(TCV_ch)			
		UpdatingCheck(t:INTEGER; locup:B_2)			
32		(TCV_Cnt := 0)			
33		START T_dly(t)			
34	L_1041	?TIMEOUT T_dly		(F)	
35		REPEAT localtree1 UNTIL [TCV_Cnt = 8]			
36		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
37	L_0081	L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(locup))	(P)	
38		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1, C_PLMN_1, TCV_lac))		6.
39		+PostMainLinkRel(TCV_ch)			
		imsiDetachCheck			

40		+MM_PwrOrSimOff(C_SIMIn)			12.
41		(TCV_Cnt := 0)			
42		REPEAT localtree2 UNTIL [TCV_Cnt = 8]			
43		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
44		+MM_PwrOrSimOn(C_SIMIn)			2.
		localtree1			
45		[TCV_Cnt1=0]			
46	L_0082	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_09)	(P)	4.
47		CANCEL T_dly			
48		(TCV_Cnt := TCV_Cnt+1)			
49		[TCV_Cnt1=1]			
50	L_0083	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_18)	(P)	8.
51		(TCV_Cnt := TCV_Cnt+1)			
		localtree2			
52		[TCV_Cnt1=0]			
53	L_0084	L?DL_RacInChRq (TCV_Fn := DL_RacInChRq.fn, TCV_Rr := DL_RacInChRq. msg.ecau_rrf)	ChReq(ChRequest_04)	(P)	13.
54		(TCV_Cnt := TCV_Cnt+1)			
55		[TCV_Cnt1=1]			
56	L_0085	L?DL_RacInChRq (TCV_Fn := DL_RacInChRq.fn, TCV_Rr := DL_RacInChRq. msg.ecau_rrf)	ChReq(ChRequest_03)	(P)	14.
57		(TCV_Cnt := TCV_Cnt+1)			
		ChgLac			
58		[TCV_Cnt1=0]			
59		(TCV_lac := C_LAC_2)			
60		[TCV_Cnt1=1]			
61		(TCV_lac := C_LAC_1)			

Detailed Comments:

1. To set ATT = 1, i.e. MS's in the cell shall apply IMSI attach and detach procedure. TCV_Cnt1 value stands for NECI bit value.
2. To switch on or power on the MS.
3. To change LAC and set T3212 = 6 minutes.
4. The establishment cause shall be '000'B.
5. The location updating type shall be IMSI attach.
6. There is no mobile identity in this LOCATION UPDATING ACCEPT message.
7. The test system waits upto 33 seconds to receive CHANNEL REQUEST messages.
8. The establishment cause shall be '0000'B.
9. The location updating type shall be normal location updating.
10. The test system waits upto 7 minutes to receive CHANNEL REQUEST messages.
11. The location updating type shall be periodic updating.
12. To switch off the MS.
13. The establishment cause shall be '111'B.
14. The establishment cause shall be '0001'B.

Test Case Dynamic Behaviour

Test Case Name: TC_26_2_4_7

Group: GSM_L3_MS_v4220/InitialTest/

Purpose: To verify that the establishment cause sent by the MS in the Max-Retrans+1 CHANNEL REQUEST messages is consistent with the requested service, with the capabilities of the MS and with the indications of the network in the following case:

If the MS supports a non call related supplementary service operation:
when the NECI bit is set to 0 and a supplementary service operation is attempted at the MS.
when the NECI bit is set to 1 and a supplementary service operation is attempted at the MS.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_Cnt1:=0)			
4		REPEAT ltree_main UNTIL [TCV_Cnt1 = 2]			
		ltree_main			
5		+SysInfoSending_nfh(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_7, TCV_Cnt1, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_NCCP_2)			1.
6		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
7		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			
8	body	+InitNonCallSupp			2.
9		(TCV_Cnt :=0)			
10		REPEAT localtree UNTIL [TCV_Cnt=7]			
11		[TCV_Cnt1=0]			
12	L_0086	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	3.
13		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
14		(TCV_Cnt1 := TCV_Cnt1 +1)			
15		[TCV_Cnt1=1]			
16	L_0087	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_03)	(P)	4.
17		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
18		(TCV_Cnt1 := TCV_Cnt1 +1)			
		localtree			
19		[TCV_Cnt1=0]			
20		L?DL_RacInChRq	ChReq(ChRequest_04)		3.
21		(TCV_Cnt := TCV_Cnt +1)			
22		[TCV_Cnt1=1]			

23		L?DL_RacInChRq	ChReq(ChRequest_03)		4.
24		(TCV_Cnt := TCV_Cnt +1)			

Detailed Comments:

1. The Max_retrans =7. TCV_Cnt1 value stands for NECI bit value.
2. To attempt a non call related supplementary service at the MS under test.
3. The establishment cause shall be '111'B (NECI = 0).
4. The establishment cause shall be '0001'B (NECI = 1).

Test Case Dynamic Behaviour

Test Case Name: TC_26_2_4_8

Group: GSM_L3_MS_v4220/InitialTest/

Purpose: To verify that the establishment cause sent by the MS in the Max-Retrans+1 CHANNEL REQUEST messages is consistent with the requested service, with the capabilities of the MS and with the indications of the network in the following case:

If the MS supports SMS/PP MO:

when the NECI bit is set to 0 and a mobile originated short message service transaction is attempted.

when the NECI bit is set to 1 and a mobile originated short message service transaction is attempted.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_Cnt1:=0)			
4		REPEAT ltree_main UNTIL [TCV_Cnt1 = 2]			
		ltree_main			
5		+SysInfoSending_nfh(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_7, TCV_Cnt1, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_NCCP_2)			1.
6		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
7		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			
8	body	+AtmpShortMsg			2.
9		(TCV_Cnt :=0)			
10		REPEAT localtree UNTIL [TCV_Cnt=7]			
11		[TCV_Cnt1=0]			
12	L_0088	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	3.
13		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
14		(TCV_Cnt1 := TCV_Cnt1 +1)			
15		[TCV_Cnt1=1]			
16	L_0089	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_03)	(P)	4.
17		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
18		(TCV_Cnt1 := TCV_Cnt1 +1)			
		localtree			
19		[TCV_Cnt1=0]			
20		L?DL_RacInChRq	ChReq(ChRequest_04)		3.

21		(TCV_Cnt := TCV_Cnt +1)			
22		[TCV_Cnt1=1]			
23		L?DL_RacInChRq	ChReq(ChRequest_03)		4.
24		(TCV_Cnt := TCV_Cnt +1)			

Detailed Comments:
1. The Max_retrans =7. TCV_Cnt1 value stands for NECI bit value.
2. To attempt a mobile originated short message service transaction at the MS under test.
3. The establishment cause shall be '111'B (NECI = 0).
4. The establishment cause shall be '0001'B (NECI = 1).

Test Group IdleMode

Test Case Dynamic Behaviour

Test Case Name: TC_26_3_2
Group: GSM_L3_MS_v4220/IdleMode/
Purpose: To verify that a MS can present the available PLMNs to the user when asked to do so in manual mode according to the requirements of GSM 05.08 and 02.11.
Default: OtherEventsFail
Comments: For the test the SIM shall contain a PLMN-Selector that contains only the HPLMN and a empty forbidden PLMN list. Final verdict is assigned in the test step PLMNsCHK

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(65, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_10, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_1, C_cch_1Comb, C_BPM_2, C_T3212_0, C_ci_cellA, C_MCC_2, C_PLMN_4, C_LAC_1, CellOpt_02, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys4, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, BcchFreqLst_05, BcchFreqLst_05, BcchFreqLst_13, BcchFreqLst_13, C_noRestablishment, C_BCC, C_NCC_0, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcn_1, C_arfcnAd_1, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+MM_PwrOrSimOff(C_SIMIn)			
4		+StartMultiCells_01(C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_ATT_0, C_BABR_1, C_cch_1Comb, C_BPM_2, C_T3212_0, C_MCC_3, C_PLMN_5, C_MCC_4, C_PLMN_6, C_MCC_5, C_PLMN_7, C_MCC_6, C_PLMN_8, C_MCC_7, C_PLMN_9, C_MCC_8, C_PLMN_10, C_MCC_1, C_PLMN_1, C_LAC_1)			
5	body	+MM_PwrOrSimOn(C_SIMIn)			
6		(TCV_Null := OO_PLMNselModeMan())			
7		+PLMNsCHK			1.
8		(TCV_Null := OO_PLMNselModeAuto())			

Detailed Comments: 1. Final verdict is assigned in the test step.

Test Case Dynamic Behaviour

Test Case Name: TC_26_3_3
Group: GSM_L3_MS_v4220/IdleMode/
Purpose: To verify that the MS will not produce any RF transmission if no BSS is received.
Default: OtherEventsFail
Comments: For the test the SIM shall contain a PLMN-Selector that contains only the HPLMN and a empty forbidden PLMN list. Final verdict is assigned in the test steps ServiceIndCHK and RFtransCHK

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcB, TSPX_MOChRateB)			
3		+IdleUpdated(65, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_10, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_1, C_cch_1Comb, C_BPM_2, C_T3212_0, C_ci_cellA, C_MCC_2, C_PLMN_4, C_LAC_1, CellOpt_02, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys4, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, BcchFreqLst_05, BcchFreqLst_05, BcchFreqLst_13, BcchFreqLst_13, C_noRestablishment, C_BCC, C_NCC_0, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcn_1, C_arfcnAd_1, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+StartMultiCells_01(C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), 0, '001'B, '001'B, '010'B, '00'O, C_MCC_3, C_PLMN_5, C_MCC_4, C_PLMN_6, C_MCC_5, C_PLMN_7, C_MCC_6, C_PLMN_8, C_MCC_7, C_PLMN_9, C_MCC_8, C_PLMN_10, C_MCC_1, C_PLMN_1, C_LAC_1)			
5	body	+StopAllBCCH			
6		START T_dly(20000)			
7		?TIMEOUT T_dly			
8		+InitCall(TCV_Service)			
9		+RFtransCHK			1.
10		[TSPC_Serv_TS12]			
11		+InitCall(C_EmgCall)			
12		+RFtransCHK			1.
13		[NOT TSPC_Serv_TS12]			

Detailed Comments: 1. The verdict is assigned in these test steps.

Test Case Dynamic Behaviour

Test Case Name: TC_26_3_4

Group: GSM_L3_MS_v4220/IdleMode/

Purpose: To verify that in manual mode the MS is able to obtain normal service on a PLMN which is neither the better nor a preferred PLMN and that it tries to obtain service on VPLMN if and only if the user selects it manually.

Default: OtherEventsFail

Comments: For the test the preferred PLMN list of the SIM does not contain PLMN2('02'O) but contains PLMN3('03'O). The MS shall be set to manual mode before the test starts.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(38, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC_2, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+StartCellB_1(33, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_2, C_LAC_2, C_noRestablishment, C_BCC, C_NCC)			
4		(TCV_Null := OO_PLMNselModeMan())			
5		+ltree_body			
		ltree_body			
6		(TCV_Null := OM_StopCell(C_CellA))			
7		+NoReaction(120000)			1.
8		+StartCellA_1(48, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_3, C_LAC_1, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment)			
9		+NoReaction(120000)			1.
10		(TCV_Null := OO_SelPLMN(C_PLMN_2))			
11		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
12		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
13		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
14	L_0090	L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_05)	(P)	2.
15		ACTIVATE(OtherEventsFail)			Restore Normal default
16		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1, C_PLMN_1, C_LAC_1))		
17		+PostMainLinkRel(TCV_ch)			
18		(TCV_Null := OM_StopCell(C_CellB))			
19		+NoReaction(120000)			1.

20			(TCV_Null := OO_PLMNselModeAuto())			

Detailed Comments:
1. During 2 minutes the MS does not send any CHANNEL REQUEST, pass. If the MS does the test case fail in the default tree.
2. The expected LOCATION UPDATING REQUEST message received on Cell B.

Test Group BiBo

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_1
Group: GSM_L3_MS_v4220/BiBo/
Purpose: To verify that a MS supporting TCH and the call control protocol ignores a message containing an undefined protocol discriminator in the special case of a message coded otherwise like a CC STATUS ENQUIRY message received by the MS having a mobile terminating call in CC-state U10, "active".
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3	body	[TCV_CC]			1.
4		+BasicServiceMT(TSPX_MTBscSvcA, TSPX_MTChRateA)			
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
7		L!DL_DatRqUnknown	Unknown(TCV_chTch, UnknownMsg_01(TI_02))		2.
8		START T_dly(10000)			
9	L_0091	?TIMEOUT T_dly		(P)	
10		+PostMainLinkRel(TCV_chTch)			
11		[NOT TCV_CC]			3.
12		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
13		L!DL_DatRqUnknown	Unknown(TCV_ch, UnknownMsg_02(TI_04))		4.
14		START T_dly(10000)			
15	L_0092	?TIMEOUT T_dly		(P)	
16		+PostMainLinkRel(TCV_ch)			

Detailed Comments:
1. If the MS supports any bearer capability the test case goes through this branch.
2. To send a CC STATUS ENQUIRY alike unknown message on channel FACCH.
3. If the MS does not support any bearer capability the test case goes through this branch.
4. To send a CC STATUS ENQUIRY alike unknown message on channel SDCCH4.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_2_1_1
Group: GSM_L3_MS_v4220/BiBo/
Purpose: To verify that the MS ignores an RR message with skip indicator different from H'0 in the special case of a PAGING REQUEST TYPE 1 message received in the MM-state "idle, updated" and in RR-idle mode.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_Cnt :=1)			
4	body	REPEAT localtree UNTIL [TCV_Cnt = 7]			
5		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_inv_01(8))		1.
6		START T_dly(3000)			
7	L_0093	?TIMEOUT T_dly		(P)	
		localtree			
8		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_inv_01(TCV_Cnt))		1.
9		START T_dly(3000)			
10		?TIMEOUT T_dly			
11	L_0094	(TCV_Cnt := TCV_Cnt+1)		(P)	

Detailed Comments: 1. To send PAGING REQUEST TYPE 1 message with unknown skip indictor.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_2_1_2

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that the MS ignores RR messages with skip indicator different from H'0 in the case of a message being received during the RR-connection establishment in the MM-state "idle, updated" / "wait for network command" and in RR-connected mode.
RR-connected mode.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
4		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
5		L!DL_UdatRqImmass START T_dly1(1000)	ImmAss(TCV_agch, ImmAsgn_inv_01(TCV_Rr, TCV_Fn, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		2.
6		+chRq_check			
7		L!DL_UdatRqImmassRej START T_dly1(1000)	ImmAssRej(TCV_agch, ImmAsgnRej_inv_01(TCV_Rr, TCV_Fn))		4.
8		+chRq_check			
9		+continue			
		continue			
10		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		valid message
11		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
12		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNDef, TSPX_RANDDef))		
13		L?DL_DatInAuthRes	AuthRes(AuthResponse)		
14		L!DL_DatRqCphmCmd	CphCmd(TCV_ch, CphModeCmd_inv_02)		5.
15		L!DL_DatRqIdRq	IDReq(TCV_ch, IDRequest_01('0001'B))		
16		L?DL_DatInIdRes	IDRes(IDResponse_01)		
17		L!DL_DatRqAssCmd	AssCmd(TCV_ch, AsgnCmd_inv_01(TCV_slot, TCV_tsc))		6.

18		+check2			
19		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_inv_01(TCV_slot, TCV_tsc, C_NCC_1, C_BCC_3, C_arfcnA))		7.
20		+check3			
21		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_inv_02)		9.
22		L!DL_DatRqIdRq	IDReq(TCV_ch, IDRequest_01('0001'B))		
23	L_0095	L?DL_DatInIdRes	IDRes(IDResponse_01)	(P)	
24		+PostLinkRelEnd(TCV_ch)			
		check2			
25		START T_dly(3000)			
26	L_0096	L?DL_EstIn	DLEstInd(TCV_ch)	(F)	
27	L_0097	?TIMEOUT T_dly		(P)	
		check3			
28		START T_dly(3000)			8.
29		L?DL_EstIn	DLEstInd(TCV_ch)		
30	L_0098	L?DL_DatInHofl	HndOvFlRcv(TCV_ch, HandOvFail_01)	(F)	
31	L_0099	L?DL_DatInRrst	RrStatusRcv(TCV_ch, RRStatus_01)	(F)	
32	L_0100	?TIMEOUT T_dly		(P)	
		chRq_check			
33	L_0101	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly1	ChReq(ChRequest_17)	(P)	3.
34	L_0102	?TIMEOUT T_dly1		(F)	

Detailed Comments:

1. To set the Max_Retrans = 2.
2. To send an invalid IMMEDIATE ASSIGNMENT message with skip indicator = 1.
3. Retransmission of Channel request within 1s indicates that mobile ignored the message with invalid skip indicator.
4. To send an invalid IMMEDIATE ASSIGNMENT REJECT message with skip indicator = 2.
5. To send an invalid CIPHERING MODE COMMAND message with skip indicator = 3.
6. To send an invalid ASSIGNMENT COMMAND message with skip indicator = 4.
7. To send an invalid HANDOVER COMMAND message with skip indicator = 5.
8. To check that there is no HANDOVER FAILURE or RR-STATUS message on old channel.
9. To send an invalid CHANNEL RELEASE message with skip indicator = 6.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_2_2
Group: GSM_L3_MS_v4220/BiBo/
Purpose: To verify that the MS ignores an MM message with skip indicator different from H'0 in the special case of an MS supporting the call control protocol and an IDENTITY REQUEST message received in the active state of a mobile terminating call.
Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3	body	[TCV_CC]			
4		+BasicServiceMT(TSPX_MTBscSvcB, TSPX_MTChRateB)			
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			1.
7		+maintest1			
8		[NOT TCV_CC]			2.
9		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
10		+maintest2			
		maintest1			
11		(TCV_Cnt := 0)			
12		REPEAT subtree1 UNTIL [TCV_Cnt = 6]			
13		L!DL_DatRqIdRq	IDReq(TCV_chTch, IDRequest_inv_01(8))		3.
14		START T_dly(5000)			
15	L_0103	?TIMEOUT T_dly		(P)	
16		+PostMainLinkRel(TCV_chTch)			
		maintest2			
17		(TCV_Cnt := 0)			
18		REPEAT subtree2 UNTIL [TCV_Cnt = 6]			
19		L!DL_DatRqIdRq	IDReq(TCV_ch, IDRequest_inv_01(8))		4.
20		START T_dly(5000)			
21	L_0104	?TIMEOUT T_dly		(P)	
22		+PostMainLinkRel(TCV_ch)			
		subtree1			
23		L!DL_DatRqIdRq	IDReq(TCV_chTch, IDRequest_inv_01(TCV_Cnt+1))		3.

24		START T_dly(1000)			
25	L_0105	?TIMEOUT T_dly		(P)	
26		(TCV_Cnt := TCV_Cnt + 1)			
		subtree2			
27		L!DL_DatRqIdRq	IDReq(TCV_ch, IDRequest_inv_01(TCV_Cnt+1))		4.
28		START T_dly(1000)			
29	L_0106	?TIMEOUT T_dly		(P)	
30		(TCV_Cnt := TCV_Cnt + 1)			

Detailed Comments:

1. This subtree is for the MS supporting at least one bearer capability.
2. This subtree is for the MS not supporting any bearer capability.
3. To send an invalid IDENTITY REQUEST message containing incorrect skip indicator on channel
   FACCH.
4. To send an invalid IDENTITY REQUEST message containing incorrect skip indicator on channel
   SDCCH4.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_2_3

Group: GSM_L3_MS_v4220/BiBo/

Purpose:

a) To verify that the MS having a mobile terminating call in CC-state U10, "active", on receipt of a DISCONNECT message which includes a transaction identifier which is not recognised as relating to an active call or a call in progress, sends a RELEASE COMPLETE message with cause value #81 and referring to the latter TI without changing the state of the active call (this is verified by use of the status enquiry procedure).

b) To verify that the MS having a mobile terminating call in CC-state U10, "active", on receipt of a

b1) RELEASE COMPLETE message which includes a transaction identifier with a value different from 111, which is not recognised as relating to an active call or a call in progress, or a

b2) SETUP message with TI flag referring to a transaction originated by the MS (in the special case where the TI value is equal to the TI value relating to the active call), or a

b3) SETUP message with TI referring to the active call, ignores that message without changing the state of the active call (this is verified by use of the status enquiry procedure).

c) To verify that the MS ignores a CC mesage with a TI value 111.

The test is only applicable to an MS supporting the call control protocol for at least one BC.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcC, TSPX_MTChRateC)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
6	body	+test_a			
7		+test_b1			
8		+test_b2			
9		+test_b3			
10		+test_c			
		test_a			
11		L!DL_DatRqDisc	Disc_inv(TCV_chTch, Disconn_inv_01(TI_03))		1.
12		L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_52(TI_05))		
13		+CCstatuschk_01(TCV_chTch, C_U10)			

		test_b1			
14		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_02(TI_04))		2.
15		START T_dly(5000)			
16	L_0107	?TIMEOUT T_dly		(P)	
17		+CCstatuschk_01(TCV_chTch, C_U10)			
		test_b2			
18		L!DL_DatRqSetup	SetupSnd(TCV_chTch, Setup_inv_01(TI_01))		3.
19		START T_dly(5000)			
20	L_0108	?TIMEOUT T_dly		(P)	
21		+CCstatuschk_01(TCV_chTch, C_U10)			
		test_b3			
22		L!DL_DatRqSetup	SetupSnd(TCV_chTch, Setup_03(TI_02))		4.
23		START T_dly(5000)			
24	L_0109	?TIMEOUT T_dly		(P)	
25		+CCstatuschk_01(TCV_chTch, C_U10)			
		test_c			
26		L!DL_DatRqDisc	Disc_inv(TCV_chTch, Disconn_inv_01(TI_06))		5.
27		START T_dly(5000)			
28	L_0110	?TIMEOUT T_dly		(P)	
29		+CCstatuschk_01(TCV_chTch, C_U10)			
30		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. The TI value of the DISCONNECT message does not refer to the active call.
2. The TI value of the RELEASE COMPLETE message does not refer to the active call ("0000'B).
3. The TI flag of the SETUP message is set to 1 and TI value refers to the active call.
4. This SETUP contains TI refers to the active call ('0000'B).
5. The TI value of the DISCONNECT message is '111'B.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_3_1

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that a MS supporting the call control protocol for at least one BC, having a mobile terminating call in CC-state U10, "active", on receipt of a message with CC protocol discriminator and an arbitrary undefined message, returns a STATUS message with cause value #97 to the peer CC entity without changing the state of the active call (this is verified by use of the status enquiry procedure.)

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcD, TSPX_MTChRateD)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Reestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
6	body	L!DL_DatRqUndefCC	UndefCC(TCV_chTch, UndefCC_02(TI_02))		1.
7	L_0111	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_02(TI_01))	(P)	
8		+CCstatuschk_01(TCV_chTch, C_U10)			
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments: 1. This is an undefined CC message.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_3_2

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that a MS supporting the call control protocol for at least one BC, having a mobile terminating call in CC-state U10, "active", on receipt of a message with MM protocol discriminator and message type undefined for the mobility management protocol, returns an MM-STATUS message with reject cause value #97 without changing the state of the active call (this is verified by use of the status enquiry procedure.) This is tested in the special case where the CC TI has value 0 (so that it has the same encoding as the skip indicator when sent from the SS) and where the message type has the same encoding as DISCONNECT in CC.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcE, TSPX_MTChRateE)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
6	body	L!DL_DatRqUndefMM	UndefMM(TCV_chTch, UndefMM_01)		1.
7	L_0112	L?DL_DatInMmst	MMSt(MMstatus_01)	(P)	
8		+CCstatuschk_01(TCV_chTch, C_U10)			
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments: 1. This is an undefined MM message.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_3_3
Group: GSM_L3_MS_v4220/BiBo/
Purpose: To verify that an MS in RR connected mode on receipt of a message with RR protocol discriminator and message type undefined for the RR protocol, returns an RR-STATUS message with reject cause value #97 without changing its state (this is checked by observing that the MS does not send L3 messages.)
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		[TSPC_MTsvc]			
3		+BasicServiceMT(TSPX_MTBscSvcD, TSPX_MTChRateD)			
4		+maintree			
5		[NOT TSPC_MTsvc]			
6		+maintree			
		maintree			
7		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
8		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
9	body	L!DL_DatRqUndefRR	UndefRR(TCV_ch, UndefRR_01)		1.
10		L?DL_DatInRrst	RrStatusRcv(TCV_ch, RRStatus_03)		
11		START T_dly(5000)			Start L2 fill frame check
12	L_0113	L?OTHERWISE		F	L3 msg received
13	L_0114	?TIMEOUT T_dly		(P)	
14		[TSPC_MTsvc]			2.
15		+Authentication(TCV_ch, TSPX_CKSNDef, TSPX_RANDDef)			
16		+Ciphering_on(TCV_ch)			
17		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		
18	L_0115	L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))	(P)	
19		+PostMainLinkRel(TCV_ch)			
20		[NOT TSPC_MTsvc]			3.
21		L!DL_DatRqSetup	SetupSnd(TCV_ch, Setup_01(TI_02))		
22	L_0116	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TI_01))	(P)	
23		+PostMainLinkRel(TCV_ch)			

Detailed Comments:
1. To send an undefined RR message.
2. If the MS supports at least one bearer capability, the test case goes through this subtree.
3. If the MS does not support any bearer capability, the test case goes through this subtree.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_3_4

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that a MS supporting the call control protocol for at least one BC, having a call in CC-state U10, "active", on receipt of an inopportune CC message, returns a STATUS message with reject cause value #98 without changing the state of the active call (this is verified by use of the status enquiry procedure.) This is tested in the special case where the inopportune CC message is a CALL PROCEEDING message relating to the active call.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcF, TSPX_MTChRateF)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			1.
6	body	L!DL_DatRqCallProc	CallProc(TCV_chTch, CallProced_01(TI_02))		2.
7	L_0117	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_03(TI_01))	(P)	
8		+CCstatuschk_01(TCV_chTch, C_U10)			
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To enter CC state U10.
2. To send an inopportune CALL PROCEEDING message.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_4_1
Group: GSM_L3_MS_v4220/BiBo/
Purpose: To verify that the MS ignores an unforeseen second occurrence of an information element with format T, TV, or TLV in the special case of the mobile identity IE which has format TLV in the LOCATION UPDATING ACCEPT message.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+StartCellB_1(C_E_neighbourdefault, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, C_noRestablishment, C_BCC, C_NCC)			
4	body	+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			1.
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq_01(TCV_RaCh, ChRequest_09)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		2.
8		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_04(C_normal_updating, MiTmsi_01))		
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		L!DL_DatRqLupAcpErr	LocAcp_inv(TCV_ch, MiImsi_01iei, MiTmsi_01iei, C_MCC_1, C_PLMN_1, C_LAC_2)		3.
11		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
12		L?DL_RelIn	DLRelInd_01		
13		START T_dly(5000)			
14		?TIMEOUT T_dly			
15		+localtree			
		localtree			
16		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		4.
17		START T_dly(C_T_Wait)			
18		?TIMEOUT T_dly			
19		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg,		5.

			PgReqTp1_05)		
20	L_0118	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq_01(C_RACH_B_1, ChRequest_01)	(P)	
21		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		

Detailed Comments:
1. MS reselects cell B.
2. The cell B assigns C_SDCCH4_B_1 to the MS.
3. To send an invalid LOCATION UPDATING ACCEPT message containing duplicated IE's.
4. To send a PAGING REQUEST TYPE 1 message containing mobile identity = TMSI of the MS.
5. To send a PAGING REQUEST TYPE 1 message containing mobile identity = IMSI of the MS.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_5_1_1_1
Group: GSM_L3_MS_v4220/BiBo/
Purpose: To verify that the MS in RR connected mode releases the connection upon receipt of a CHANNEL RELEASE message with missing RR cause (which is "mandatory" in that message).
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
4	body	L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_inv_01)		1.
5	L_0119	L?DL_RelIn	DLRelInd_01	P	

Detailed Comments:
1. To send an invalid CHANNEL RELEASE message missing mandatory RR cause IE.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_5_1_1_2
Group: GSM_L3_MS_v4220/BiBo/
Purpose: To verify that the MS in RR connected mode ignores a ciphering mode command message in which the ciphering mode setting IE and cipher response IE are missing except for the fact that it returns a RR-STATUS message.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
4	body	L!DL_DatRqCphmCmd	CphCmd(TCV_ch, CphModeCmd_inv_01)		1.
5	L_0120	L?DL_DatInRrst	RrStatusRcv(TCV_ch, RRStatus_02)	(P)	
6		+PostMainLinkRel(TCV_ch)			

Detailed Comments: 1. To send an invalid CIPHERING MODE COMMAND missing mandatory ciphering mode setting IE and cipher response IE.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_5_1_2
Group: GSM_L3_MS_v4220/BiBo/
Purpose: To verify that the MS having an RR-connection established ignores a HANDOVER COMMAND message containing in the non-imperative part an IE encoded as comprehension required except for the fact that it returns a RR-STATUS message with cause # 96 "invalid mandatory information".
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3	body	[TCV_CC]			1.
4		+BasicServiceMT(TSPX_MTBscSvcG, TSPX_MTChRateG)			
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
7		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_inv_02(TSPX_TmSltDef, TSPX_TscDef, C_NCC_1, C_BCC_3, C_arfcnA))		2.
8	L_0121	L?DL_DatInRrst	RrStatusRcv(TCV_chTch, RRStatus_02)	(P)	
9		+PostMainLinkRel(TCV_chTch)			
10		[NOT TCV_CC]			3.
11		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
12		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_inv_02(TSPX_TmSltDef, TSPX_TscDef, C_NCC_1, C_BCC_3, C_arfcnA))		4.
13	L_0122	L?DL_DatInRrst	RrStatusRcv(TCV_ch, RRStatus_02)	(P)	
14		+PostMainLinkRel(TCV_ch)			

Detailed Comments:
1. If the MS supports at least one bearer capability, the test case goes through this subtree.
2. To send an invalid HANDOVER COMMAND message containing comprehension requires IE on
the channel TCV_chTch.
3. The test case goes through this subtree if the MS does not support any bearer capability.
4. To send an invalid HANDOVER COMMAND message containing comprehension requires IE on
the channel TCV_ch.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_5_2_1

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that an MS supporting at least one BC, having a CC entity in state U10, "active", ignores an MM message with syntactically incorrect IE except for the fact that it sends an MM-STATUS message with reject cause #96. This is tested in the special case of an IDENTITY REQUEST message in which the (mandatory) identity type IE specifies a reserved value for the type of identity; that the MS otherwise ignores the message is checked by means of the status enquiry procedure.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcH, TSPX_MTChRateH)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
6	body	L!DL_DatRqIdRq	IDReq(TCV_chTch, IDRequest_inv_02)		1.
7	L_0123	L?DL_DatInMmst	MMSt(MMstatus_02)	(P)	
8		+CCstatuschk_02(TCV_chTch, C_U10, TI_02, TI_01)			2.
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To send an invalid IDENTITY REQUEST message containing the identity type IE = reserved value.
2. To check whether the MS is still in the state U10. If not the test case fails in the test step.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_5_2_2

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that an MS having been paged and having an RR connection established ignores an MM message with syntactically incorrect IE except for the fact that it sends an MM-STATUS message with reject cause #96. This is tested in the special case of an IDENTITY REQUEST message in which the (mandatory) identity type IE specifies a reserved value for the type of identity; the fact that the MS otherwise ignores the message is checked by testing that it answers as usual to an incoming SETUP message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		[TSPC_MTsvc]			
3		+BasicServiceMT(TSPX_MTBscSvcF, TSPX_MTChRateF)			
4		+maintree			
5		[NOT TSPC_MTsvc]			
6		+maintree			
		maintree			
7		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
8		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
9	body	L!DL_DatRqIdRq	IDReq(TCV_ch, IDRequest_inv_02)		1.
10	L_0124	L?DL_DatInMmst	MMSt(MMstatus_02)	(P)	
11		[TSPC_MTsvc]			2.
12		+Authentication(TCV_ch, TSPX_CKSNDef, TSPX_RANDDef)			
13		+Ciphering_on(TCV_ch)			
14		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		
15	L_0125	L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))	(P)	
16		+PostLinkRelEnd(TCV_ch)			
17		[NOT TSPC_MTsvc]			3.
18		L!DL_DatRqSetup	SetupSnd(TCV_ch, Setup_01(TI_02))		
19	L_0126	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TI_01))	(P)	
20		+PostLinkRelEnd(TCV_ch)			

Detailed Comments:

1. To send an invalid IDENTITY REQUEST message in which the identity type IE contains reserved value on the channel C_SDCCH4_A_1.
2. This subtree is for the MS which supports at least one bearer capability.
3. This subtree is for the MS which does not support any bearer capability.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_5_2_3
Group: GSM_L3_MS_v4220/BiBo/
Purpose: To verify that the MS on receipt of an MM message containing an IE unknown in the message, but encoded as "comprehension required" ignores the message except for the fact that it returns an MM-STATUS message with cause value #96 "invalid mandatory information"; this in the special case of the MM message being a LOCATION UPDATING ACCEPT responding to a LOCATION UPDATING REQUEST from the MS.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+StartCellB_1(C_E_neighbourdefault, C_Immass,TCV_slot, TCV_tsc, TimingAdv(0), C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, C_noRestablishment, C_BCC, C_NCC)			
4	body	+localbody			
		localbody			
5		+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			1.
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq_01(TCV_RaCh, ChRequest_09)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
9		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_04(C_normal_updating, MiTmsi_01))		
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(Mi_01, C_MCC_1, C_PLMN_1, C_LAC_2))		2.
12		L?DL_DatInMmst	MMSt(MMstatus_02)		
13		L?DL_RelIn	DLRelInd_01		
14		+ltree_continue			
		ltree_continue			
15		L?DL_RacInChRq(TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
16		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
17		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(		

			TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
18		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_04(C_normal_updating, MiImsi_01))		
19		ACTIVATE(OtherEventsFail)			Restore Normal default
20		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiTmsi_01iei, C_MCC_1, C_PLMN_1, C_LAC_2))		3.
21		L?DL_DatInTmsireCom	TmsiReallocCmp(TCV_ch)		
22	L_0127	L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)	(P)	
23		L?DL_RelIn	DLRelInd_01		

Detailed Comments:

1. To lower the RF level of cell A until the MS selects cell B.
2. To send a LOCATION UPDATING ACCEPT message containing comprehension required IE.
3. To send a LOCATION UPDATING ACCEPT message containing location area identification = cell B and mobile identity = TMSI.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_5_3_1_1
Group: GSM_L3_MS_v4220/BiBo/
Purpose: To verify that the MS having an MT call in state U10, "active", on receipt of a DISCONNECT message in which the mandatory cause IE is missing shall return a RELEASE message with cause value #96 "invalid mandatory information".
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
6	body	L!DL_DatRqDisc	Disc_inv(TCV_chTch, Disconn_inv_02(TI_02))		1.
7	L_0128	L?DL_DatInRel	ReleaseRcv(Release_01(TI_01))	(P)	
8		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_02(TI_02))		
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments: 1. To send an invalid DISCONNECT message in which the mandatory cause IE is missing.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_5_3_1_2

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that the MS having an MT call in state U10, "active", on receipt of a STATUS message in which the mandatory cause IE and call state IE are missing shall ignore the message except for the fact that it return a STATUS message with cause value #96 "invalid mandatory information" (that the MS doesn't change state is checked by use of the status enquiry procedure).

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcJ, TSPX_MTChRateJ)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
6	body	L!DL_DatRqCcst	CCStSnd(TCV_chTch, CCStatus_inv_01(TI_02))		1.
7	L_0129	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_04(TI_01))	(P)	
8		+CCstatuschk_01(TCV_chTch, C_U10)			
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To send an invalid STATUS message in which the mandatory cause IE and call state IE are missing.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_5_3_2

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that an MS supporting the call control protocol for at least one BC having a call control entity in state U3 ignores a CONNECT message containing in the non-imperative part an IE encoded as comprehension required except for the fact that it returns a STATUS message with cause value #96 "invalid mandatory information".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcG, TSPX_MOChRateG)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+PreEnterCCstateU3(TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
5	body	L!DL_DatRqConnErr	Conn_invSnd(TCV_ch, Connect_inv_01(TCV_TI))		1.
6	L_0130	L?DL_DatInCcst	CCStRcv(TCV_ch, CCStatus_04(TCV_TI0))	(P)	2.
7		+CCstatuschk_03(C_U3, TCV_TI)			3.
8		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
9		L?DL_RelIn	DLRelInd_01		

Detailed Comments:

1. To send an invalid CONNECT message containing comprehension required IE.
2. The expected STATUS message received.
3. To check whether the MS is still in the state U3.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_6_1_1
Group: GSM_L3_MS_v4220/BiBo/
Purpose: To verify that the MS on receipt of an MM message containing an IE unknown in the message and unknown in the MM protocol which is not encoded as "comprehension required" ignores that IE; this in the special case of the MM message being a LOCATION UPDATING ACCEPT responding to a LOCATION UPDATING REQUEST from the MS.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_neighbourdefault, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_2, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_01, BcchFreqLst_48, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
3		+StartCellA(C_E_default, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_02, CellChDes_03, BcchFreqLst_45, BcchFreqLst_03, C_noRestablishment)			
4		+ltree_body			
		ltree_body			
5		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
9		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_04(C_normal_updating, MiTmsi_01))		2.
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		L!DL_DatRqLupAcpErr	LocAcp_inv(TCV_ch, Mi_02, MiTmsi_02iei, C_MCC_1, C_PLMN_1, C_LAC_1)		3.
12	L_0131	L?DL_DatInTmsireCom	TmsiReallocCmp(TCV_ch)	(P)	
13		+local_postamble			
14		+PostMainLinkRel(TCV_ch)			
		local_postamble			
15		+TmsiReallocation(MiTmsi_01, C_MCC_1, C_PLMN_1, C_LAC_1)			4.

Detailed Comments:
1. The MS listen to cell B.
2. The mobile identity is TMSI of the MS.

	3. To send an invalid LOCATION UPDATING ACCEPT message containing unknown IEI and new TMSI. 4. Local Postamble: to ensure that the test case terminates with the default TMSI.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_6_1_2

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that the MS on receipt of an MM message containing an IE unknown in the message, but known in the MM protocol, which is not encoded as "comprehension required" ignores that IE; this in the special case of the MM message being a LOCATION UPDATING ACCEPT responding to a LOCATION UPDATING REQUEST from the MS.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_neighbourdefault, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_2, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_01, BcchFreqLst_48, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
3		+StartCellA(C_E_default, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_02, CellChDes_03, BcchFreqLst_45, BcchFreqLst_03, C_noRestablishment)			
4		+ltree_body			
		ltree_body			
5		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
9		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_04(C_normal_updating, MiTmsi_01))		2.
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		L!DL_DatRqLupAcpErr	LocAcp_inv(TCV_ch, Mi_06, MiTmsi_02iei, C_MCC_1, C_PLMN_1, C_LAC_1)		3.
12	L_0132	L?DL_DatInTmsireCom	TmsiReallocCmp(TCV_ch)	(P)	
13		+local_postamble			
14		+PostMainLinkRel(TCV_ch)			
		local_postamble			
15		+TmsiReallocation(MiTmsi_01, C_MCC_1, C_PLMN_1, C_LAC_1)			4.

Detailed Comments:

1. The MS listen to cell B.
2. The mobile identity is TMSI of the MS.
3. To send a LOCATION UPDATING ACCEPT message containing unknown IE and new

TMSI.
4. Local Postamble: to ensure that the test case terminates with the default TMSI.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_6_2_1
Group: GSM_L3_MS_v4220/BiBo/
Purpose: To verify that an MS supporting the CC protocol for at least one BC receiving a CC message containing an IE unknown in the message which is not encoded as "comprehension required" ignores that IE; this in the special case of the CC message being a CALL PROCEEDING message received by the MS in state U1.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcF, TSPX_MOChRateF)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+PreEnterCCstateU1(TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
5	body	L!DL_DatRqCallProc	CallProc(TCV_ch, CallProced_inv_02(TCV_TI))		1.
6		+CCstatuschk_03(C_U3, TCV_TI)			2.
7		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
8		L?DL_RelIn	DLRelInd_01		

Detailed Comments:
1. To send an invalid CALL PROCEEDING message containing optional unknown IE.
2. To check whether the MS enters the state U3, the verdict is assigned in the test step.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_6_2_2

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that an MS supporting the CC protocol for at least one BC receiving a CC message containing an IE unknown in the message which is not encoded as "comprehension required" ignores that IE; this in the special case of a DISCONNECT message received by the MS in state U10.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcJ, TSPX_MTChRateJ)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			1.
6	body	L!DL_DatRqDiscErr	Disc_inv_err(TCV_chTch, Disconn_inv_03(TI_02))		2.
7	L_0133	L?DL_DatInRel	ReleaseRcv(Release_10(TI_01))	(P)	
8		+CCstatuschk_01(TCV_chTch, C_U19)			
9		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
10		L?DL_RelIn	DLRelInd_01		

Detailed Comments:
1. To enter CC state U10.
2. To send an invalid DISCONNECT message containing optional unknown IE.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_6_2_3

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that an MS supporting the CC protocol for at least one BC receiving a CC message containing an IE unknown in the message which is not encoded as "comprehension required" ignores that IE; this in the special case of a RELEASE message received by the MS having sent in state U10 a DISCONNECT message

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcA, TSPX_MTChRateA)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			1.
6	body	(TCV_Null := OO_TermCall())			
7		L?DL_DatInDisc	DiscRcv(TCV_chTch, DisconnR(TI_01, Cause_Def))		
8		L!DL_DatRqRel	ReleaseSnd(TCV_chTch, ReleaseReq_inv_01(TI_02))		2.
9	L_0134	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TI_01))	(P)	
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To enter CC state U10.
2. To send an invalid RELEASE REQUEST message containing unknown optional IE.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_6_2_4

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that an MS supporting the CC protocol for at least one BC receiving a CC message containing an IE unknown in the message which is not encoded as "comprehension required" ignores that IE; this in the special case of a RELEASE COMPLETE message received by the MS in state U19.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcB, TSPX_MTChRateB)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
6	body	L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TI_02, Cause_01, ProgInd_omit, UuInfo_omit))		
7		L?DL_DatInRel	ReleaseRcv(Release_02)		
8		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_inv_02(TI_02))		1.
9	L_0135	L?DL_RelIn	DLRelInd_01	(P)	

Detailed Comments: 1. To send an invalid RELEASE COMPLETE message containing unknown optional IE.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_6_3

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that the MS ignores an IE which is unknown in a message for Radio Resource Management in the special cases of CIPHERING MODE COMMAND, ASSIGNMENT COMMAND and CHANNEL RELEASE.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_ChRate := C_Full)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_12man, CellChDes_12dman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltNotZero1, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_ChMod_s, FreqTCHa_21g, FreqTCHa_21d, C_Noarfcn, C_Noarfcn)			
5		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
6		(TCV_Null := OM_CphMdChg(TCV_ch, CphMod_01, TCV_CphKey))			
7	body	L!DL_DatRqCphmCmdErr	CphCmd_inv(TCV_ch, CphModeCmd_inv_03)		1.
8	L_0136	L?DL_DatInCphmCom	CphCom(CphModeCmp_01)	(P)	
9		+AssCmdGen_fh(C_Full, TSPX_TmSltNotZero1, TSPX_TscDef, TSPX_TCHHSubDef, 7, 7, '000000'B, '000000'B, Frql_02, Frql_02, CellChDes_12, CellChDes_12d, ChMod_sign_iei, MoblAllc_02, MoblAllc_02, CphMod_omit)			With power level 7, 1st unknow IE
10		(TCV_AssCmd.ch2d_at := UnknownIE)			2nd unkown IE
11		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			2.
12		L!DL_DatRqChRelErr	ChRel_inv(TCV_chTch)		3.
13	L_0137	L?DL_RelIn	DLRelInd_01	(P)	

Detailed Comments:

1. To send a CIPHERING MODE COMMAND message containing an unknown IE.
2. To send an ASSIGNMENT COMMAND message containing 2 unknown IEs.
3. To send a CHANNEL RELEASE message containing an unknown IE.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_7_1_1

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that the MS ignores the value of spare bits in the special case of the spare bits occurring in the P1 Rest Octets IE of a PAGING REQUEST TYPE 1 message. That the spare bits are ignored is checked by addressing the MS in that PAGING REQUEST message and verifying that the MS responds to that paging.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_inv_04)		1.
4	L_0138	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)	(P)	
5		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		

Detailed Comments: 1. To send a PAGING REQUEST TYPE1 message containing rest octets which are not all '2B'O.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_7_1_2

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that the MS in the MM-state "idle, updated" and in RR-idle mode ignores the value of spare bits in the special case where these spare bits are contained in SI3 rest octets IE and SI4 rest octets IE. That the MS ignores the value of the spare bits is checked by changing the LAI in those message and observing the MS initiating a location update though the spare bits do not all have the default value.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+modifysysinfo			
4	body	START T_dly(30000)			
5	L_0139	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_09)	(P)	
6		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rr, TCV_Fn))		
7	L_0140	?TIMEOUT T_dly		F	
		modifysysinfo			
8		[TSPC_PGSM OR TSPC_EGSM]			
9		L!DL_UdatRqSysinfo3	SysInfo3_inv_01(C_MCC_1, C_PLMN_1, C_LAC_2)		1.
10		L!DL_UdatRqSysinfo4	SysInfo4_inv_01(C_MCC_1, C_PLMN_1, C_LAC_2)		
11		[TSPC_DCS]			
12		L!DL_UdatRqSysinfo3 (DL_UdatRqSysinfo3.msg.csp := CellSelPara_04)	SysInfo3_inv_01(C_MCC_1, C_PLMN_1, C_LAC_2)		1.
13		L!DL_UdatRqSysinfo4 (DL_UdatRqSysinfo4.msg.csp := CellSelPara_04)	SysInfo4_inv_01(C_MCC_1, C_PLMN_1, C_LAC_2)		

Detailed Comments: 1. To send modified SYSTEM INFORMATION TYPE 3 and TYPE 4.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_7_1_3

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that the MS in the MM-state "idle, updated" and in RR-idle mode ignores the value of spare bits in the special case of the spare bits occurring in the Page Mode IE, the Spare Half Octet IE, the Channel Description IE, the Timing Advance IE, the IA Rest Octet IE, and in the IAR Rest Octet IE.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3	body	+localtree1			
		localtree1			
4		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L! DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_inv_04(TCV_Rr, TCV_Fn, TCV_slot, TCV_tsc, TimingAdv_inv, TCV_chdescr_arfcn))		1.
8	L_0141	L?DL_EstInPgRes	PagingRes(PagingRes_01)	(P)	
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
11	L_0142	L?DL_RelIn	DLRelInd_01	(P)	
12		START T_dly(10000)			
13		?TIMEOUT T_dly			
14		+localtree2			
		localtree2			
15		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
16		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
17		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_inv_02(TCV_Rr,TCV_Fn))		2.
18		START T_dly(6000)			
19		?TIMEOUT T_dly			
20		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
21	L_0143	L?DL_RacInChRq	ChReq(ChRequest_17)	P	

Detailed Comments:
1. To send an invalid IMMEDIATE ASSIGNMENT message containing arbitrary spare bits.
2. To send an invalid IMMEDIATE ASSIGNMENT REJECT message containing arbitrary spare bits.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_7_1_4

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that the MS in the MM-state "MM-Connection active" and in RR-Connected mode ignores the value of spare bits in the special case of the spare bits occurring in the Cell Channel Description IE and in the Power Command IE.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcC, TSPX_MTChRateC)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCHa1, FreqTCHa2, C_Noarfcn, C_Noarfcn)			1.
5		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
6	body	+Ciphering_on(TCV_ch)			
7		(TCV_Setup_mt.sig := Signal_01)			
8		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		2.
9		L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))		
10		L?DL_DatInAlert	AlertRcv(AlertingInd_01(TI_01))		
11		+localtree1			
12		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
13		+localtree1			
		localtree1			
14		+AssCmdGen_fh(TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, 7, 3, '000000'B, '111111'B, Frql_omit, Frql_omit, CellChDes_01, CellChDes_01d, TCV_ChMod, MoblAllc_r03, MoblAllc_r03, CphMod_04iei(TSPX_CphAlgE))			3.
15		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
16		+PostLinkRelEnd(TCV_chTch)			4.

Detailed Comments:

1. To setup a physical channel as hopping traffic channel.
2. To send SETUP message indicating full rate channel and containing signal IE.
3. To send a modified ASSIGNMENT COMMAND containing randomly chosen spare bits.
4. The assignment procedure succeeds.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_7_2

Group: GSM_L3_MS_v4220/BiBo/

Purpose: To verify that the MS in the MM-state "wait net cmd" and in RR-Connected mode ignores the value of spare bits in the special case of the spare bits occurring in the Cipher Key Seq. Number IE or in the Identity Type IE.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
4	body	L!DL_DatRqAuthRq	AuthReq_inv_01(TCV_ch, TSPX_RANDDef)		1.
5	L_0144	L?DL_DatInAuthRes	AuthRes(AuthResponse)	(P)	
6		L!DL_DatRqIdRq	IDReq(TCV_ch, IDRequest_inv_03)		2.
7	L_0145	L?DL_DatInIdRes	IDRes(IDResponse_02)	(P)	
8		+PostLinkRelEnd(TCV_ch)			

Detailed Comments:

1. To send a modified AUTHENTICATION REQUEST message containing arbitrary spare bits.
2. To send a modified IDENTITY REQUEST message containing arbitrary spare bits.

Test Case Dynamic Behaviour

Test Case Name: TC_26_5_7_3
Group: GSM_L3_MS_v4220/BiBo/
Purpose: To verify that the MS in the MM-state "connection established" and in RR-Connected mode ignores the value of spare bits in the special case of the spare bits occurring in the Calling party BCD Number IE, Calling Party Subaddress IE, Called Party Subaddress IE, Cause IE and Progress Indicator IE.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcI, C_Full)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
5		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
6	body	L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNDef, TSPX_RANDDef))		
7		L?DL_DatInAuthRes	AuthRes(AuthResponse)		
8		+Ciphering_on(TCV_ch)			
9		+localtree			
		localtree			
10		(TCV_Setup_mt.sig :=Signal_01, TCV_Setup_mt.cgpn := Cgpn_01, TCV_Setup_mt.cgps := Cgps_01, TCV_Setup_mt.cdpn := Cdpn_01, TCV_Setup_mt.cdps := Cdps_01)			
11		L!DL_DatRqSetup	SetupSnd(TCV_ch, Setup_inv_02(TI_02))		1.
12	L_0146	L?DL_DatInCallCo	CallCfm(CallConfirm_03(TI_01))	(P)	
13		L?DL_DatInAlert	AlertRcv(AlertingInd_01(TI_01))		
14		(TCV_Null := OO_HookOff())			
15		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
16		+localtree1			
17		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
18		+localtree1			
		localtree1			
19		+AssCmdGen(TCV_cellid, C_Full, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
20		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
21		L!DL_DatRqConnAck	ConnAckSnd(		

			TCV_chTch, ConnectAck_01(TI_02))		
22		L!DL_DatRqDisc	Disc_inv(TCV_chTch, Disconn_inv_04(TI_02))		
23		+CCstatuschk_01(TCV_chTch, C_U12)			2.
24		L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_03(TI_02))		
25		L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TI_01))		
26		+PostMainLinkRel(TCV_chTch)			
Detailed Comments:		1. To send a SETUP message containing arbitrary spare bits. 2. To check whether the MS enters the state U12, if no the test case fails in the test step.			

Test Case Dynamic Behaviour					
Test Case Name:		TC_26_6_1_1			
Group:		GSM_L3_MS_v4220/RR/			
Purpose:		To verify that the MS can correctly set up a dedicated SDCCH control channel and that the MS can correctly set up a dedicated TCH/FACCH control channel.			
Default:		OtherEventsFail_01			
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+Varinit_tch(C_arfcn_tchA, C_arfcn_tchAd)			
4	body	+testSdcch8			
5		[TCV_CC AND(NOT TSPC_DualRate)]			
6		+testTchf			
7		[TSPC_DualRate]			
8		+testTchh			
9		+testTchf			
10		[C_Yes]			
		testSdcch8			
11		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubA, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(TCV_tch_arfcn), FreqTCH(TCV_tch_arfcn), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			2.
12		(TCV_Tchtype := INT_TO_BIT((8 + BIT_TO_INT(TSPX_SDCCH8SubA)), 5))			
13		+channelass(TCV_ch)			3.
		testTchf			
14		+FullRateCh_A_1_nociph(C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(TCV_tch_arfcn), FreqTCH(TCV_tch_arfcn), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			4.
15		?TIMEOUT T_dly			
16		(TCV_Tchtype := '00001'B)			
17		+channelass(TCV_chTch)			5.
		testTchh			
18		+HalfRateCh_A_1_nociph(TSPX_TCHHSubA, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(TCV_tch_arfcn), FreqTCH(TCV_tch_arfcn), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			6.
19		?TIMEOUT T_dly			
20		(TCV_Tchtype := INT_TO_BIT((2 + BIT_TO_INT(TSPX_TCHHSubA)), 5))			
21		+channelass(TCV_chTch)			7.
		channelass(ch: LOGICCH)			
22		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
23		L?DL_RacInChRq(TCV_Rr := DL_RacInChRq.	ChReq(		

		msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) START T_dly(C_T_Wait)	ChRequest_17)		
24		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r(ChDescrp_nfh(TCV_Tchtype, TSPX_TmSltDef, TSPX_TscDef, TCV_tch_arfcn), TCV_Rr, TCV_Fn, TimingAdv(30)))		
25	L_0147	L?DL_EstInPgRes	PgRes(ch, PagingRes_01)	(P)	
26		+PostMainLinkRel(ch)			

Detailed Comments:

1. To setup a physical channel as non-combined ccch/sdcch, 5 slots for Tx-int, 1 retransmission
2. To setup a physical channel as SDCCH8.
3. To assign a SDCCH8 channel.
4. To setup a physical channel as full rate traffic channel.
5. To assign a full rate channel channel.
6. To setup a physical channel as half rate traffic channel.
7. To assign a half rate channel.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_1_2

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS goes to the allocated SDCCH/4 and sends a PAGING RESPONSE message containing its identity and its classmark. To verify that the MS goes to the allocated SDCCH/8 and sends a PAGING RESPONSE message containing its identity and its classmark. To verify that the MS can correctly identify its own assignment in either the Request Reference 1 or the Request Reference 2 information element in an extended assignment message. To verify that the MS only reacts to an Immediate Assignment which references one of the last 3 CHANNEL REQUEST messages from the MS.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TSPX_TimadvB), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3	body	+firstPart			
4		+secondPart			
5		+PreEnterIdleState_r02(C_Immass, TCV_slot, TCV_tsc, C_TxInt_5, C_Max_7, C_SlotNU, C_SlotNU, C_SlotNU, TimingAdv(TSPX_TimadvB), C_ATT_0, C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_1)			2.
6		+Varinit_tch(C_arfcn_tchA, C_arfcn_tchAd)			
7		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubC, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(TCV_tch_arfcn), FreqTCH(TCV_tch_arfcn), TimingAdv(TSPX_TimadvB), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
8		+CCCH_group_Paging_group(TCV_Ccd0A, TSPX_IMSI)			
9		START T_dly(40000)			
10		?TIMEOUT T_dly			
11		+thirdPart			
		firstPart			
12		[(TSPX_nPara < 9) AND(TSPX_nPara > 0) AND(TSPX_i1Para >0) AND (TSPX_i1Para <= TSPX_nPara)]			
13		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
14		(TCV_Cnt := 0)			
15		REPEAT localTree UNTIL [TCV_Cnt = TSPX_nPara]			5.
16		(TCV_Rqr := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, TSPX_i1Para, 0), TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, 10, 1))			6.
17		L!DL_UdatRqImmassx	ImmAssX(TCV_agch, ImmAsgnX_r01(TCV_Rqr.ra, TCV_Rqr.fn, TCV_Rqr9.ra, TCV_Rqr9.fn, TCV_slot, TCV_tsc, TimingAdv(		7. 3.

			TSPX_TimadvB), TCV_chdescr_arfcn))		
18	L_0148	L?DL_EstInPgRes	PgRes(TCV_ch, PagingRes_01)	(P)	
19		+PostMainLinkRel(TCV_ch)			
20		START T_dly(12000)			
21		?TIMEOUT T_dly			
22	L_0149	[(TSPX_nPara > 8) OR(TSPX_nPara < 1) OR(TSPX_i1Para <1) OR(TSPX_i1Para > TSPX_nPara)]		(I)	
		secondPart			
23		[(TSPX_kPara < 9) AND(TSPX_kPara > 3) AND(TSPX_i2Para >0) AND (TSPX_i2Para <= TSPX_kPara)]			
24		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
25		(TCV_Cnt := 0)			
26		REPEAT localTree UNTIL [TCV_Cnt = TSPX_kPara]			8.
27		(TCV_Rqr := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, TSPX_i2Para, 0), TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, 11, 1))			9. 10.
28		L!DL_UdatRqImmassx	ImmAssX(TCV_agch, ImmAsgnX_r01(TCV_Rqr.ra, TCV_Rqr.fn, TCV_Rqr9.ra, TCV_Rqr9.fn, TCV_slot, TCV_tsc, TimingAdv(TSPX_TimadvB), TCV_chdescr_arfcn))		3.
29		(TCV_Cnt := 0)			
30		[TSPX_kPara < 8]			
31		REPEAT localTree UNTIL [TCV_Cnt = 8 - TSPX_kPara]			11.
32		+ltree_noL2			
33		[TSPX_kPara = 8]			
34		+ltree_noL2			
35	L_0150	[(TSPX_kPara > 8) OR(TSPX_kPara < 4) OR(TSPX_i2Para <1) OR(TSPX_i2Para > TSPX_kPara)]		(I)	
		ltree_noL2			
36		START T_dly(3000)			
37	L_0151	L?OTHERWISE		F	
38	L_0152	?TIMEOUT T_dly		(P)	
39		START T_dly(7000)			
40		?TIMEOUT T_dly			
		thirdPart			
41		[(TSPX_rPara < 9) AND(TSPX_rPara > 3) AND(TSPX_i3Para >0) AND (TSPX_i3Para <= TSPX_rPara)]			
42		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
43		(TCV_Cnt := 0)			
44		REPEAT localTree UNTIL [TCV_Cnt = TSPX_rPara]			13.
45		(TCV_Rqr := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, TSPX_i3Para, 0), TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, 12, 1))			14.

46		L!DL_UdatRqImmassx	ImmAssX(TCV_agch, ImmAsgnX_r02(TCV_Rqr9.ra, TCV_Rqr9.fn, TCV_Rqr.ra, TCV_Rqr.fn, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(TSPX_TimadvB), TCV_tch_arfcn))		4.
47	L_0153	L?DL_EstInPgRes	PgRes(TCV_ch, PagingRes_01)	(P)	
48		+PostMainLinkRel(TCV_ch)			
49	L_0154	[(TSPX_rPara > 8) OR(TSPX_rPara < 4) OR(TSPX_i3Para <1) OR(TSPX_i3Para > TSPX_rPara)]		(I)	
		localTree			
50		L?DL_RacInChRq(TCV_Rqr.ra := DL_RacInChRq.msg.ecau_rrf, TCV_Rqr.fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		15.
51		(TCV_Cnt := TCV_Cnt +1)			
52		(TCV_Rqr10 := OC_SaveAndRetrv(TCV_Rqr, C_SAVE, TCV_Cnt, 0))			

Detailed Comments:

1. To broadcast sys_info messages with default contents for RR tests with exception for Max_Retrans that is set to 7 and also to set up a physical channel as combined CCCH/SDCCH.
2. Non combined CCCH/SDCCH, 5 slots for Tx-int, 7 retransmission.
3. To send immediate Assignment Extended message with Request reference (TCV_Rqr) for MS1, and with request reference (TCV_Rqr9) for MS2.
4. To send immediate Assignment Extended message with Request reference (TCV_Rqr) for MS2, request reference (TCV_Rqr9) for MS1.
5. Reception of the first TSPX_nPara Channel Request messages.
6. To have a request reference pertaining to the TSPX_i1Para_i-th Channel Request where TSPX_i1Para_i-th is within {max(1, TSPX_nPara - 2), TSPX_nPara)}
7. To have a request reference different from any request reference the MS has generated.
8. Reception of TSPX_kPara Channel Request messages. TSPX_kPara within the set{4...8}.
9. To have a request reference pertaining to the TSPX_i2Para_i-th Channel Request where TSPX_i2Para_i-th is within {max(1...TSPX_kPara - 3}
10. To have a request reference different from any request reference the MS has generated.
11. Channel Request 8 - TSPX_kPara Channel Requests are sent.
12. Check that the MS does not transmit any layer 2 frames for at least 3 seconds.
13. Reception of TSPX_rPara Channel Request messages.
14. To have a request reference pertaining to the TSPX_i3Para_i-th Channel Request where TSPX_i3Para_i-th is within {max(TSPX_rPara - 2...TSPX_rPara}.
15. This local tree is used to collect and store the frame number and the request reference which are included in each ChReq(ChRequest_17) primitive received.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_1_3

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS can accept an IMMEDIATE ASSIGNMENT REJECT.
To verify that the MS can respond to paging after an IMMEDIATE ASSIGNMENT REJECT is received on a different cell.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+StartCellB(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			2. The LAI is the same as cell A.
4	body	+test1			
5		+ltree_switchcelltoB			
6		+test2			
		test1			
7		[(TSPX_n1Para < 9) AND(TSPX_n1Para > 0) AND(TSPX_i4Para >0) AND (TSPX_i4Para <= TSPX_n1Para) AND(TSPX_xPara < 256) AND(TSPX_xPara > 4)]			
8		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		Cell A
9		(TCV_Cnt := 0)			
10		REPEAT localTree UNTIL [TCV_Cnt = TSPX_n1Para]			
11		(TCV_Rqr := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, TSPX_i4Para, 0), TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, 10, 1), TCV_Rqr10 := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, 11, 1), TCV_Rqr11 := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, 12, 1), TCV_Time := ((TSPX_xPara + 1) * 1000))			
12		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_r01(TCV_Rqr9, TCV_Rqr, TCV_Rqr10, TCV_Rqr11, TSPX_xPara, 0))		
13		START T_dly(TCV_Time)			
14		REPEAT continuePaging UNTIL [TCV_Time = 0]			3.
15		(TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, 10, 1), TCV_Rqr10 := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, 11, 1), TCV_Rqr11 := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, 12, 1))			
16		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch,		

			ImmAsgnRej_r01(TCV_Rqr9, TCV_Rqr10, TCV_Rqr, TCV_Rqr11, 0, 255))		
17	L_0155	[(TSPX_n1Para > 8) OR(TSPX_n1Para < 1) OR(TSPX_i4Para <1) OR(TSPX_i4Para > TSPX_n1Para) AND(TSPX_xPara > 255) AND(TSPX_xPara < 5)]		(I)	
		test2			
18		START T_dly(12000)			
19		?TIMEOUT T_dly			
20		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		Cell B
21	L_0156	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq_01(C_RACH_B_1, ChRequest_01)	(P)	5.
22		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_r02(TCV_Rr, TCV_Fn))		
		continuePaging			
23		L!DL_UdatRqPg1Rq START T_dly1(1177)	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		Every 5 multiframes
24		L?DL_RacInChRq (TCV_Rqr.ra := DL_RacInChRq. msg.ecau_rrf, TCV_Rqr.fn := DL_RacInChRq.fn) READTIMER T_dly(TCV_Time), CANCEL T_dly1, CANCEL T_dly	ChReq_01(C_RACH_A_1, ChRequest_01)		Received at least one channel request within x+1s.
25		(TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_SAVE,(TCV_Cnt + 1), 0))			
26		[(TCV_Time <= (TSPX_xPara * 1000)) AND(TCV_Time > 0)]			received within x s.
27	L_0157	(TCV_Time := 0)		(F)	Fail return
28		[(TCV_Time >(TSPX_xPara * 1000)) OR (TCV_Time = 0)]			received btw x, x+1 s.
29	L_0158	(TCV_Time := 0)		(P)	Pass return
30		?TIMEOUT T_dly1 READTIMER T_dly(TCV_Time)			
31		[TCV_Time = 0]			x+1 s
32		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
33	L_0159	L?DL_RacInChRq (TCV_Rqr.ra := DL_RacInChRq. msg.ecau_rrf, TCV_Rqr.fn := DL_RacInChRq.fn)	ChReq_01(C_RACH_A_1, ChRequest_01)	(P)	4.
34		(TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_SAVE,(TCV_Cnt + 1), 0))			
35		[TCV_Time <> 0]			Return
		localTree			
36	L_0160	L?DL_RacInChRq (TCV_Rqr.ra := DL_RacInChRq. msg.ecau_rrf, TCV_Rqr.fn := DL_RacInChRq.fn)	ChReq_01(C_RACH_A_1, ChRequest_01)	(P)	
37		(TCV_Cnt := TCV_Cnt +1)			
38		(TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_SAVE, TCV_Cnt, 0))			
		ltree_switchcelltoB			
39		+Varinit_fix(C_CellB, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
40		(TCV_Null := OM_ChangeRFOf2Cells(C_CellA, C_E_notsuitable_fSel, C_CellB, C_E_suitable))			

Detailed Comments: 1. To setup a physical channel as combined CCCH/SDCCH, 5 for Tx-int, 7 retransmission used

	as cell A. 2. To setup a physical channel as combind ccch/sdcch4 used as cell B. 3. To send PAGING REQUEST message every 5 multiframes of the mobile station's paging subgroup for x+1 seconds. 4. The MS responds the paging after x+1 s. expires, pass. 5. The MS responds the paging in cell B, pass.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_1_4
Group: GSM_L3_MS_v4220/RR/
Purpose: To verify that the MS ignores an assignment for another MS while it is waiting for an assignment of its own.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
4		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := OC_FnInc(DL_RacInChRq.fn, 2))	ChReq(ChRequest_17)		2.
5		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubB, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
6		START T_dly(2000)			
7		?TIMEOUT T_dly			
8	L_0161	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)	(P)	
9		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_r02(TCV_Rr, TCV_Fn))		
10		+localtree1			
11	L_0162	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)	(P)	
12		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_r02(TCV_Rr, TCV_Fn))		
13		?TIMEOUT T_dly			
14		+localtree1			
		localtree1			
15		START T_dly(6000)			
16		?TIMEOUT T_dly			
17		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
18		L?DL_RacInChRq (TCV_Rr := OC_BinAdd(DL_RacInChRq.msg.ecau_rrf, '00000010'B), TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		3.
19		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubB, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		

20		START T_dly(2000)			
21		?TIMEOUT T_dly			
22		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
23	L_0163	L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_r02(TCV_Rr, TCV_Fn))	P	
24		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
25	L_0164	L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_r02(TCV_Rr, TCV_Fn))	(P)	
26		?TIMEOUT T_dly			

Detailed Comments:

1. To setup a physical channel as combined bcch/sdcch, default parameters for cell A.
2. making a frame number is of 2 higher.
3. Making a wrong request reference.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_1_5

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS correctly responds to an IMMEDIATE ASSIGNMENT message sent after an IMMEDIATE ASSIGNMENT REJECT message.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_7, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+Varinit_tch(C_arfcn_tchA, C_arfcn_tchAd)			
4		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(TCV_tch_arfcn), FreqTCH(TCV_tch_arfcn), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			2.
5	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		1st channel request
7		L?DL_RacInChRq (TCV_Rr1 := DL_RacInChRq. msg.ecau_rrf, TCV_Fn1 := DL_RacInChRq.fn)	ChReq(ChRequest_17)		2nd channel request
8		L?DL_RacInChRq	ChReq(ChRequest_17)		3rd ChReq
9		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_04(TCV_Rr, TCV_Fn))		Ref to 1st, wait 6s.
10		START T_dly1(OC_Random(750, 1250))			
11		?TIMEOUT T_dly1			
12		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r02(TCV_Rr1, TCV_Fn1, TSPX_SDCCH8SubD, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), TCV_tch_arfcn))		Ref. to 2nd
13	L_0165	L?DL_EstInPgRes	PgRes(TCV_ch, PagingRes_01)	(P)	3.
14		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To setup a physical channel as non-combined CCCH/SDCCH, Tx-int= 7 , retransmission = 7.
2. To setup aphysical channel as SDCCH8.
3. The expected PAGING RESPONSE received on the correct channel (TSPX_SDCCH8SubD)

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_2_1_1

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that the MS is able to determine its CCCH group and paging group correctly and that the MS responds correctly to various PAGING REQUEST TYPE 1 messages when the page mode is set to normal paging. All valid ways of addressing the MS are tested. It is tested that the MS responds with the same type of identity that is used in the PAGING REQUEST TYPE 1 message. It is tested that the MS ignores fill paging.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_S2, C_S4, C_S6, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(30), INT_TO_BIT(TSPX_AGBLKS1, 3), TSPX_CcchConf1, INT_TO_BIT((TSPX_PAMFRMS1-2), 3), C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubC, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+local_tree			
4	body	+test1			
5		+test2			
6		+test3			
7		+test4			
8		+test5			
		test1			
9		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_05)		1.
10		+local_immass			
11	L_0166	L?DL_EstInPgRes	PagingRes(PagingRes_r02)	(P)	
12		+PostMainLinkRel(TCV_ch)			
		test2			
13		?TIMEOUT T_dly			
14		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_r01)		2.
15		+local_immass			
16	L_0167	L?DL_EstInPgRes	PagingRes(PagingRes_r01)	(P)	
17		+PostMainLinkRel(TCV_ch)			
		test3			
18		?TIMEOUT T_dly			
19		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_r02)		3.
20		+local_immass			
21	L_0168	L?DL_EstInPgRes	PagingRes(PagingRes_r02)	(P)	
22		+PostMainLinkRel(TCV_ch)			
		test4			
23		?TIMEOUT T_dly			
24		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_r03)		4.
25		+local_immass			

26	L_0169	L?DL_EstInPgRes	PagingRes(PagingRes_r01)	(P)	
27		+PostMainLinkRel(TCV_ch)			
		test5			
28		?TIMEOUT T_dly			
29		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_r04)		5.
30		START T_dly(1000)			
31	L_0170	?TIMEOUT T_dly		P	
32	L_0171	L?OTHERWISE		F	6.
		local_immass			
33		L?DL_RacInChRq	ChReq(ChRequest_17)		
34		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
35		[(TSPX_CcchConf1 = C_cch_1nonComb) OR (TSPX_CcchConf1 = C_cch_2nonComb) OR (TSPX_CcchConf1 = C_cch_3nonComb) OR (TSPX_CcchConf1 = C_cch_4nonComb)]			
36		L!DL_UdatRqImmass START T_dly(12000)	ImmAss(TCV_agch, ImmAsgn_r02(TCV_Rr, TCV_Fn, TSPX_SDCCH8SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), TCV_tch_arfcn))		
37		[TSPX_CcchConf1 = C_cch_1Comb]			
38		L!DL_UdatRqImmass START T_dly(12000)	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubC, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
39	L_0172	[(TSPX_CcchConf1 = C_cch_2Comb) OR (TSPX_CcchConf1 = C_cch_3Comb) OR (TSPX_CcchConf1 = C_cch_4Comb)]		I	
		local_tree			
40		[TSPX_CcchConf1 = C_cch_1Comb]			
41		[TSPX_CcchConf1 <> C_cch_1Comb]			
42		+Varinit_tch(C_arfcn_tchA, C_arfcn_tchAd)			
43		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(TCV_tch_arfcn), FreqTCH(TCV_tch_arfcn), TimingAdv(30), INT_TO_BIT(TSPX_AGBLKS1, 3), TSPX_CcchConf1, INT_TO_BIT((TSPX_PAMFRMS1-2), 3))			

Detailed Comments:

1. Within the paging request message, the 1st MI contains IMSI of the MS, the 2nd is absent.
2. Within the paging request message, the 1st MI contains TMSI of the MS, the 2nd has the IMSI of an another MS.
3. Within the paging request message, the 1st MI contains TMSI of an another MS, the 2nd has the IMSI of the MS.
4. Within the paging request message, the 1st MI contains TMSI of an another MS, the 2nd has the TMSI of the MS.
5. Within the paging request message, the 1st MI contains TMSI of the MS together with type of no identity, the 2nd is absent .
6. If received any L3 frame FAIL.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_2_1_2
Group: GSM_L3_MS_v4220/RR/
Purpose: To test that the MS is able to determine its CCCH group and paging group correctly and that the MS responds correctly to various PAGING REQUEST TYPE 2 messages when the page mode is set to normal paging. All valid ways of addressing the MS are tested. It is tested that the MS responds with the same type of identity that is used in the PAGING REQUEST TYPE 2 message. It is tested that the MS ignores a PAGING REQUEST TYPE 2 message that does not address it.
Default: OtherEventsFail_01
Comments: The configuration is Max-Retrans = 2, combined CCCH/BCCH, BS-AG-BLKS-RES = 2, and BS-PA-MFRMS = 3.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_S2, C_S4, C_S6, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(30), INT_TO_BIT(TSPX_AGBLKS2, 3), TSPX_CcchConf2, INT_TO_BIT((TSPX_PAMFRMS2-2), 3), C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubB, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+local_tree			
4	body	+test1			
5		+test2			
6		+test3			
7		+test4			
8		+test5			
		test1			
9		L!DL_UdatRqPg2Rq	PgReq2(TCV_PgCh, TCV_Pgg, PgReqTp2_r01)		1.
10		+localtree			
11	L_0173	L?DL_EstInPgRes	PagingRes(PagingRes_r01)	(P)	
12		+PostMainLinkRel(TCV_ch)			
		test2			
13		?TIMEOUT T_dly			
14		L!DL_UdatRqPg2Rq	PgReq2(TCV_PgCh, TCV_Pgg, PgReqTp2_r02)		2.
15		+localtree			
16	L_0174	L?DL_EstInPgRes	PagingRes(PagingRes_r01)	(P)	
17		+PostMainLinkRel(TCV_ch)			
		test3			
18		?TIMEOUT T_dly			
19		L!DL_UdatRqPg2Rq	PgReq2(TCV_PgCh, TCV_Pgg, PgReqTp2_r03)		3.
20		+localtree			
21	L_0175	L?DL_EstInPgRes	PagingRes(PagingRes_r01)	(P)	
22		+PostMainLinkRel(TCV_ch)			
		test4			
23		?TIMEOUT T_dly			
24		L!DL_UdatRqPg2Rq	PgReq2(TCV_PgCh, TCV_Pgg,		4.

			PgReqTp2_r04)		
25		+localtree			
26	L_0176	L?DL_EstInPgRes	PagingRes(PagingRes_r02)	(P)	
27		+PostMainLinkRel(TCV_ch)			
		test5			
28		?TIMEOUT T_dly			
29		L!DL_UdatRqPg2Rq	PgReq2(TCV_PgCh, TCV_Pgg, PgReqTp2_r05)		5.
30		START T_dly(1000)			
31	L_0177	?TIMEOUT T_dly		P	
32	L_0178	L?OTHERWISE		F	6.
		localtree			
33		L?DL_RacInChRq	ChReq(ChRequest_17)		
34		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
35		[(TSPX_CcchConf2 = C_cch_1nonComb) OR (TSPX_CcchConf2 = C_cch_2nonComb) OR (TSPX_CcchConf2 = C_cch_3nonComb) OR (TSPX_CcchConf2 = C_cch_4nonComb)]			
36		L!DL_UdatRqImmass START T_dly(C_T_Wait)	ImmAss(TCV_agch, ImmAsgn_r02(TCV_Rr, TCV_Fn, TSPX_SDCCH8SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), TCV_tch_arfcn))		
37		[TSPX_CcchConf2 = C_cch_1Comb]			
38		L!DL_UdatRqImmass START T_dly(C_T_Wait)	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubB, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
39	L_0179	[(TSPX_CcchConf2 = C_cch_2Comb) OR (TSPX_CcchConf2 = C_cch_3Comb) OR (TSPX_CcchConf2 = C_cch_4Comb)]		I	
		local_tree			
40		[TSPX_CcchConf2 = C_cch_1Comb]			
41		[TSPX_CcchConf2 <> C_cch_1Comb]			
42		+Varinit_tch(C_arfcn_tchA, C_arfcn_tchAd)			
43		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(TCV_tch_arfcn), FreqTCH(TCV_tch_arfcn), TimingAdv(30), INT_TO_BIT(TSPX_AGBLKS2, 3), TSPX_CcchConf2, INT_TO_BIT((TSPX_PAMFRMS2-2), 3))			

Detailed Comments:

1. Within the paging request message, the 1st MI contains TMSI of the IUT, the 2nd has the TMSI of an another MS, the 3rd is absent.
2. Within the paging request message, the 1st MI contains TMSI of an another MS, the 2nd has the TMSI of the IUT, the 3rd is absent.
3. Within the paging request message, the 1st and 2nd MI contain TMSI of an another MS, the 3rd has the TMSI of the IUT.
4. Within the paging request message, the 1st and 2nd MI contain TMSI of an another MS, the 3rd has the IMSI of the IUT.
5. Within the paging request message, the 1st and 2nd MI contain TMSI of an another MS, the 3rd has the TMSI of the IUT but with the type of no identity.
6. If received any L3 frame FAIL.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_2_1_3

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that the MS is able to determine its CCCH group and paging group correctly and that the MS responds correctly to various PAGING REQUEST TYPE 3 messages when the page mode is set to normal paging. All valid ways of addressing the MS are tested.

Default: OtherEventsFail_01

Comments: The configuration is Max-Retrans = 2, 2 non-combined CCCH/BCCH, BS-AG-BLKS-RES = 5, and BS-PA-MFRMS = 6.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_S2, C_S4, C_S6, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(30), INT_TO_BIT(TSPX_AGBLKS3, 3), TSPX_CcchConf3, INT_TO_BIT((TSPX_PAMFRMS3-2), 3), C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubB, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+local_tree			
4	body	+test1			
5		+test2			
6		+test3			
7		+test4			
		test1			
8		L!DL_UdatRqPg3Rq	PgReq3(TCV_PgCh, TCV_Pgg, PgReqTp3_r01)		2.
9		+local_immass			
10	L_0180	L?DL_EstInPgRes	PagingRes(PagingRes_r01)	(P)	
11		+PostMainLinkRel(TCV_ch)			
		test2			
12		?TIMEOUT T_dly			
13		L!DL_UdatRqPg3Rq	PgReq3(TCV_PgCh, TCV_Pgg, PgReqTp3_r02)		3.
14		+local_immass			
15	L_0181	L?DL_EstInPgRes	PagingRes(PagingRes_r01)	(P)	
16		+PostMainLinkRel(TCV_ch)			
		test3			
17		?TIMEOUT T_dly			
18		L!DL_UdatRqPg3Rq	PgReq3(TCV_PgCh, TCV_Pgg, PgReqTp3_r03)		4.
19		+local_immass			
20	L_0182	L?DL_EstInPgRes	PagingRes(PagingRes_r01)	(P)	
21		+PostMainLinkRel(TCV_ch)			
		test4			
22		?TIMEOUT T_dly			
23		L!DL_UdatRqPg3Rq	PgReq3(TCV_PgCh, TCV_Pgg, PgReqTp3_r04)		5.
24		+local_immass			
25	L_0183	L?DL_EstInPgRes	PagingRes(PagingRes_r01)	(P)	

26		+PostMainLinkRel(TCV_ch)			
		local_immass			
27		L?DL_RacInChRq	ChReq(ChRequest_17)		
28		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
29		[(TSPX_CcchConf3 = C_cch_1nonComb) OR (TSPX_CcchConf3 = C_cch_2nonComb) OR (TSPX_CcchConf3 = C_cch_3nonComb) OR (TSPX_CcchConf3 = C_cch_4nonComb)]			
30		L!DL_UdatRqImmass START T_dly(C_T_Wait)	ImmAss(TCV_agch, ImmAsgn_r02(TCV_Rr, TCV_Fn, TSPX_SDCCH8SubE, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), TCV_tch_arfcn))		
31		[TSPX_CcchConf3 = C_cch_1Comb]			
32		L!DL_UdatRqImmass START T_dly(C_T_Wait)	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubB, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
33	L_0184	[(TSPX_CcchConf3 = C_cch_2Comb) OR (TSPX_CcchConf3 = C_cch_3Comb) OR (TSPX_CcchConf3 = C_cch_4Comb)]		I	
		local_tree			
34		[TSPX_CcchConf3 = C_cch_1Comb]			
35		[TSPX_CcchConf3 <> C_cch_1Comb]			
36		+Varinit_tch(C_arfcn_tchA, C_arfcn_tchAd)			
37		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubE, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(TCV_tch_arfcn), FreqTCH(TCV_tch_arfcn), TimingAdv(30), INT_TO_BIT(TSPX_AGBLKS3, 3), TSPX_CcchConf3, INT_TO_BIT((TSPX_PAMFRMS3-2), 3))			

Detailed Comments:

1. To setup three physical channels two as BCCH/CCCH's and one as SDCCH8 channel.
2. Within the paging request message, the 1st MI contains TMSI of the IUT, the 2nd, 3rd and 4th have the TMSIs of another MSs.
3. Within the paging request message, the 2nd MI contains TMSI of the IUT, the 1st, 3rd and 4th have the TMSIs of another MSs.
4. Within the paging request message, the 3rd MI contains TMSI of the IUT, the 1st, 2nd and 4th have the TMSIs of another MSs.
5. Within the paging request message, the 4th MI contains TMSI of the IUT, the 1st, 2nd and 3rd have the TMSIs of another MSs.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_2_2
Group: GSM_L3_MS_v4220/RR/
Purpose: To test that the MS is operating in the extended page mode when this is ordered by the SS in either a PAGING REQUEST message or an IMMEDIATE ASSIGNMENT message.
Default: OtherEventsFail
Comments: The configuration is Max-Retrans = 2, 1 non-combined CCCH/BCCH, BS-AG-BLKS-RES = 7, and BS-PA-MFRMS = 9.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_S2, C_S4, C_S6, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(30), INT_TO_BIT(TSPX_AGBLKS3, 3), TSPX_CcchConf3, INT_TO_BIT((TSPX_PAMFRMS3-2), 3), C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubB, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3	body	+test1			
4		+test2			
5		+test3			
6		+test4			
7		+test5			
		test1			
8		(TCV_Null := OM_2Msgs(TCV_PgCh, TCV_Pgg, C_NxtButOne))			
9		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_06)		
10		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		2.
11		L?DL_RacInChRq (TCV_Rqr.ra := DL_RacInChRq. msg.ecau_rrf, TCV_Rqr.fn := DL_RacInChRq.fn, TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_SAVE, 1, 0))	ChReq(ChRequest_02)		
12	L_0185	L?DL_RacInChRq (TCV_Rqr.ra := DL_RacInChRq. msg.ecau_rrf, TCV_Rqr.fn := DL_RacInChRq.fn, TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_SAVE, 2, 0))	ChReq(ChRequest_02)	(P)	
13		L!DL_UdatRqImmassRej START T_dly(C_T_Wait)	ImmAssRej(TCV_agch, ImmAsgnRej_03(TCV_Rqr.ra, TCV_Rqr.fn))		3.
		test2			
14		?TIMEOUT T_dly			
15		(TCV_Null := OM_2Msgs(TCV_PgCh, TCV_Pgg, C_NxtButOne), TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, 10, 1))			
16		+gsmOrDcs			4.
17		L!DL_UdatRqPg2Rq	PgReq2(TCV_PgCh, TCV_Pgg, PgReqTp2_04)		5.
18		L?DL_RacInChRq(TCV_Rqr.ra := DL_RacInChRq. msg.ecau_rrf, TCV_Rqr.fn := DL_RacInChRq.fn, TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_SAVE, 3, 0))	ChReq(ChRequest_02)		
19	L_0186	L?DL_RacInChRq(TCV_Rqr.ra := DL_RacInChRq. msg.ecau_rrf, TCV_Rqr.fn	ChReq(ChRequest_02)	(P)	

		:= DL_RacInChRq.fn, TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_SAVE, 4, 0))			
20		L!DL_UdatRqImmassRej START T_dly(C_T_Wait)	ImmAssRej(TCV_agch, ImmAsgnRej_03(TCV_Rqr.ra, TCV_Rqr.fn))		3.
		test3			
21		?TIMEOUT T_dly			
22		(TCV_Null := OM_2Msgs(TCV_PgCh, TCV_Pgg, C_NxtButOne), TCV_Rqr9 := OC_SaveAndRetrv(TCV_Rqr, C_RETRV, 11, 1))			
23		+gsmOrDcs1			6.
24		L!DL_UdatRqPg3Rq	PgReq3(TCV_PgCh, TCV_Pgg, PgReqTp3_r05)		7.
25		+ltree_chreq			
		test4			
26		?TIMEOUT T_dly			
27		(TCV_Null := OM_2Msgs(TCV_PgCh, TCV_Pgg, C_NxtButOne))			
28		L!DL_UdatRqPg3Rq	PgReq3(TCV_PgCh, TCV_Pgg, PgReqTp3_01)		8.
29		L!DL_UdatRqPg2Rq	PgReq2(TCV_PgCh, TCV_Pgg, PgReqTp2_r06)		9.
30		+ltree_chreq			
		test5			
31		?TIMEOUT T_dly			
32		(TCV_Null := OM_2Msgs(TCV_PgCh, TCV_Pgg, C_NxtButOne))			
33		L!DL_UdatRqPg2Rq	PgReq2(TCV_PgCh, TCV_Pgg, PgReqTp2_02)		8.
34		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_05)		10.
35		+ltree_chreq			
36		?TIMEOUT T_dly			
		ltree_chreq			
37		L?DL_RacInChRq	ChReq(ChRequest_02)		
38	L_0187	L?DL_RacInChRq(TCV_Rqr.ra := DL_RacInChRq.msg.ecau_rrf, TCV_Rqr.fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)	(P)	
39		L!DL_UdatRqImmassRej START T_dly(C_T_Wait)	ImmAssRej(TCV_agch, ImmAsgnRej_03(TCV_Rqr.ra, TCV_Rqr.fn))		
		gsmOrDcs			
40		[(TSPX_CcchConf3 = C_cch_1nonComb) OR (TSPX_CcchConf3 = C_cch_2nonComb) OR (TSPX_CcchConf3 = C_cch_3nonComb) OR (TSPX_CcchConf3 = C_cch_4nonComb)]			
41		L!DL_UdatRqImmass_sp	ImmAssSp(TCV_PgCh, TCV_Pgg, ImmAsgn_nfh(TCV_Rqr9.ra, TCV_Rqr9.fn, TCV_Tchtype, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30),		4.

			TCV_tch_arfcn, C_extended_paging))		
42		[TSPX_CcchConf3 = C_cch_1Comb]			
43		L!DL_UdatRqImmass_sp	ImmAssSp(TCV_PgCh, TCV_Pgg, ImmAsgn_nfh(TCV_Rqr9.ra, TCV_Rqr9.fn, TCV_Tchtype, TCV_slot, TCV_tsc, TimingAdv(30), TCV_chdescr_arfcn, C_extended_paging))		4.
44	L_0188	[(TSPX_CcchConf3 = C_cch_2Comb) OR (TSPX_CcchConf3 = C_cch_3Comb) OR (TSPX_CcchConf3 = C_cch_4Comb)]		I	
		gsmOrDcs1			
45		[(TSPX_CcchConf3 = C_cch_1nonComb) OR (TSPX_CcchConf3 = C_cch_2nonComb) OR (TSPX_CcchConf3 = C_cch_3nonComb) OR (TSPX_CcchConf3 = C_cch_4nonComb)]			
46		L!DL_UdatRqImmassx_sp	ImmAssXSp(TCV_PgCh, TCV_Pgg, ImmAsgnX_01(TCV_Rqr9.ra, TCV_Rqr9.fn, TSPX_SDCCH8SubDef, TSPX_SDCCH8SubA, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), TCV_tch_arfcn))		6.
47		[TSPX_CcchConf3 = C_cch_1Comb]			
48		L!DL_UdatRqImmassx_sp	ImmAssXSp(TCV_PgCh, TCV_Pgg, ImmAsgnX_r03(TCV_Rqr9.ra, TCV_Rqr9.fn, TSPX_SDCCH4SubB, TSPX_SDCCH4SubA, TCV_slot, TCV_tsc, TimingAdv(30), TCV_chdescr_arfcn))		6.
49	L_0189	[(TSPX_CcchConf3 = C_cch_2Comb) OR (TSPX_CcchConf3 = C_cch_3Comb) OR (TSPX_CcchConf3 = C_cch_4Comb)]		I	
		ltree_sdcch			
50		[(TSPX_CcchConf3 = C_cch_1nonComb) OR (TSPX_CcchConf3 = C_cch_2nonComb) OR (TSPX_CcchConf3 = C_cch_3nonComb) OR (TSPX_CcchConf3 = C_cch_4nonComb)]			
51		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(C_arfcn_tchA), FreqTCH(C_arfcn_tchAd), TimingAdv(30), INT_TO_BIT(TSPX_AGBLKS3, 3), TSPX_CcchConf3, INT_TO_BIT((TSPX_PAMFRMS3-2), 3))			
52		(TCV_Tchtype := INT_TO_BIT((8 + BIT_TO_INT(TSPX_SDCCH8SubDef)), 5))			
53		[TSPC_PGSM OR TSPC_EGSM]			
54		(TCV_tch_arfcn:= C_arfcn_tchA)			
55		[TSPC_DCS]			
56		(TCV_tch_arfcn:= C_arfcn_tchAd)			
57		[TSPX_CcchConf3 = C_cch_1Comb]			
58		(TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubB)), 5))			
59	L_019	[(TSPX_CcchConf3 = C_cch_2Comb) OR		I	

	0	(TSPX_CcchConf3 = C_cch_3Comb) OR (TSPX_CcchConf3 = C_cch_4Comb)]			

Detailed Comments:

1.Tx-integer = 5, Max-Retrans = 2.

2.To send PAGING REQUEST TYPE 1 message with normal page mode in the next but one paging subblock on the same CCCH as previous paging message.

3.The page mode = normal paging, wait time = 5 seconds.

4.The page mode = "extended paging", request reference is different from any one already sent by the MS.

5.The page mode = "same as before", address the MS by TMSI, in the next but one paging subblock on the same CCCH.

6.The page mode = "extended paging", request reference is different from any one already sent by the MS. The ImmAsgnX message is sent on the MS paging channel.

7.The page mode = "extended paging", address the MS by TMSI, in the next but one paging subblock on the same CCCH.

8. Extended paging, not address the MS.

9. The page mode = "same as before", address the MS with IMSI.

10. normal paging, address the MS by IMSI.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_2_3_1

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that the MS correctly determines its new paging subchannel when the CCCH structure is changed from non-combined to combined and when the number of CCCHs is changed.

Default: OtherEventsFail

Comments: The configuration is Max-Retrans = 2, 1 non-combined CCCH/BCCH, BS-AG-BLKS-RES and, BS-PA-MFRMS are controlled by PIXIT parameters.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		[(TSPX_PAMFRMS1 < 7) AND (TSPX_PAMFRMS1 >= 0)]			6.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_S2, C_S4, C_S6, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(TSPX_TimadvB), INT_TO_BIT(TSPX_AGBLKS1, 3), C_cch_1nonComb, INT_TO_BIT((TSPX_PAMFRMS1-2), 3), C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4	body	+test1			
5		+test2			
6		+test3('001'B)			
7		+secondexec			
8	L_0191	[(TSPX_PAMFRMS1 > 6) OR(TSPX_PAMFRMS1 < 0)]		I	6.
		secondexec			
9		+PreEnterIdleState_r02(C_Immass, TCV_slot, TCV_tsc, C_TxInt_5, C_Max_2, C_S2, C_S4, C_S6, TimingAdv(TSPX_TimadvB), C_ATT_0, INT_TO_BIT(TSPX_AGBLKS1, 3), C_cch_1nonComb, INT_TO_BIT((TSPX_PAMFRMS1 - 2), 3), C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1)			1.
10		+ltree_waitSelectPaging			
11		+test1			
12		+test2			
13		+test3('010'B)			
		test1			
14		(TCV_Null := OM_2Msgs(TCV_PgCh, TCV_Pgg, C_BfReOcc))			
15		L!DL_UdatRqImmassx_sp	ImmAssXSp(TCV_PgCh, TCV_Pgg, ImmAsgnX_03(TCV_slot, TCV_tsc, TimingAdv(TSPX_TimadvB), TCV_chdescr_arfcn))		
16		L!DL_UdatRqPg2Rq	PgReq2(TCV_PgCh, TCV_Pgg, PgReqTp2_01)		3.
17		+ltree_chreq			
		test2			
18		(TCV_Null := OM_PgFill(C_CellA, PgReqTp1Reorg))			
19		?TIMEOUT T_dly			
20		L!DL_UdatRqPg2Rq	PgReq2(TCV_PgCh, INT_TO_BIT(		4.

			TSPX_PgSubch, 8), PgReqTp2_03)		
21		+ltree_chreq			
		test3(cch_con:B_3)			
22		START T_dly(1000)			Ensure having sent ImmAssRej_01
23		?TIMEOUT T_dly			
24		+PreEnterIdleState_r02(C_Immass, TCV_slot, TCV_tsc, C_TxInt_5, C_Max_2, C_S2, C_S4, C_S6, TimingAdv(TSPX_TimadvB), C_ATT_0, C_BABR_2, cch_con, C_BPM_7, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1)			
25		+ltree_waitSelectPaging			
26		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		5.
27		+ltree_chreq			
28		?TIMEOUT T_dly			
29		L!DL_UdatRqPg2Rq	PgReq2(TCV_PgCh, TCV_Pgg, PgReqTp2_01)		5.
30		+ltree_chreq			
31		?TIMEOUT T_dly			
		ltree_chreq			
32		L?DL_RacInChRq	ChReq(ChRequest_02)		
33	L_0192	L?DL_RacInChRq(TCV_Rqr.ra := DL_RacInChRq.msg.ecau_rrf, TCV_Rqr.fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)	(P)	
34		L!DL_UdatRqImmassRej START T_dly(C_T_Wait)	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rqr.ra, TCV_Rqr.fn))		5s wait
		ltree_waitSelectPaging			
35		(TCV_Null := OM_PgFill(C_CellA, PgReqTp1Reorg))			
36		START T_dly(C_T_Wait)			5s wait
37		+CCCH_group_Paging_group(TCV_Ccd0A, TSPX_IMSI)			
38		+SelectPagingCh(C_CellA)			
39		?TIMEOUT T_dly			
40		(TCV_Null := OM_PgFill(C_CellA, PgReqTp1Norm))			
41		START T_dly(3000)			3s wait
42		?TIMEOUT T_dly			

Detailed Comments:

1. Tx-integer = 5, Max-Retrans = 2 and CCCH-CONF = "1 basic physical channel used for CCCH not combined with SDCCHs", BS-AG-BLKS-RES and BS-PA-MFRMS are from PIXIT.
2. Page mode = "paging reorgnisation", not address the MS.
3. Paging mode = "normal", address the MS by TMSI.
4. Page mode = "paging reorgnisation", address the MS.
5. Paging mode = "normal", address the MS by TMSI.
6. BS-PA-MFRMS shall not be set to 9 for this test case.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_2_3_2
Group: GSM_L3_MS_v4220/RR/
Purpose: To test that the MS is operating in the "paging reorganisation" page mode when this is ordered by the SS and the MS is paged in its former access grant channel.
Default: OtherEventsFail
Comments: The configuration is Max-Retrans = 1, CCCH-CONF, BS-AG-BLKS-RES and BS-PA-MFRMS are controlled by PIXIT parameters with the constraint that BS-AG-BLKS-RES > 0.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		[((TSPX_CcchConf2 = C_cch_1nonComb) OR (TSPX_CcchConf2 = C_cch_2nonComb) OR (TSPX_CcchConf2 = C_cch_3nonComb) OR (TSPX_CcchConf2 = C_cch_4nonComb)) AND(TSPX_AGBLKS2 > 0) AND(TSPX_AGBLKS2 < 8) OR (TSPX_CcchConf2 = C_cch_1Comb) AND(TSPX_AGBLKS2 > 0) AND(TSPX_AGBLKS2 < 3)]			4.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_S2, C_S4, C_S6, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(TSPX_TimadvB), INT_TO_BIT(TSPX_AGBLKS2, 3), TSPX_CcchConf2, INT_TO_BIT((TSPX_PAMFRMS2-2), 3), C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4	body	(TCV_Null := OM_2Msgs(TCV_PgCh, TCV_Pgg, C_FmrAGB))			
5		[(TSPX_CcchConf2 = C_cch_1nonComb) OR (TSPX_CcchConf2 = C_cch_2nonComb) OR (TSPX_CcchConf2 = C_cch_3nonComb) OR (TSPX_CcchConf2 = C_cch_4nonComb)]			
6		L!DL_UdatRqImmassx_sp	ImmAssXSp(TCV_PgCh, TCV_Pgg, ImmAsgnX_04(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(TSPX_TimadvB), TCV_chdescr_arfcn))		
7		+localtree			
8		[TSPX_CcchConf2 = C_cch_1Comb]			
9		L!DL_UdatRqImmassx_sp	ImmAssXSp(TCV_PgCh, TCV_Pgg, ImmAsgnX_02(TCV_slot, TCV_tsc, TimingAdv(TSPX_TimadvB), TCV_chdescr_arfcn))		
10		+localtree			
11	L_0193	[(TSPX_CcchConf2 = C_cch_2Comb) OR (TSPX_CcchConf2 = C_cch_3Comb) OR (TSPX_CcchConf2 = C_cch_4Comb)]		I	Reserved
12	L_0194	[(((TSPX_CcchConf2 = C_cch_1nonComb) OR (TSPX_CcchConf2 = C_cch_2nonComb) OR (TSPX_CcchConf2 = C_cch_3nonComb) OR (TSPX_CcchConf2 = C_cch_4nonComb)) AND(TSPX_AGBLKS2 > 7)) OR ((TSPX_CcchConf2 = C_cch_1Comb) AND(TSPX_AGBLKS2 > 2)) OR(TSPX_AGBLKS2 < 1)]		I	4.
		localtree			

13		L!DL_UdatRqPg2Rq	PgReq2(TCV_PgCh, TCV_Pgg, PgReqTp2_01)		3.
14		L?DL_RacInChRq	ChReq(ChRequest_02)		
15	L_0195	L?DL_RacInChRq (TCV_Rqr.ra := DL_RacInChRq. msg.ecau_rrf, TCV_Rqr.fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)	(P)	
16		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rqr.ra, TCV_Rqr.fn))		

Detailed Comments:

1. Tx-integer = 5, Max-Retrans = 1 and CCCH-CONF, BS-AG-BLKS-RES and BS-PA-MFRMS are controlled by PIXIT parameters.
2. paging reorgnisation.
3. send in former access grant block.
4. BS-AG-BLKs-RES > 0.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_2_4
Group: GSM_L3_MS_v4220/RR/
Purpose: To test that the MS remembers the page mode from the previous paging request message.
Default: OtherEventsFail
Comments: The configuration is Tx-integer = 5, Max-Retrans = 2 and CCCH-CONF, BS-AG-BLKS-RES and BS-PA-MFRMS are from PIXIT paramerters.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_S2, C_S4, C_S6, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(30), INT_TO_BIT(TSPX_AGBLKS3, 3), TSPX_CcchConf3, INT_TO_BIT((TSPX_PAMFRMS3-2), 3), C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	(TCV_Null := OM_2Msgs(TCV_PgCh, TCV_Pgg, C_NxtButOne))			
4		L!DL_UdatRqImmassRej_sp	ImmAssRejSp(TCV_PgCh, TCV_Pgg, ImmAsgnRej_r04)		2.
5		L!DL_UdatRqPg3Rq	PgReq3(TCV_PgCh, TCV_Pgg, PgReqTp3_03)		3.
6		(TCV_Null := OM_2Msgs(TCV_PgCh, TCV_Pgg, C_NxtButOne))			
7		L!DL_UdatRqPg3Rq	PgReq3(TCV_PgCh, TCV_Pgg, PgReqTp3_02)		4.
8		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_07)		5.
9		L?DL_RacInChRq	ChReq(ChRequest_17)		
10	L_0196	L?DL_RacInChRq (TCV_Rqr.ra := DL_RacInChRq. msg.ecau_rrf, TCV_Rqr.fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)	(P)	
11		L!DL_UdatRqImmassRej	ImmAssRej(TCV_agch, ImmAsgnRej_01(TCV_Rqr.ra, TCV_Rqr.fn))		

Detailed Comments:

1. Tx-integer = 5, Max-Retrans = 2 and CCCH-CONF, BS-AG-BLKS-RES and BS-PA-MFRMS are from PIXIT paramerters.
2. extended paging mode.
3. not address the MS.
4. paging mode = "same as before", not address the MS,sent in the next paging subblock on the MS's specific paging subchannel.
5. paging mode = "paging reorgnisation", address the MS, sent in the next but one paging subblock.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_2_5

Group: GSM_L3_MS_v4220/RR/

Purpose: 1) To test that the MS is able to determine its CCCH group and paging group correctly in the case of a CCCH configuration on more than one timeslot when it is paged on a timeslot other than 0. The MS is addressed with a PAGING REQUEST TYPE 1 message when the page mode is set to normal paging. The MS is paged with its IMSI in the 1st Mobile Identity field, the optional Mobile Identity field being not present, is the only way of addressing tested.

2) To test that in such conditions the MS answers to the paging message on the timeslot on which the paging message was sent.

Default: OtherEventsFail

Comments: The configuration is Tx-integer = 5, Max-Retrans = 1 and CCCH-CONF, BS-AG-BLKS-RES and BS-PA-MFRMS are from PIXIT paramerters.

In the PIXITs CCCH-CONF is set to 3, BS-AG-BLKS-RES = 2 and BS-PA-MFRMS = 3.

For this test to be useful, we need

(((IMSI mod 10000) mod (cc * (9 - bsagres) * bspamfrms)) > (9 - bsagres) * bspamfrms)

ie. ((IMSI mod 1000) mod 3*7*9) > 7*9

thus if the last 3 digits of the IMSi are 065 then this test will work.

When the CCCHs that are not on timeslot zero are set, the SYNC channel must not be started on them. The SYNC channel is used by mobiles to identify TS zero. (the FCCH is supposed to be excluded as well)

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_S2, C_S4, C_S6, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), INT_TO_BIT(TSPX_AGBLKS4, 3), TSPX_CcchConf4, INT_TO_BIT((TSPX_PAMFRMS4-2), 3), C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+local_tree			
4	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_05)		2.
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq_01(TCV_RaCh, ChRequest_12)		3.
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		+local_immass			
8	L_0197	L?DL_EstInPgRes	PgRes(TCV_ch, PagingRes_r02)	(P)	
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		+PostMainLinkRel(TCV_ch)			
		local_immass			
11		[(TSPX_CcchConf4 = C_cch_1nonComb) OR (TSPX_CcchConf4 = C_cch_2nonComb) OR (TSPX_CcchConf4 = C_cch_3nonComb) OR (TSPX_CcchConf4 = C_cch_4nonComb)]			
12		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r02(TCV_Rr, TCV_Fn, TSPX_SDCCH8SubF, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), TCV_tch_arfcn))		
13		[TSPX_CcchConf4 = C_cch_1Comb]			
14		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubD		

			ef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
15	L_0198	[(TSPX_CcchConf4 = C_cch_2Comb) OR (TSPX_CcchConf4 = C_cch_3Comb) OR (TSPX_CcchConf4 = C_cch_4Comb)]		I	Reserved
		local_tree			
16		[TSPX_CcchConf4 = C_cch_1Comb]			
17		[(TSPX_CcchConf4 = C_cch_1nonComb) OR (TSPX_CcchConf4 = C_cch_2nonComb) OR (TSPX_CcchConf4 = C_cch_3nonComb) OR (TSPX_CcchConf4 = C_cch_4nonComb)]			
18		+Varinit_tch(C_arfcn_tchA, C_arfcn_tchAd)			
19		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubF, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(TCV_tch_arfcn), FreqTCH(TCV_tch_arfcn), TimingAdv(30), INT_TO_BIT(TSPX_AGBLKS4, 3), TSPX_CcchConf4, INT_TO_BIT((TSPX_PAMFRMS4-2), 3))			

Detailed Comments:

1. Tx-integer = 5, Max-Retrans = 1 CCCH-CONF, BS-AG-BLKS-RES and BS-PA-MFRMS are from PIXIT parameters.
2. The first mobile identifier addresses the MS, the 2nd is omitted.
3. To assign TSPX_SDCCH8SubF subchannel.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_3_1
Group: GSM_L3_MS_v4220/RR/
Purpose: To test that, when the SS gives absolutely no information about neighbouring cells, the MS does not report on neighbouring cells..
Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcA, TSPX_MTChRateA)			
3		+IdleUpdated(C_E_neighbourdefault, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_1, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_20, BcchFreqLst_Omit, BcchFreqLst_50, BcchFreqLst_Omit, C_noRestablishment, C_BCC_3, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcn_2, C_arfcnAd_2, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+StartMultiCells_02(BcchFreqLst_20, BcchFreqLst_50, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_ATT_0, C_BABR_1, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1)			1.
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubA, 0, 3, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			
7	body	(TCV_Null := OM_StartMsrReport(TCV_sacchTch))			
8	L_0199	L?DL_UdatInMsrRpt	MsrRept(MsrReport_01)	(P)	
9		START T_dly1(960)			960 ms
10	L_0200	?TIMEOUT T_dly1		(F)	2.
11		+PostMainLinkRel(TCV_chTch)			
12	L_0201	L?DL_UdatInMsrRpt	MsrRept(MsrReport_01)	(P)	
13		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
14		+PostMainLinkRel(TCV_chTch)			
15		+local_execution2			
		local_execution2			
16		+local_SysInfo5and6Sending			1.
17		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubA, 0, 3, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			
18		(TCV_Null := OM_StartMsrReport(TCV_sacchTch))			
19		L?DL_UdatInMsrRpt	MsrRept(MsrReport_01)		

20		START T_dly1(960)			960 ms
21	L_0202	?TIMEOUT T_dly1		(F)	2.
22		+PostMainLinkRel(TCV_chTch)			
23	L_0203	L?DL_UdatInMsrRpt	MsrRept(MsrReport_01)	(P)	
24		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
25		+PostMainLinkRel(TCV_chTch)			
		local_SysInfo5and6Sending			
26		+Wait(C_T_Wait1)			
27		[TSPC_PGSM OR TSPC_EGSM]			
28		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_24))		
29		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_26))		
30		[TSPC_DCS]			
31		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_25))		
32		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_25))		
33		L!DL_UdatRqSysinfo6 (DL_UdatRqSysinfo6.msg.co.pwrc := '1'B)	SysInfo6(TCV_sacch, SysInf6(C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, C_NCCP_2))		

Detailed Comments:

1. No channels listed in the neighbour cells description.
2. The interval between 2 successive layer 2 frames containing MEASUREMENT REPORT exceeds
   one layer 2 frame, fail.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_3_2
Group: GSM_L3_MS_v4220/RR/
Purpose: To test that, when the SS gives information about neighbouring cells, the MS reports appropriate results.
Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcB, TSPX_MTChRateB)			
3		+IdleUpdated(C_E_neighbourdefault, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_21, BcchFreqLst_Omit, BcchFreqLst_27, BcchFreqLst_Omit, C_noRestablishment, C_BCC_3, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcn_2, C_arfcnAd_2, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+StartMultiCells_02(BcchFreqLst_21, BcchFreqLst_27, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1)			
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6	body	+local_execution(1, 2)			
7		+local_SendSyInfo5and6			
8		+local_execution(1, 3)			
		local_execution(indx1, indx2:INTEGER)			
9		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubA, 0, 3, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			
10		(TCV_Null := OM_StartMsrReport(TCV_sacchTch))			
11		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_03)		
12		+ChkMsrmntRpt(indx1, indx2)			
13		START T_dly1(960)			960 ms
14	L_0204	?TIMEOUT T_dly1		(F)	1.
15		+PostMainLinkRel(TCV_chTch)			
16		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_03)		
17		+ChkMsrmntRpt(indx1, indx2)			
18		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
19		+PostMainLinkRel(TCV_chTch)			
		local_SendSyInfo5and6			
20		+Wait(C_T_Wait1)			
21		[TSPC_PGSM OR TSPC_EGSM]			
22		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_28))		
23		L!DL_UdatRqSysinfo5bis	SysInfo5bis(		

			TCV_sacch, SysInf5bis(BcchFreqLst_30))		
24		[TSPC_DCS]			
25		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_29))		
26		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_31))		
27		L!DL_UdatRqSysinfo6 (DL_UdatRqSysinfo6.msg.co.pwrc := '1'B)	SysInfo6(TCV_sacch, SysInf6(C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, C_NCCP_2))		

Detailed Comments: 1. The interval between 2 successive layer 2 frames containing MEASUREMENT REPORT exceeds
one layer 2 frame, fail.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_3_3
Group: GSM_L3_MS_v4220/RR/
Purpose: To test that, when a combination of normal neighbours, barred cells and non-permitted NCCs is "on air", the MS reports only on normal neighbours.
Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcC, TSPX_MTChRateC)			
3		+IdleUpdated(C_E_neighbourdefault, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_22, BcchFreqLst_Omit, BcchFreqLst_32, BcchFreqLst_Omit, C_noRestablishment, C_BCC_3, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcn_2, C_arfcnAd_2, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+StartMultiCells_02(BcchFreqLst_22, BcchFreqLst_32, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1)			
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubA, 0, 3, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			
7		(TCV_Null := OM_StartMsrReport(TCV_sacchTch))			
8		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_04)		
9		+ChkMsrmntRpt(4, 5)			
10		START T_dly1(960)			960 ms
11	L_0205	?TIMEOUT T_dly1		(F)	1.
12		+PostMainLinkRel(TCV_chTch)			
13		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_04)		
14		+ChkMsrmntRpt(4, 5)			
15		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
16		+PostMainLinkRel(TCV_chTch)			
17		+local_execution2			
		local_execution2			
18		+local_SendSysinfo5and6			
19		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubA, 0, 3, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			

20		(TCV_Null := OM_StartMsrReport(TCV_sacchTch))			
21		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_04)		
22		+ChkMsrmntRpt(4, 5)			
23		START T_dly1(960)			960 ms
24	L_0206	?TIMEOUT T_dly1		(F)	1.
25		+PostMainLinkRel(TCV_chTch)			
26		L?DL_UdatInMsrRpt(TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_04)		
27		+ChkMsrmntRpt(4, 5)			
28		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
29		+PostMainLinkRel(TCV_chTch)			
		local_SendSysinfo5and6			
30		+Wait(C_T_Wait1)			
31		[TSPC_PGSM OR TSPC_EGSM]			
32		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_33))		
33		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_30))		
34		[TSPC_DCS]			
35		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_34))		
36		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_34d))		
37		L!DL_UdatRqSysinfo6 (DL_UdatRqSysinfo6.msg.co.pwrc := '1'B)	SysInfo6(TCV_sacch, SysInf6(C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, C_NCCP_2))		

Detailed Comments: 1. The interval between 2 successive layer 2 frames containing MEASUREMENT REPORT exceeds
one layer 2 frame, fail.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_3_4
Group: GSM_L3_MS_v4220/RR/
Purpose: To test that, in the case of the MS using DTX and the SS indicating that power control is in use, the MS reports appropriate results
Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcJ, TSPX_MTChRateJ)			
3		(TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(73, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellH, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_11, CellChDes_11d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC_1, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnH, C_arfcnHd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			Cell N7
5		+StartMultiCells_03(BcchFreqLst_01, BcchFreqLst_48, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1)			
6		(TCV_chTch1 := C_FACCHF_A_1, TCV_sysinfo6.co.sprb := '1'B, TCV_sysinfo6_B.co.sprb := '1'B)			DTXindicator = '101'B
7		+FullRateCh_A_1(C_Synho, TSPX_TmSltDef, TSPX_TscDef, TCV_ChMod, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
8		+CCConfigTCH_B(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltC, TSPX_TscC, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH(TSPX_TCHcarrierB), FreqTCH(TSPX_TCHcarrierBd), C_Noarfcn, C_Noarfcn)			
9		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubDef, 0, 3, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltC, TSPX_TscC, TSPX_TCHHSubDef, TCV_ChRate)			
10		+local_HndovCmd			
11		+local_RcvHndovAcc			
12		L?DL_EstIn	DLEstInd(TCV_chTch1)		
13		L?DL_DatInHoCom	HndOvCmpRcv(TCV_chTch1, HandOverCmp_01)		
14		+local_continue1			
		local_continue1			
15		(TCV_Null := OM_StartMsrReport(TCV_sacchTch))			
16		[NOT TSPC_TranspDataOnly]			
17	body	L?DL_UdatInMsrRpt (TCV_MsrRes :=	MsrRept(		

		DL_UdatInMsrRpt.msg.msrr)	MsrReport_05)		
18		+ChkMsrmntRpt(6, 7)			
19		START T_dly1(960)			960 ms
20	L_0207	?TIMEOUT T_dly1		(F)	1.
21		+PostMainLinkRel(TCV_chTch)			
22		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_05)		
23		+ChkMsrmntRpt(6, 7)			
24		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
25		+PostMainLinkRel(TCV_chTch)			
26		+local_execution2			
27		[TSPC_TranspDataOnly]			
28		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_06)		
29		+ChkMsrmntRpt(6, 7)			
30		START T_dly1(960)			960 ms
31	L_0208	?TIMEOUT T_dly1		(F)	1.
32		+PostMainLinkRel(TCV_chTch)			
33		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_06)		
34		+ChkMsrmntRpt(6, 7)			
35		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
36		+PostMainLinkRel(TCV_chTch)			
37		+local_execution2			
		local_execution2			
38		+local_SendSysInfo5and6			
39		START T_dly(20000)			2.
40		?TIMEOUT T_dly			
41		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubA, 0, 3, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			
42		+local_HndovCmd			
43		+local_RcvHndovAcc			
44		L?DL_EstIn	DLEstInd(TCV_chTch1)		
45		L?DL_DatInHoCom	HndOvCmpRcv(TCV_chTch1, HandOverCmp_01)		
46		+local_continue2			
		local_continue2			
47		(TCV_Null := OM_StartMsrReport(TCV_sacchTch))			
48		[NOT TSPC_TranspDataOnly]			
49		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_05)		
50		+ChkMsrmntRpt(6, 7)			
51		START T_dly1(960)			960 ms
52	L_0209	?TIMEOUT T_dly1		(F)	1.
53		+PostMainLinkRel(TCV_chTch)			
54		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_05)		
55		+ChkMsrmntRpt(6, 7)			
56		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
57		+PostMainLinkRel(TCV_chTch)			
58		+local_execution2			
59		[TSPC_TranspDataOnly]			
60		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_06)		
61		+ChkMsrmntRpt(6, 7)			

62		START T_dly1(960)			960 ms
63	L_0210	?TIMEOUT T_dly1		(F)	1.
64		+PostMainLinkRel(TCV_chTch)			
65		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_06)		
66		+ChkMsrmntRpt(6, 7)			
67		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
68		+PostMainLinkRel(TCV_chTch)			
		local_RcvHndovAcc			
69		L?DL_RacInHoacc	HndOvAccRcv(TCV_chTch, HandOverAcc_01)		
70		L?DL_RacInHoacc	HndOvAccRcv(TCV_chTch, HandOverAcc_01)		
71		L?DL_RacInHoacc	HndOvAccRcv(TCV_chTch, HandOverAcc_01)		
72		L?DL_RacInHoacc	HndOvAccRcv(TCV_chTch, HandOverAcc_01)		
		local_HndovCmd			
73		(TCV_Tchtype := '00001'B)			
74		[TSPC_PGSM OR TSPC_EGSM]			
75		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC_3, C_arfcn_2, C_arfcn_tchA, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
76		[TSPC_DCS]			
77		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC_3, C_arfcnAd_2, C_arfcn_tchAd, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
		local_SendSysInfo5and6			
78		[TSPC_PGSM OR TSPC_EGSM]			
79		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_36))		SysInfo 1, 2, 3, 4, 6 as before
80		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_26))		

81		[TSPC_DCS]			
82		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_29))		SysInfo 1, 2, 3, 4, 6 as before
83		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_51))		

Detailed Comments:

1. The interval between 2 successive layer 2 frames containing MEASUREMENT REPORT exceeds
   one layer 2 frame, fail.
2. To allow the MS camp in cell H again.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_3_5
Group: GSM_L3_MS_v4220/RR/
Purpose: To test that when the SS gives information about neighbouring cells the MS reports appreprate results.
Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcH, TSPX_MTChRateH)			
3		+IdleUpdated(C_E_neighbourdefault, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_37, BcchFreqLst_Omit, BcchFreqLst_38, BcchFreqLst_Omit, C_noRestablishment, C_BCC_3, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcn_2, C_arfcnAd_3, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+StartCellB(28, 8, 815, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_05, CellChDes_05d, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_5, C_NCC)			
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+local_SysInfo5bisSending			
7	body	+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubA, 0, 3, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			
8		(TCV_Null := OM_StartMsrReport(TCV_sacchTch))			
9		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_07)		
10		+ChkMsrmntRpt(8, 9)			
11		+local_continue			
		local_continue			
12		START T_dly1(960)			960 ms
13	L_0211	?TIMEOUT T_dly1		(F)	1.
14		+PostMainLinkRel(TCV_chTch)			
15		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_07)		
16		+ChkMsrmntRpt(8, 9)			
17		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
18		+PostMainLinkRel(TCV_chTch)			
19		+local_execution2			
		local_execution2			
20		+local_SysInfo5and6Sending(BcchFreqLst_41, BcchFreqLst_39, BcchFreqLst_42)			1.
21		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubA, 0, 3, TSPX_CKSNDef,			

		TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			
22		(TCV_Null := OM_StartMsrReport(TCV_sacchTch))			
23		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_07)		
24		+ChkMsrmntRpt(8, 9)			
25		START T_dly1(960)			960 ms
26	L_0212	?TIMEOUT T_dly1		(F)	1.
27		+PostMainLinkRel(TCV_chTch)			
28		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_07)		
29		+ChkMsrmntRpt(8, 9)			
30		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
31		+PostMainLinkRel(TCV_chTch)			
32		+local_execution3			
		local_execution3			
33		+local_SysInfo5and6Sending(BcchFreqLst_43, BcchFreqLst_40, BcchFreqLst_44)			1.
34		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubA, 0, 3, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			
35		(TCV_Null := OM_StartMsrReport(TCV_sacchTch))			
36		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_07)		
37		+ChkMsrmntRpt(8, 10)			
38		START T_dly1(960)			960 ms
39	L_0213	?TIMEOUT T_dly1		(F)	1.
40		+PostMainLinkRel(TCV_chTch)			
41		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_07)		
42		+ChkMsrmntRpt(8, 10)			
43		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
44		+PostMainLinkRel(TCV_chTch)			
		local_SysInfo5and6Sending(ba1, ba2, ba3 :NCD)			
45		+Wait(C_T_Wait1)			
46		[TSPC_PGSM OR TSPC_EGSM]			
47		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(ba1))		
48		[TSPC_DCS]			
49		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(ba2))		
50		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(ba3))		
51		L!DL_UdatRqSysinfo6	SysInfo6(TCV_sacch, SysInf6(C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_06, C_NCCP_2))		
		local_SysInfo5bisSending			
52		[TSPC_PGSM OR TSPC_EGSM]			
53		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_38))		

54		[TSPC_DCS]			
55		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_37))		

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_4_1

Group: GSM_L3_MS_v4220/RR/

Purpose: 1. To verify that upon receipt of an ASSIGNMENT COMMAND, the MS switches to the channel defined in the ASSIGNMENT COMMAND, establishes the link and sends an ASSIGNMENT COMPLETE message. This is tested for an MS supporting TCH in the special cases of a transition

1.1 from non-hopping SDCCH to hopping TCH/F using a different timeslot

1.2 from hopping TCH/F to non-hopping TCH/F using a different timeslot

1.3 from non-hopping TCH/F to non-hopping TCH/F using a different timeslot

1.4 from non-hopping TCH/F to hopping TCH/H using a different timeslot; this test purpose is only applicable if the MS supports TCH/H

1.5 from hopping TCH/H to non-hopping TCH/H using a different timeslot; this test purpose is only applicable if the MS supports TCH/H

1.6 from non-hopping TCH/H to hopping TCH/F using a different timeslot; this test purpose is only applicable if the MS supports TCH/H.

2. To verify that an MS supporting TCH, having sent an MM- or CM message which was not acknowledged on L2 before the channel assignment procedure was initiated and before the MS has left the old channel, repeats that message after completion of the assignment procedure without incrementing N(SD). This is tested in the special case of MM message AUTHENTICATION RESPONSE.

3. To verify that, if an MS supporting TCH has received an ASSIGNMENT COMMAND message which contains only the description of a channel to be used after the starting time, and if the starting time has not already elapsed, the mobile station shall wait up to the starting time before accessing the channel.

4. To verify that an MS supporting TCH, having received an ASSIGNMENT COMMAND, having sent an SABM frame to establish the main signalling link on the assigned channel, reports the power level specified in the ASSIGNMENT COMMAND message, in the uplink SACCH L1 header of the SACCH message sent in the SACCH period following the transmission of the SABM frame.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_05, CellChDes_15, CellChDes_15d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_02, BcchFreqLst_01, BcchFreqLst_04, BcchFreqLst_48, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqBCCH(C_arfcnA), FreqBCCH(C_arfcnAd_4), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			2.
4		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
6		+ltree_ImmAss			
7		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
8		+localtree1			
9		+localtree2			
10		[NOT TSPC_DualRate]			
11		+PostMainLinkRel(TCV_chTch1)			
12		[TSPC_DualRate]			
13		+gofurther			

		localtree1			
14		+ltree_Asgn1			
15		+ltree_AssCh_complete1(TCV_ch, TCV_chTch, TCV_AssCmd)			
16		+ltree_Asgn2			
17		+ltree_AssCh_complete(TCV_chTch, TCV_chTch1, TCV_AssCmd)			
		localtree2			
18		(TCV_Null :=OM_NoL2Ack(1, TCV_chTch1))			
19		L!DL_DatRqAuthRq	AuthReq(TCV_chTch1, AuthRequest(TSPX_CKSNDef, TSPX_RANDDef))		
20		L?DL_DatInAuthRes (TCV_Mt := DL_DatInAuthRes.msg.mt)	AuthRes(AuthResponse)		
21		+ltree_Asgn3			
22		+ltree_AssCh_complete(TCV_chTch1, TCV_chTch, TCV_AssCmd)			
23		L?DL_DatInAuthRes (TCV_Mt1 := DL_DatInAuthRes.msg.mt)	AuthRes(AuthResponse)		
24		+SendSeqNo_chk			
25		+ltree_Asgn4			
26		+ltree_AssCh_complete4(TCV_chTch, TCV_chTch1, TCV_AssCmd)			
		gofurther			
27		+ltree_Asgn5			
28		+ltree_AssCh_complete(TCV_chTch1, TCV_chTch, TCV_AssCmd)			
29		L!DL_DatRqIdRq	IDReq(TCV_chTch, IDRequest_01('0010'B))		
30		L?DL_DatInIdRes	IDRes(IDResponse_01)		
31		+ltree_Asgn6			
32		+ltree_AssCh_complete(TCV_chTch, TCV_chTch1, TCV_AssCmd)			
33		+ltree_Asgn7			
34		+ltree_AssCh_complete(TCV_chTch1, TCV_chTch, TCV_AssCmd)			
35		+PostMainLinkRel(TCV_chTch)			
		ltree_AssCh_complete(oldch,newch:LOGICCH; pdu_ass: ASS_CMD_PDU)			
36		L!DL_DatRqAssCmd START T_dlyAss	AssCmd(oldch,pdu_ass)		
37		?TIMEOUT T_dlyAss		(F)	
38		+PostLinkRelEnd(oldch)			
39		L?DL_EstIn CANCEL T_dlyAss	DLEstInd(newch)		
40		(TCV_Fn := OM_ReturnFn(oldch))			
41		L?DL_DatInAssCom (TCV_Fn1 := DL_DatInAssCom.fn)	AssCmp(newch, AsgnCmp_02)		
42		L!MDL_RelRq	MDLRelReq(oldch)		
43		+Check_Time(C_T_AssCmd)			
		ltree_AssCh_complete1(oldch,newch:LOGICCH; pdu_ass: ASS_CMD_PDU)			
44		L!DL_DatRqAssCmd START T_dlyAss	AssCmd(oldch,pdu_ass)		
45		?TIMEOUT T_dlyAss		(F)	
46		+PostLinkRelEnd(oldch)			
47		L?DL_EstIn CANCEL T_dlyAss	DLEstInd(newch)		
48		(TCV_Fn:=OM_ReturnFn(oldch), TCV_L1Head:=OM_GetL1Hd(TCV_sacchTch))			
49		L?DL_DatInAssCom (TCV_Fn1 := DL_DatInAssCom.fn)	AssCmp(newch, AsgnCmp_02)		

50		L!MDL_RelRq	MDLRelReq(oldch)		
51		+Check_Time(C_T_AssCmd)			
52	L_0214	[TCV_L1Head.mspwrlvl<>'00111'B]		(F)	
53	L_0215	[TCV_L1Head.mspwrlvl='00111'B]		(P)	
		ltree_AssCh_complete4(oldch,newch:LOGICCH; pdu_ass: ASS_CMD_PDU)			
54		L!DL_DatRqAssCmd START T_dlyAss	AssCmd(oldch,pdu_ass)		
55		?TIMEOUT T_dlyAss		(F)	
56		+PostLinkRelEnd(oldch)			
57		L?DL_EstIn CANCEL T_dlyAss	DLEstInd(newch)		
58		L?DL_DatInAssCom (TCV_Fn1 := DL_DatInAssCom.fn)	AssCmp(newch, AsgnCmp_02)		
59		L!MDL_RelRq	MDLRelReq(oldch)		
60		+FnArith(TCV_Fn,TCV_Fn1)			
61		[TCV_K>17]		(F)	
62		[(TCV_K>=0) AND (TCV_K<=17)]		(P)	
		ltree_ImmAss			
63		[TSPC_PGSM OR TSPC_EGSM]			
64		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_sdcch8(TCV_Rr, TCV_Fn, TSPX_TmSltDef, TSPX_TscDef, TSPX_SDCCH8SubDef, C_arfcnA, TimingAdv(30)))		
65		[TSPC_DCS]			
66		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_sdcch8(TCV_Rr, TCV_Fn, TSPX_TmSltDef, TSPX_TscDef, TSPX_SDCCH8SubDef, C_arfcnAd_4, TimingAdv(30)))		
		ltree_Asgn1			
67		(TCV_chTch := C_FACCHF_A_1, TCV_sacchTch := C_SACCHF_A_1, TCV_slt2 := INT_TO_BIT(((BIT_TO_INT(TSPX_TmSltDef) + 1) MOD 8), 3))			
68		[TSPC_PGSM OR TSPC_EGSM]			
69		+Config_FACCHF_A_1(C_E_default, 7, ChMod_sign, C_Ass, TCV_slt2, TSPX_TscDef, FreqTCHa3, C_TCHF_ACCHF_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
70		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
71		(TCV_AssCmd := AsgnCmd_tchf_fh(TCV_slt2, TSPX_TscDef, 7, '000001'B, '000001'B, Frql_omit, CellChDes_Omit, ChMod_sign_iei, MoblAllc_r01, CphMod_omit))			
72		[TSPC_DCS]			
73		+Config_FACCHF_A_1(C_E_default, 7, ChMod_sign, C_Ass, TCV_slt2, TSPX_TscDef, FreqTCHa8, C_TCHF_ACCHF_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
74		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
75		(TCV_AssCmd := AsgnCmd_tchf_fh(TCV_slt2, TSPX_TscDef, 7, '000001'B, '000001'B, Frql_omit, CellChDes_Omit, ChMod_sign_iei, MoblAllc_r01, CphMod_omit))			
		ltree_Asgn2			

(TCV_chTch1 := C_FACCHF_A_2, TCV_sacchTch1 := C_SACCHF_A_2, TCV_slt2 := INT_TO_BIT(((BIT_TO_INT(TSPX_TmSltDef) + 3) MOD 8), 3))
  [TSPC_PGSM OR TSPC_EGSM]
    +Config_FACCHF_A_2(C_E_default, 16, ChMod(TSPX_modF), C_Ass, TCV_slt2, TSPX_TscDef, FreqTCH(C_arfcnA), C_TCHF_ACCHF_2, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)
      +SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)
        (TCV_AssCmd := AsgnCmd_nfh('00001'B, TCV_slt2, TSPX_TscDef, 16, C_arfcnA, CellChDes_r03, ChMod_iei(TSPX_modF), StartingTm_omit, CphMod_omit))
  [TSPC_DCS]
    +Config_FACCHF_A_2(C_E_default, 12, ChMod(TSPX_modF), C_Ass, TCV_slt2, TSPX_TscDef, FreqTCH(C_arfcnAd_4), C_TCHF_ACCHF_2, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)
      +SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)
        (TCV_AssCmd := AsgnCmd_nfh('00001'B, TCV_slt2, TSPX_TscDef, 12, C_arfcnAd_4, CellChDes_r01, ChMod_iei(TSPX_modF), StartingTm_omit, CphMod_omit))

ltree_Asgn3

(TCV_chTch := C_FACCHF_A_1, TCV_sacchTch := C_SACCHF_A_1, TCV_slt2 := INT_TO_BIT(((BIT_TO_INT(TSPX_TmSltDef) + 4) MOD 8), 3))
  [TSPC_PGSM OR TSPC_EGSM]
    +Config_FACCHF_A_1(C_E_default, 9, ChMod(TSPX_modF), C_Ass, TCV_slt2, TSPX_TscDef, FreqTCHa7, C_TCHF_ACCHF_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)
      +SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)
        (TCV_AssCmd := AsgnCmd_tchf_fh(TCV_slt2, TSPX_TscDef, 9, '000011'B, '001000'B, Frql_omit, CellChDes_Omit, ChMod_omit, MoblAllc_r02, CphMod_omit))
  [TSPC_DCS]
    +Config_FACCHF_A_1(C_E_default, 9, ChMod_speech, C_Ass, TCV_slt2, TSPX_TscDef, FreqTCHa6, C_TCHF_ACCHF_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)
      +SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)
        (TCV_AssCmd := AsgnCmd_tchf_fh(TCV_slt2, TSPX_TscDef, 9, '000011'B, '001000'B, Frql_omit, CellChDes_Omit, ChMod_omit, MoblAllc_r04, CphMod_omit))

ltree_Asgn4

(TCV_chTch1 := C_FACCHF_A_2, TCV_sacchTch1 := C_SACCHF_A_2, TCV_slt2 := INT_TO_BIT(((BIT_TO_INT(TSPX_TmSltDef) + 5) MOD 8), 3), TCV_Fn := OM_ComingFn(TCV_chTch), TCV_Fn1 := OC_FnInc(TCV_Fn, 100), TCV_Null := OM_SendNextOn(TCV_chTch, TCV_Fn), TCV_Fn:=TCV_Fn1)
  [TSPC_PGSM OR TSPC_EGSM]
    +Config_FACCHF_A_2(C_E_default, 14, ChMod_sign, C_Ass, TCV_slt2, TSPX_TscDef, FreqTCH(C_arfcn_10), C_TCHF_ACCHF_2, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)
      +SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)

98		(TCV_AssCmd := AsgnCmd_nfh('00001'B, TCV_slt2, TSPX_TscDef, 14, C_arfcn_10, CellChDes_Omit, ChMod_sign_iei, StartingTm_01(TCV_Fn1), CphMod_omit))				
99		[TSPC_DCS]				
100		+Config_FACCHF_A_2(C_E_default, 14, ChMod_sign, C_Ass, TCV_slt2, TSPX_TscDef, FreqTCH(C_arfcnAd_5), C_TCHF_ACCHF_2, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)				
101		+SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)				
102		(TCV_AssCmd := AsgnCmd_nfh('00001'B, TCV_slt2, TSPX_TscDef, 14, C_arfcnAd_5, CellChDes_Omit, ChMod_sign_iei, StartingTm_01(TCV_Fn1), CphMod_omit))				
		ltree_Asgn5				
103		(TCV_chTch := OC_SubchOfFacchh(TSPX_TCHHSubDef, C_CellA, 1), TCV_sacchTch := OC_SubchOfSacchh(TSPX_TCHHSubDef, C_CellA, 1), TCV_slt2 := INT_TO_BIT(((BIT_TO_INT(TSPX_TmSltDef) +6) MOD 8), 3))				
104		[TSPC_PGSM OR TSPC_EGSM]				
105		+Config_FACCHH_A_1(C_E_default, 8, ChMod(TSPX_modH), C_Ass, TCV_slt2, TSPX_TscDef, FreqTCHa9, C_TCHH_ACCHH_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)				
106		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)				
107		(TCV_AssCmd := AsgnCmd_tchh_fh(TSPX_TCHHSubDef, TCV_slt2, TSPX_TscDef, 8, '000101'B, '000000'B, Frql_08, CellChDes_Omit, ChMod_iei(TSPX_modH), MoblAllc_omit, CphMod_omit))				
108		[TSPC_DCS]				
109		+Config_FACCHH_A_1(C_E_default, 3, ChMod(TSPX_modH), C_Ass, TCV_slt2, TSPX_TscDef, FreqTCHa5, C_TCHH_ACCHH_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)				
110		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)				
111		(TCV_AssCmd := AsgnCmd_tchh_fh(TSPX_TCHHSubDef, TCV_slt2, TSPX_TscDef, 3, '000101'B, '000000'B,Frql_09, CellChDes_Omit, ChMod_iei(TSPX_modH), MoblAllc_omit, CphMod_omit))				
		ltree_Asgn6				
112		(TCV_chTch1 := OC_SubchOfFacchh(TSPX_TCHHSubA, C_CellA, 2), TCV_sacchTch1 := OC_SubchOfSacchh(TSPX_TCHHSubDef, C_CellA, 2), TCV_slt2 := INT_TO_BIT(((BIT_TO_INT(TSPX_TmSltDef) +7) MOD 8), 3))				
113		[TSPC_PGSM OR TSPC_EGSM]				
114		+Config_FACCHH_A_2(C_E_default, 12, ChMod(TSPX_modH), C_Ass, TCV_slt2, TSPX_TscDef, FreqTCH(C_arfcn_34), C_TCHH_ACCHH_2, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)				
115		+SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)				
116		(TCV_AssCmd := AsgnCmd_tchh_nfh(TSPX_TCHHSubA, TCV_slt2, TSPX_TscDef, 12, C_arfcn_34, CellChDes_Omit, ChMod_omit, StartingTm_omit))				
117		[TSPC_DCS]				
118		+Config_FACCHH_A_2(C_E_default, 9, ChMod(TSPX_modH), C_Ass, TCV_slt2,				

		TSPX_TscDef, FreqTCH(C_arfcnAd_6), C_TCHH_ACCHH_2, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
119		+SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)			
120		(TCV_AssCmd := AsgnCmd_tchh_nfh(TSPX_TCHHSubA, TCV_slt2, TSPX_TscDef, 9, C_arfcnAd_6, CellChDes_Omit, ChMod_omit, StartingTm_omit))			
		ltree_Asgn7			
121		(TCV_slt2 := INT_TO_BIT(((BIT_TO_INT(TSPX_TmSltDef) + 1) MOD 8), 3))			
122		+FullRateCh_A_1_nociph(C_Ass, TCV_slt2, TSPX_TscDef, ChMod(TSPX_modH), FreqTCHa10, FreqTCHa4, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
123		[TSPC_PGSM OR TSPC_EGSM]			
124		(TCV_AssCmd := AsgnCmd_tchf_fh(TCV_slt2, TSPX_TscDef, 19, '000000'B, '101000'B, Frql_omit, CellChDes_r04, ChMod_omit, MoblAllc_r03, CphMod_omit))			
125		[TSPC_DCS]			
126		(TCV_AssCmd := AsgnCmd_tchf_fh(TCV_slt2, TSPX_TscDef, 15, '000000'B, '101000'B, Frql_omit, CellChDes_r02, ChMod_omit, MoblAllc_r03, CphMod_omit))			

Detailed Comments: 1. Default parameters exept CA.
2. The time slot is TSPX_TmSltDef, the ARFCN is the BCCH carrier.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_4_2_1

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that, when the MS fails to seize the new channel, the MS reactivates the old channel, reporting use of the last power level used on the old channel. This is tested in the special cases of a transition:
- from TCH/F to hopping TCH/F in state U10 if the MS supports TCH/F and call control
- from TCH/H to hopping TCH/H in state U10 if the MS supports TCH/H and call control

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcB, C_Full)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
5		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubA, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubA, C_Full)			1.
6	body	+localtree			
		localtree			
7		+AssCmdGen(TCV_cellid, C_Full, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
8		+AssCh_failure(TCV_chTch, TCV_AssCmd, FALSE)			
9		+localtree3			
10		+localtree1			
		localtree1			
11		[TSPC_DualRate]			
12		+Wait(C_T_Wait1)			
13		(TCV_CphKey := OC_CphKeyGen(TSPX_Ki, TSPX_RANDDef))			
14		+HalfRateCh_A_1(TSPX_TCHHSubA, C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
15		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubA, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubA, C_Half)			3.
16		+localtree2			
17		[TCV_CC AND(NOT TSPC_DualRate)]			
		localtree2			
18		+AssCmdGen(TCV_cellid, C_Half, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
19		+AssCh_failure(TCV_chTch, TCV_AssCmd, FALSE)			

20		+localtree3			
		localtree3			
21		(TCV_L1Head := OM_GetL1Hd(C_SACCHF_A_1))			
22		[TSPC_PGSM OR TSPC_EGSM]			
23	L_0221	[TCV_L1Head.mspwrlvl <> '00111'B]		(F)	6.
24		+PostMainLinkRel(TCV_chTch)			
25	L_0222	[TCV_L1Head.mspwrlvl = '00111'B]		(P)	
26		+PostMainLinkRel(TCV_chTch)			
27		[TSPC_DCS]			
28	L_0223	[TCV_L1Head.mspwrlvl <> '00011'B]		(F)	6.
29		+PostMainLinkRel(TCV_chTch)			
30	L_0224	[TCV_L1Head.mspwrlvl = '00011'B]		(P)	
31		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To set up a full rate non hopping call and power level = 7.
2. To assign a full rate hopping channel with power level = 9, but the channel is not activated.
3. To set up a half rate non hopping call and power level = 7.
4. To assign a half rate channel with power level = 9, but the channel is not activated.
5. The expected ASSIGNMENT FAILURE with " protocol error unspecified" received on the old channel.
6. The power level is not the old power level, fail.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_4_2_2

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that, when the MS fails to seize the new channel, the MS reactivates the old channel. This is tested in the special cases of a transition:

- from SDCCH to hopping TCH/F; this test part is only applicable if the MS supports TCH/F.
- from non-hopping SDCCH to hopping TCH/H; this test part is only applicable if the MS supports TCH/H.
- from hopping TCH/F to hopping TCH/H; this test part is only applicable if the MS supports TCH/H.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubB, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
4		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
5		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubB, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
6		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
7		+localtree			
		localtree			
8		(TCV_AssCmd := AsgnCmd_tchf_fh(C_S3, C_BCC, 7, '000000'B, '000100'B, Frql_omit, CellChDes_Omit, ChMod_omit, MoblAllc_04, CphMod_omit))			
9		+Adjust_gsmanddcs_powerlvl(0, 3, TCV_AssCmd)			
10		+AssCh_failure(TCV_ch, TCV_AssCmd, FALSE)			
11		[TCV_CC AND(NOT TSPC_DualRate)]			
12		+PostMainLinkRel(TCV_ch)			
13		[TSPC_DualRate]			
14		+localtree1			
15		[C_Yes]			
		localtree1			
16		(TCV_AssCmd := AsgnCmd_tchh_fh('0'B, '001'B, TSPX_TscDef, 7, '000001'B, '000110'B, Frql_omit, CellChDes_Omit, ChMod_omit, MoblAllc_04, CphMod_omit))			
17		+Adjust_gsmanddcs_powerlvl(0, 3, TCV_AssCmd)			
18		+AssCh_failure(TCV_ch, TCV_AssCmd, FALSE)			
19		+FullRateCh_A_1_nociph(C_Ass, C_S3, C_BCC, ChMod_sign, Freq_rg8, Freq_rd8, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			4.
20		(TCV_AssCmd := AsgnCmd_tchf_fh(C_S3, C_BCC, 7, '000000'B, '000100'B, Frql_omit, CellChDes_Omit, ChMod_omit, MoblAllc_04, CphMod_omit))			

21			+Adjust_gsmanddcs_powerlvl(0, 3, TCV_AssCmd)			
22			+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
23			(TCV_AssCmd := AsgnCmd_tchh_fh('0'B, '001'B, TSPX_TscDef, 7, '000001'B, '000110'B, Frql_omit, CellChDes_Omit, ChMod_omit, MoblAllc_04, CphMod_omit))			
24			+Adjust_gsmanddcs_powerlvl(0, 3, TCV_AssCmd)			
25			+AssCh_failure(TCV_chTch, TCV_AssCmd, FALSE)			
26			+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. Default parameters: CCCH combined with SDCCH4, Tx-integer = 5, Max-retrans = 1.
2. To assign a TCH/F hopping channel which is not activated.
3. To assign a TCH/H hopping channel which is not activated.
4. To setup a physical channel as TCH/F hopping channel.
5. To assign the TCH/F hopping channel which is now activated.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_1

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a non-synchronised handover it continuously sends access bursts on the main DCCH until it receives a PHYSICAL INFORMATION message from the SS. To test that the MS correctly handles the values of any Starting Time IE in the HANDOVER COMMAND message and the Timing Advance IE in the PHYSICAL INFORMATION message. To test that the MS activates the new channel correctly.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+BasicServiceMT(TSPX_MTBscSvcl, TSPX_MTChRatel)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(20), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_201_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(20), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell B is 100 in advance of frame number in cell A.
6		+FullRateCh_A_1(C_Ass, TCV_asscmd_ts, TCV_tsc, TCV_ChMod, FreqTCH_omit, FreqTCH_omit, TSPX_TCHcarrierA_ho, TSPX_TCHcarrierA_hod, TimingAdv(20), C_BABR_0, C_cch_1Comb, C_BPM_3)			
7		+Est_MT_CallNonFH(TimingAdv(20), C_Full, TSPX_TCHHSubDef, TSPX_RANDDef)			0.
8		(TCV_M := 1, TCV_ch := TCV_chTch)			
9	body	REPEAT local_handover UNTIL [TCV_M = 9]			
10		+ChanRel_end(TCV_ch)			
		local_handover			
11		(TCV_chTch1 := TCV_ch)			
12		+local_channel_config			
13		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
14		+local_send_hoCMD			
15		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		1.
16		+RR_hocomp1(TCV_n, TCV_chTch1)			
17		+local_firstburst_ontime_CHK			
18		[(TCV_M = 3) AND(NOT TSPC_DualRate)]			
19		(TCV_M := 9)			
20		[C_Yes]			
21		(TCV_M := TCV_M + 1)			

		local_send_hoCMD			
22		[(TCV_M = 3) OR (TCV_M = 5)]			
23		(TCV_Fn := OM_ComingFn(TCV_chTch1), TCV_Null:= OM_SendNextOn(TCV_chTch1, TCV_Fn), TCV_Strt := OC_StartTime(TCV_Fn,(C_StartingTimeHO + 100), 1))			starting time is set to 1.1 s pluse 100 frames
24		L!DL_DatRqHoCmd(DL_DatRqHoCmd.msg.strt:= TCV_Strt)	HndOvSnd(TCV_chTch1, TCV_HoCMD)		
25		[C_Yes]			
26		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch1, TCV_HoCMD)		
		local_channel_config			
27		+local_ch_config			
28		(TCV_ch := TCV_chTch)			
		local_ch_config			
29		[TCV_M = 1]			2.
30		(TCV_n := 500, TCV_Cntref := TSPX_NoOfHoAccA, TCV_T := 20, TCV_Horf:= TSPX_HoRefA, TCV_ts := TSPX_TmSltB, TCV_tsc := TSPX_TscB, TCV_chtype := '00001'B)			
31		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, TCV_ChMod, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			
32		+local_set_arfcn_B			
33		(TCV_HoCMD := HandOverCmd_nfh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
34		[TCV_M = 2]			3.
35		(TCV_n := 500, TCV_Cntref := TSPX_NoOfHoAccB, TCV_T := TSPX_TimadvA, TCV_ts := TSPX_TmSltNotZero, TCV_Horf:= TSPX_HoRefB, TCV_tsc := TSPX_TscC, TCV_chtype := '00001'B)			
36		+FullRateCh_A_1(C_Asynho, TCV_ts, TCV_tsc, TCV_ChMod, FreqTCHa_hof1, FreqTCHa_hof1d, C_Noarfcn, C_Noarfcn, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			
37		+local_handover2			
38		[TCV_M = 3]			4.
39		(TCV_n := 500, TCV_Cntref := TSPX_NoOfHoAccC, TCV_T := 20, TCV_ts := TSPX_TmSltC, TCV_Horf:= TSPX_HoRefC, TCV_tsc := TSPX_TscC, TCV_chtype := '00001'B)			
40		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, TCV_ChMod, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			
41		+local_set_arfcn_B			
42		(TCV_HoCMD := HandOverCmd_nfh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), TCV_ChMod, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
43		[TCV_M = 4]			5.
44		(TCV_n := 750, TCV_Cntref := TSPX_NoOfHoAccD, TCV_T := TSPX_TimadvB, TCV_ts := C_S0, TCV_Horf:= TSPX_HoRefD, TCV_tsc := TSPX_TscD, TCV_chtype := INT_TO_BIT((2 + BIT_TO_INT(TSPX_TCHHSubA)), 5))			
45		+HalfRateCh_A_1(TSPX_TCHHSubA,			

		C_Asynho, TCV_ts, TCV_tsc, TCV_ChMod, FreqTCHa_hof4, FreqTCHa_hof4d, C_Noarfcn, C_Noarfcn, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			
46		+local_handover4			
47		[TCV_M = 5]			6.
48		(TCV_n := 750, TCV_Cntref := TSPX_NoOfHoAccE, TCV_T := 20, TCV_ts := TSPX_TmSltNotZero1, TCV_Horf:= TSPX_HoRefE, TCV_tsc := TSPX_TscE, TCV_chtype := INT_TO_BIT((2 + BIT_TO_INT(TSPX_TCHHSubDef)), 5))			
49		+HalfRateCh_B_1(TSPX_TCHHSubDef, C_Asynho, TCV_ts, TCV_tsc, TCV_ChMod, FreqTCHb_hof1, FreqTCHb_hof1d, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			
50		+local_handover5			
51		[TCV_M = 6]			7.
52		(TCV_n := 750, TCV_Cntref := TSPX_NoOfHoAccF, TCV_T := 20, TCV_ts := TSPX_TmSltNotZero, TCV_Horf:= TSPX_HoRefF, TCV_tsc := TSPX_TscF, TCV_chtype := INT_TO_BIT((2 + BIT_TO_INT(TSPX_TCHHSubDef)), 5))			
53		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Asynho, TCV_ts, TCV_tsc, TCV_ChMod, FreqTCH_omit, FreqTCH_omit, TSPX_TCHcarrierA_ho, TSPX_TCHcarrierA_hod, TimingAdv(TSPX_TimadvB), C_BABR_0, C_cch_1Comb, C_BPM_3)			
54		+local_set_arfcn_A			
55		+local_HoCMD_nofh			
56		[TCV_M = 7]			8.
57		(TCV_n := 750, TCV_Cntref := TSPX_NoOfHoAccD, TCV_T := TSPX_TimadvA, TCV_ts := TSPX_TmSltNotZero, TCV_Horf:= TSPX_HoRefG, TCV_tsc := TSPX_TscG, TCV_chtype := INT_TO_BIT((2 + BIT_TO_INT(TSPX_TCHHSubA)), 5))			
58		+HalfRateCh_B_1(TSPX_TCHHSubA, C_Asynho, TCV_ts, TCV_tsc, TCV_ChMod, FreqTCHb_hof2, FreqTCHb_hof2d, TimingAdv(TSPX_TimadvA), C_BABR_0, C_cch_1Comb, C_BPM_3)			
59		+local_handover7			
60		[TCV_M = 8]			9.
61		(TCV_n := 500, TCV_Cntref := TSPX_NoOfHoAccA, TCV_T := TSPX_TimadvB, TCV_ts := TSPX_TmSltNotZero1, TCV_Horf:= TSPX_HoRefB, TCV_tsc := TSPX_TscB, TCV_chtype := '00001'B)			
62		+FullRateCh_A_1(C_Asynho, TCV_ts, TCV_tsc, TCV_ChMod, FreqTCH_omit, FreqTCH_omit, C_arfcnA, C_arfcnAd_4, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			
63		+local_set_arfcn_A1			
64		+local_HoCMD_nofh			
		local_set_arfcn_A			
65		[TSPC_PGSM OR TSPC_EGSM]			
66		(TCV_chdescr_arfcn := C_arfcnA, TCV_tch_arfcn := TSPX_TCHcarrierA_ho)			
67		[TSPC_DCS]			
68		(TCV_chdescr_arfcn := C_arfcnAd_4, TCV_tch_arfcn := TSPX_TCHcarrierA_hod)			
		local_set_arfcn_A1			
69		[TSPC_PGSM OR TSPC_EGSM]			
70		(TCV_chdescr_arfcn := C_arfcnA, TCV_tch_arfcn := C_arfcnA)			
71		[TSPC_DCS]			
72		(TCV_chdescr_arfcn := C_arfcnAd_4,			

		TCV_tch_arfcn := C_arfcnAd_4)			
		local_set_arfcn_B			
73		[TSPC_PGSM OR TSPC_EGSM]			
74		(TCV_chdescr_arfcn := C_BCCHcarrierB_ho, TCV_tch_arfcn := TSPX_TCHcarrierB_ho)			
75		[TSPC_DCS]			
76		(TCV_chdescr_arfcn := C_BCCHcarrierB_hod, TCV_tch_arfcn := TSPX_TCHcarrierB_hod)			
		local_firstburst_ontime_CHK			
77		[(TCV_M = 3) OR (TCV_M = 5)]			
78		+FnArith(TCV_FnBurst, TCV_Strt.fn)			
79		[TCV_M = 3]			
80	L_0225	[(TCV_Time > 5) OR(TCV_Time < 0)]		(F)	10.
81	L_0226	[(TCV_Time >= 0) AND (TCV_Time <= 5)]		(P)	
82		[TCV_M = 5]			
83	L_0227	[(TCV_Time > 10) OR(TCV_Time < 0)]		(F)	10.
84	L_0228	[(TCV_Time >= 0) AND (TCV_Time <= 10)]		(P)	
85		[C_Yes]			
		local_handover2			
86		[TSPC_PGSM OR TSPC_EGSM]			
87		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnA, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_22, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
88		[TSPC_DCS]			
89		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnAd_4, 9, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_20_Ad, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
		local_handover4			
90		[TSPC_PGSM OR TSPC_EGSM]			
91		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnA, 16, C_HSN_0, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_20_A, ChMod_omit, Freqchseq_omit, MoblAllc_20_A1, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
92		[TSPC_DCS]			
93		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnA, 16, C_HSN_0, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_202_Ad, ChMod_omit, Freqchseq_omit, MoblAllc_20_A1, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
		local_handover5			
94		[TSPC_PGSM OR TSPC_EGSM]			
95		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_BCCHcarrierB_ho, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_20_B3, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			

96		[TSPC_DCS]			
97		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_BCCHcarrierB_hod, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_20_B3d, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
		local_handover7			
98		[TSPC_PGSM OR TSPC_EGSM]			
99		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_BCCHcarrierB_ho, 8, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_omit, CellChDes_Omit, TCV_ChMod, Freqchseq_01, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
100		[TSPC_DCS]			
101		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_BCCHcarrierB_hod, 8, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_20_B4d, CellChDes_Omit, TCV_ChMod, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
		local_HoCMD_nofh			
102		(TCV_HoCMD := HandOverCmd_nfh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			

Detailed Comments:

0. IUT enters state U10 with TCH/F_nonFH in cell A
1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.
2. HO from TCH/F_nonFH of cell A to TCH/F_nonFH in CELL B
3. HO from TCH/F_FH of cell A to TCH/F_FH in CELL B
4. HO from TCH/F_FH of cell A to TCH/F_nonFH in CELL B
5. HO from TCH/F_nonFH of cell A to TCH/H_FH in CELL B
6. HO from TCH/H_FH of cell A to TCH/H_FH in CELL B
7. HO from TCH/H_FH of cell A to TCH/H_nonFH in CELL B
8. HO from TCH/H_nonFH of cell A to TCH/H_FH in CELL B
9. HO from TCH/H_FH of cell A to TCH/F_nonFH in CELL B
10. The first handover access burst starts too late or too early. (10ms means 2 frames tolerance for
TCH/H channel, 5ms means 1 frame tolerance for TCH/F channel)

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_2_1

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a Non-synchronised handover, it continuously sends access bursts on the main DCCH until it receives a PHYSICAL INFORMATION from the SS. To test that the MS correctly takes the values of the Timing Advance and Starting Time information elements into account. To test that the MS activates the new channel correctly. To test that the MS correctly retransmits Layer 3 MM or CC messages, that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		(TCV_ia_ts:= '000'B, TCV_ts:= '000'B, TCV_Cntref:= TSPX_NoOfHoAccA, TCV_Horf:= TSPX_HoRefA, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(20), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+FullRateCh_B_1(C_Asynho, TCV_ts, TSPX_TscA, TCV_ChMod, FreqTCHb_hof3, FreqTCHb_hof3d, TimingAdv(20), C_BABR_0, C_cch_1Comb, C_BPM_3)			
7		+localtree_setArfcn			
8		+Est_MO_Call_init(C_CHSDCCH4_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(30), C_BCC, C_one, C_one)			1.
9	body	+ltree_sendHoCMD			
10		(TCV_chTch1 := TCV_ch)			
11		(TCV_ch:= C_FACCHF_B_1)			
12		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
13		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(20))		
14		+RR_hocomp1(500, TCV_chTch1)			
15		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		3.
16		+SendSeqNo_chk			4.
17	post	+ChanRel_end(TCV_ch)			
		ltree_sendHoCMD			

18		[TSPC_PGSM OR TSPC_EGSM]			
19		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscA, C_NCC_3, C_BCC_0, C_BCCHcarrierB_ho, 15, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_Omit, TCV_ChMod, Freqchseq_06, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.
20		[TSPC_DCS]			
21		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscA, C_NCC_3, C_BCC_0, C_BCCHcarrierB_hod , 11, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_20_B5d, Frql_omit, CellChDes_Omit, TCV_ChMod, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.
		localtree_setArfcn			
22		[TSPC_PGSM OR TSPC_EGSM]			
23		(TCV_chdescr_arfcn:= C_arfcnA)			
24		[TSPC_DCS]			
25		(TCV_chdescr_arfcn:= C_arfcnAd_4)			

Detailed Comments:

1. Initiate a call and L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from SDCCH/4_nonFH to TCH/F_FH.
3. Last L3-msg shall be retransmitted by the MS with the same sequence number of last L3-msg.
4. Check of the sequence number.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_2_2

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a non-synchronised handover, it continuously sends access bursts on the main DCCH until it receives a PHYSICAL INFORMATION from the SS. To test that the MS correctly takes the values of the Timing Advance and Starting Time information elements into account. To test that the MS activates the new channel correctly. To test that the MS correctly retransmits Layer 3 MM or CC messages, that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcE, TSPX_MOChRateE)			
3		(TCV_ia_ts:= '000'B, TCV_ts:= TSPX_TmSltNotZero, TCV_Cntref:= TSPX_NoOfHoAccD, TCV_Horf:= TSPX_HoRefB, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(20), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+HalfRateCh_B_1(TSPX_TCHHSubDef, C_Asynho, TCV_ts, TSPX_TscB, TCV_ChMod, FreqTCHb_hof4, FreqTCHb_hof4d, TimingAdv(20), C_BABR_0, C_cch_1Comb, C_BPM_3)			
7		+local_setArfcn			
8		+Est_MO_Call_init(C_CHSDCCH4_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(30), C_BCC, C_one, C_one)			1.
9	body	+local_sendHoCMD			
10		(TCV_chTch1 := TCV_ch)			
11		(TCV_ch:= OC_SubchOfFacchh(TSPX_TCHHSubDef, C_CellB, 1))			
12		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
13		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(20))		
14		+RR_hocomp1(750, TCV_chTch1)			
15		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		3.
16		+SendSeqNo_chk			4.
17	post	+ChanRel_end(			

		TCV_ch)			
		local_sendHoCMD			
18		(TCV_Tchtype := INT_TO_BIT((2 + BIT_TO_INT(TSPX_TCHHSubDef)), 5))			
19		[TSPC_PGSM OR TSPC_EGSM]			
20		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_Tchtype, TCV_ts, TSPX_TscB, C_NCC_3, C_BCC_0, C_BCCHcarrierB_ho, 3, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_21_B, TCV_ChMod, Freqchseq_omit, MoblAllc_252, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.
21		[TSPC_DCS]			
22		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_Tchtype, TCV_ts, TSPX_TscB, C_NCC_3, C_BCC_0, C_BCCHcarrierB_hod , 3, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised) , Frql_omit, Frql_omit, CellChDes_21_Bd, TCV_ChMod, Freqchseq_omit, MoblAllc_252d, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.
		local_setArfcn			
23		[TSPC_PGSM OR TSPC_EGSM]			
24		(TCV_chdescr_arfcn:= C_arfcnA)			
25		[TSPC_DCS]			
26		(TCV_chdescr_arfcn:= C_arfcnAd_4)			

Detailed Comments:

1. Initiate a call and L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from SDCCH/4_nonFH to TCH/H_FH.
3. Last L3-msg shall be retransmitted by the MS with the same sequence number of last L3-msg.
4. Check the sending sequence number and assign verdict.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_2_3

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a non-synchronised handover, it continuously sends access bursts on the main DCCH until it receives a PHYSICAL INFORMATION from the SS. To test that the MS correctly takes the values of the Timing Advance and Starting Time information elements into account. To test that the MS activates the new channel correctly. To test that the MS correctly retransmits Layer 3 MM or CC messages, that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcF, TSPX_MOChRateF)			
3		(TCV_ia_ts:= '000'B, TCV_ts:= '000'B, TCV_Cntref:= TSPX_NoOfHoAccG, TCV_Horf:= TSPX_HoRefC, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlB, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(20), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+SDCCH8_B_1(TSPX_SDCCH8SubA, C_Asynho, TCV_ts, TSPX_TscC, ChMod_sign, FreqSDCCH8b_hof1, FreqSDCCH8b_hof1d, TimingAdv(20), C_BABR_0, C_cch_1Comb, C_BPM_3)			
7		+localtree_setArfcn			
8		+Est_MO_Call_init(C_CHSDCCH4_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(30), C_BCC, C_one, C_one)			1.
9		+localtree_setArfcn			
10	body	+ltree_sendHoCMD			
11		(TCV_chTch1 := TCV_ch)			
12		(TCV_ch:= OC_SubchOfSdcch8(TSPX_SDCCH8SubA, C_CellB, 1))			
13		(TCV_Null := OM_SendNextOnHO(TCV_ch , TCV_Cntref))			
14		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(20))		
15		+RR_hocomp1(1500, TCV_chTch1)			
16		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		3.
17		+SendSeqNo_chk			4.

18	post	+ChanRel_end(TCV_ch)				
		ltree_sendHoCMD				
19		[TSPC_PGSM OR TSPC_EGSM]				
20		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_Tchtype, TCV_ts, TSPX_TscC, C_NCC_3, C_BCC_0, C_BCCHcarrierB_ho, 15, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_20_B7, CellChDes_Omit, ChMod_sign_iei, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.
21		[TSPC_DCS]				
22		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_Tchtype, TCV_ts, TSPX_TscC, C_NCC_3, C_BCC_0, C_BCCHcarrierB_hod , 2, C_HSN_0, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_20_B7d, CellChDes_Omit, ChMod_sign_iei, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.
		localtree_setArfcn				
23		(TCV_Tchtype := INT_TO_BIT((8 + BIT_TO_INT(TSPX_SDCCH8SubA)), 5))				
24		[TSPC_PGSM OR TSPC_EGSM]				
25		(TCV_chdescr_arfcn:= C_arfcnA)				
26		[TSPC_DCS]				
27		(TCV_chdescr_arfcn:= C_arfcnAd_4)				

Detailed Comments:

1. Initiate a call and L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from SDCCH/4_nonFH to SDCCH8_FH.
3. Last L3-msg shall be retransmitted by the MS with the same sequence number of last L3-msg.
4. To check the sending sequence number.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_2_4

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a non-synchronised handover, it continuously sends access bursts on the main DCCH until it receives a PHYSICAL INFORMATION from the SS. To test that the MS correctly takes the values of the Timing Advance and Starting Time information elements into account. To test that the MS activates the new channel correctly. To test that the MS correctly retransmits Layer 3 MM or CC messages, that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcG, TSPX_MOChRateG)			
3		(TCV_ia_ts:= TSPX_TmSltNotZero, TCV_ts:= TSPX_TmSltNotZero, TCV_Cntref:= TSPX_NoOfHoAccH, TCV_Horf:= TSPX_HoRefD, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlB, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		(TCV_ia_ts := TSPX_TmSltNotZero)			
6		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TSPX_TimadvA), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
7		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TCV_ia_ts, TSPX_TscA, ChMod_sign, FreqTCH(C_arfcnA), FreqTCH(C_arfcnAd_4), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
8		+SDCCH8_B_1(TSPX_SDCCH8SubA, C_Asynho, TCV_ts, TSPX_TscB, ChMod_sign, FreqSDCCH8b_hof2, FreqSDCCH8b_hof2d, TimingAdv(TSPX_TimadvA), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
9		+localtree_setTchArfcn			
10		+Est_MO_Call_init(C_CHSDCCH8_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(30), TSPX_TscA, C_one, C_one)			1.
11		+localtree_setTchArfcn			
12	body	+ltree_sendHoCMD			
13		(TCV_chTch1 := TCV_ch)			
14		(TCV_ch := OC_SubchOfSdcch8(			

		TSPX_SDCCH8SubA, C_CellB, 1))				
15		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))				
16		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TSPX_TimadvA))			
17		+RR_hocomp1(1500, TCV_chTch1)				
18		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		3.	
19		+SendSeqNo_chk			4.	
20	post	+ChanRel_end(TCV_ch)				
		ltree_sendHoCMD				
21		[TSPC_PGSM OR TSPC_EGSM]				
22		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_Tchtype, TCV_ts, TSPX_TscB, C_NCC_3, C_BCC_0, C_BCCHcarrierB_ho, 14, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_02, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.	
23		[TSPC_DCS]				
24		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_Tchtype, TCV_ts, TSPX_TscB, C_NCC_3, C_BCC_0, C_BCCHcarrierB_hod, 16, C_HSN_0, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_20_B8d, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_02, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.	
		localtree_setTchArfcn				
25		(TCV_Tchtype := INT_TO_BIT((8 + BIT_TO_INT(TSPX_SDCCH8SubA)), 5))				
26		[TSPC_PGSM OR TSPC_EGSM]				
27		(TCV_tch_arfcn:= C_arfcnA)				
28		[TSPC_DCS]				
29		(TCV_tch_arfcn:= C_arfcnAd_4)				

Detailed Comments:

1. Initiate a call and the last L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from SDCCH8_NonFH to SDCCH8_FH.
3. Last L3-msg shall be retransmitted by the MS with the same sequence number of last L3-msg.

4. Checking of the sending sequence number.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_2_5

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a non-synchronised handover, it continuously sends access bursts on the main DCCH until it receives a PHYSICAL INFORMATION from the SS. To test that the MS correctly takes the values of the Timing Advance and Starting Time information elements into account. To test that the MS activates the new channel correctly. To test that the MS correctly retransmits Layer 3 MM or CC messages, that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcH, TSPX_MOChRateH)			
3		(TCV_ia_ts:= TSPX_TmSltNotZero, TCV_ts:= TSPX_TmSltNotZero, TCV_Cntref:= TSPX_NoOfHoAccE, TCV_Horf:= TSPX_HoRefE, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlB, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		(TCV_ia_ts := TSPX_TmSltNotZero)			
6		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(20), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
7		+FullRateCh_A_1_nociph(C_Immass, TCV_ia_ts, TSPX_TscA, ChMod_sign, FreqTCH(TSPX_TCHcarrierA_ho), FreqTCH(TSPX_TCHcarrierA_hod), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
8		+local_set_arfcn			
9		+HalfRateCh_B_1(TSPX_TCHHSubDef, C_Asynho, TCV_ts, TSPX_TscB, ChMod_sign, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(20), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
10		+Est_MO_Call_init(C_CHTCHF_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(30), TSPX_TscA, C_one, C_one)			1.
11		+localtree_varinit			
12	body	L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_nfh(TCV_Tchtype, TCV_ts, TSPX_TscB,		2.

			C_NCC_3, C_BCC_0, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised) , ChMod_sign_iei, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
13		(TCV_chTch1 := TCV_ch)			
14		(TCV_ch:= TCV_chTch)			
15		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
16		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(20))		
17		+RR_hocomp1(750, TCV_chTch1)			
18		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		3.
19		+SendSeqNo_chk			4.
20	post	+ChanRel_end(TCV_ch)			
		localtree_varinit			
21		(TCV_Tchtype := INT_TO_BIT((2 + BIT_TO_INT(TSPX_TCHHSubDef)), 5))			
22		[TSPC_PGSM OR TSPC_EGSM]			
23		(TCV_tch_arfcn := TSPX_TCHcarrierB_ho, TCV_chdescr_arfcn := C_BCCHcarrierB_ho)			
24		[TSPC_DCS]			
25		(TCV_tch_arfcn :=TSPX_TCHcarrierB_hod, TCV_chdescr_arfcn := C_BCCHcarrierB_hod)			
		local_set_arfcn			
26		[TSPC_PGSM OR TSPC_EGSM]			
27		(TCV_tch_arfcn := TSPX_TCHcarrierA_ho, TCV_ch := TCV_chTch)			
28		[TSPC_DCS]			
29		(TCV_tch_arfcn :=TSPX_TCHcarrierA_hod, TCV_ch := TCV_chTch)			

Detailed Comments:

1. Initiate a call and last L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from TCH/F_NonFH to TCH/H_NonFH
3. Last L3-msg shall be retransmitted by the MS with the same sequence number of last L3-msg.
4. To check the sending sequence number.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_2_6

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a non-synchronised handover, it continuously sends access bursts on the main DCCH until it receives a PHYSICAL INFORMATION from the SS. To test that the MS correctly takes the values of the Timing Advance and Starting Time information elements into account. To test that the MS activates the new channel correctly. To test that the MS correctly retransmits Layer 3 MM or CC messages, that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcI, TSPX_MOChRateI)			
3		(TCV_ia_ts:= TSPX_TmSltNotZero, TCV_ts:= TSPX_TmSltNotZero, TCV_Cntref:= TSPX_NoOfHoAccB, TCV_Horf:= TSPX_HoRefF, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlC, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		(TCV_ia_ts := TSPX_TmSltNotZero)			
6		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(20), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
7		+HalfRateCh_A_1_nociph(TSPX_TCHHSubDef, C_Immass, TCV_ia_ts, TSPX_TscG, ChMod_sign, FreqTCHa_hof3, FreqTCHa_hof3d, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
8		(TCV_ch := TCV_chTch)			
9		+FullRateCh_B_1(C_Asynho, TCV_ts, TSPX_TscF, ChMod_sign, FreqTCHb_hof5, FreqTCHb_hof5d, TimingAdv(20), C_BABR_0, C_cch_1Comb, C_BPM_3)			
10		+Est_MO_Call_init(C_CHTCHH_FH, MoblAllc_20_A0, MoblAllc_20_A0, TimingAdv(30), TSPX_TscG, 17, 17)			1.
11	body	+local_HoCMD			
12		(TCV_chTch1 := TCV_ch)			
13		(TCV_ch := TCV_chTch)			
14		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
15		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(20))		

16			+RR_hocomp1(500, TCV_chTch1)			
17			L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg. mt)	SetupRcv(SetupInd_01)		3.
18			+SendSeqNo_chk			4.
19	post		+ChanRel_end(TCV_ch)			
		local_HoCMD				
20		[TSPC_PGSM OR TSPC_EGSM]				
21		L!DL_DatRqHoCmd		HndOvSnd(TCV_ch, HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscF, C_NCC_3, C_BCC_0, C_BCCHcarrierB_ho, 5, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_20_B, ChMod_sign_iei, Freqchseq_omit, MoblAllc_20_B1iei, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.
22		[TSPC_DCS]				
23		L!DL_DatRqHoCmd		HndOvSnd(TCV_ch, HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscF, C_NCC_3, C_BCC_0, C_BCCHcarrierB_hod , 9, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_20_B9d, CellChDes_Omit, ChMod_sign_iei, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.

Detailed Comments:

1. To initiate a call and the last L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from TCHH_FH to TCH/F_FH
3. Last L3-msg shall be retransmitted by the MS with the same sequence number of last L3-msg.
4. To check the sending sequence number.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_2_7

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a non-synchronised handover, it continuously sends access bursts on the main DCCH until it receives a PHYSICAL INFORMATION from the SS. To test that the MS correctly takes the values of the Timing Advance and Starting Time information elements into account. To test that the MS activates the new channel correctly. To test that the MS correctly retransmits Layer 3 MM or CC messages, that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcJ, TSPX_MOChRateJ)			
3		(TCV_ia_ts:= C_S0, TCV_ts:= TSPX_TmSltNotZero, TCV_Cntref:= TSPX_NoOfHoAccC, TCV_Horf:= TSPX_HoRefG, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlC, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TSPX_TimadvC), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+FullRateCh_A_1_nociph(C_Immass, TCV_ia_ts, TSPX_TscE, TCV_ChMod, FreqTCHa_hof2, FreqTCHa_hof2d, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
7		(TCV_ch := TCV_chTch)			
8		+FullRateCh_B_1(C_Asynho, TCV_ts, TSPX_TscB, TCV_ChMod, FreqTCHb_hof9, FreqTCHb_hof9d, TimingAdv(TSPX_TimadvC), C_BABR_0, C_cch_1Comb, C_BPM_3)			
9		+Est_MO_Call_init(C_CHTCHF_FH, MoblAllc_20_A2, MoblAllc_20_A2, TimingAdv(30), TSPX_TscE, 1, 1)			1.
10	body	+ltree_hoCMD			
11		(TCV_chTch1 := TCV_ch)			
12		(TCV_ch := TCV_chTch)			
13		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
14		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TSPX_TimadvC))		
15		+RR_hocomp1(500, TCV_chTch1)			

16			L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		3.
17			+SendSeqNo_chk			4.
18	post		+ChanRel_end(TCV_ch)			
		ltree_hoCMD				
19		[TSPC_PGSM OR TSPC_EGSM]				
20		L!DL_DatRqHoCmd		HndOvSnd(TCV_ch, HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscB, C_NCC_3, C_BCC_0, C_BCCHcarrierB_ho, 16, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_20_B2, CellChDes_Omit, TCV_ChMod, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.
21		[TSPC_DCS]				
22		L!DL_DatRqHoCmd		HndOvSnd(TCV_ch, HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscB, C_NCC_3, C_BCC_0, C_BCCHcarrierB_hod , 13, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_20_B0d, TCV_ChMod, Freqchseq_omit, MoblAllc_22, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.

Detailed Comments:

1. To initiate a call and the last L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from TCHH_FH to TCH/F_FH
3. Last L3-msg shall be retransmitted by the MS with the same sequence number of last L3-msg.
4. Checking of the sending sequence number.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_2_8

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a non-synchronised handover, it continuously sends access bursts on the main DCCH until it receives a PHYSICAL INFORMATION from the SS. To test that the MS correctly takes the values of the Timing Advance and Starting Time information elements into account. To test that the MS activates the new channel correctly. To test that the MS correctly retransmits Layer 3 MM or CC messages, that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcB, TSPX_MOChRateB)			
3		(TCV_ia_ts:= TSPX_TmSltNotZero, TCV_ts:= C_S0, TCV_Cntref:= TSPX_NoOfHoAccA, TCV_Horf:= TSPX_HoRefH, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlD, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		(TCV_ia_ts := TSPX_TmSltNotZero)			
6		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(20), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
7		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TCV_ia_ts, TSPX_TscA, ChMod_sign, FreqSDCCHa_hof1, FreqSDCCHa_hof1d, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
8		+FullRateCh_B_1(C_Asynho, TCV_ts, TSPX_TscC, TCV_ChMod, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(20), C_BABR_0, C_cch_1Comb, C_BPM_3)			
9		+Est_MO_Call_init(C_CHSDCCH8_FH, MoblAllc_281, MoblAllc_281d, TimingAdv(30), TSPX_TscA, 3, 3)			1.
10		+local_varinit			
11	body	L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_nfh('00001'B, TCV_ts, TSPX_TscC, C_NCC_3, C_BCC_0, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf,		2.

			TCV_Pwrlvl_ho, Synchi_omit, ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
12		(TCV_chTch1 := TCV_ch)				
13		(TCV_ch := TCV_chTch)				
14		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))				
15		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(20))			
16		+RR_hocomp1(500, TCV_chTch1)				
17		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)			3.
18		+SendSeqNo_chk				4.
19	post	+ChanRel_end(TCV_ch)				
		local_varinit				
20		[TSPC_PGSM OR TSPC_EGSM]				
21		(TCV_chdescr_arfcn := C_BCCHcarrierB_ho, TCV_tch_arfcn:= TSPX_TCHcarrierB_ho)				
22		[TSPC_DCS]				
23		(TCV_chdescr_arfcn := C_BCCHcarrierB_hod, TCV_tch_arfcn:= TSPX_TCHcarrierB_hod)				

Detailed Comments:

1. To initiate a call and the last L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from TCHH_FH to TCH/F_FH
3. Last L3-msg shall be retransmitted by the MS with the same sequence number of last L3-msg.
4. To check whether the sending sequence number is matched with the last unacknowledged one.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_2_9

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a non-synchronised handover, it continuously sends access bursts on the main DCCH until it receives a PHYSICAL INFORMATION from the SS. To test that the MS correctly takes the values of the Timing Advance and Starting Time information elements into account. To test that the MS activates the new channel correctly. To test that the MS correctly retransmits Layer 3 MM or CC messages, that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		(TCV_ia_ts:= TSPX_TmSltF, TCV_ts:= TSPX_TmSltB, TCV_Cntref:= TSPX_NoOfHoAccB, TCV_Horf:= TSPX_HoRefI, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlD, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		(TCV_ia_ts := TSPX_TmSltF)			
6		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(20), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
7		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TCV_ia_ts, TSPX_TscA, ChMod_sign, FreqTCH(C_arfcnC), FreqTCH(C_arfcn_tchAd), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
8		+local_setTimSlt			
9		+FullRateCh_B_1(C_Asynho, TCV_ts, TSPX_TscB, TCV_ChMod, FreqTCHb_hof6, FreqTCHb_hof6d, TimingAdv(20), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
10		+local_setTchArfcn			
11		+Est_MO_Call_init(C_CHSDCCH8_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(30), TSPX_TscA, C_one, C_one)			1.
12	body	+ltree_HoCMD			
13		(TCV_chTch1 := TCV_ch)			
14		(TCV_ch:= C_FACCHF_B_1)			
15		(TCV_Null :=			

			OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
16			L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(20))		
17			+RR_hocomp1(500, TCV_chTch1)			
18			L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		3.
19			+SendSeqNo_chk			4.
20	post		+ChanRel(TCV_ch)			
		ltree_HoCMD				
21		[TSPC_PGSM OR TSPC_EGSM]				
22		L!DL_DatRqHoCmd		HndOvSnd(TCV_ch, HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscB, C_NCC_3, C_BCC_0, C_BCCHcarrierB_ho, 2, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_Omit, TCV_ChMod, Freqchseq_03, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.
23		[TSPC_DCS]				
24		L!DL_DatRqHoCmd		HndOvSnd(TCV_ch, HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscB, C_NCC_3, C_BCC_0, C_BCCHcarrierB_ho, 3, C_HSN_0, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_20_B10d, Frql_omit, CellChDes_Omit, TCV_ChMod, Freqchseq_03, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.
		local_setTchArfcn				
25		[TSPC_PGSM OR TSPC_EGSM]				
26		(TCV_tch_arfcn:= C_arfcnC)				
27		[TSPC_DCS]				
28		(TCV_tch_arfcn:= C_arfcn_tchAd)				
		local_setTimSlt				
29		[TSPC_PGSM OR TSPC_EGSM]				
30		(TCV_ts:= C_S0)				
31		[TSPC_DCS]				
32		(TCV_ts := TSPX_TmSltNotZero)				

Detailed Comments:

1. Initiate a call and L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from TCHH_FH to TCH/F_FH
3. Last L3-msg shall be retransmitted by the MS with the same sequence number of last L3-msg.
4. Checking of the sending sequence number.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_2_10

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a non-synchronised handover, it continuously sends access bursts on the main DCCH until it receives a PHYSICAL INFORMATION from the SS. To test that the MS correctly takes the values of the Timing Advance and Starting Time information elements into account. To test that the MS activates the new channel correctly. To test that the MS correctly retransmits Layer 3 MM or CC messages, that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcC, TSPX_MOChRateC)			
3		(TCV_ia_ts:= TSPX_TmSltG, TCV_ts:= TSPX_TmSltNotZero, TCV_Cntref:= TSPX_NoOfHoAccF, TCV_Horf:= TSPX_HoRefJ, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlD, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		(TCV_ia_ts := TSPX_TmSltG)			
6		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TSPX_TimadvB), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
7		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TCV_ia_ts, TSPX_TscB, ChMod_sign, FreqTCH(C_arfcnC), FreqTCH(C_arfcn_tchAd), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
8		+HalfRateCh_B_1(TSPX_TCHHSubDef, C_Asynho, TCV_ts, TSPX_TscC, TCV_ChMod, FreqTCHb_hof7, FreqTCHb_hof7d, TimingAdv(TSPX_TimadvB), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
9		+localtree_TchArfcn			
10		+Est_MO_Call_init(C_CHSDCCH8_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(30), TSPX_TscB, C_one, C_one)			1.
11	body	+ltree_sendHoCMD			
12		(TCV_chTch1 := TCV_ch)			
13		(TCV_ch:= OC_SubchOfFacchh(			

		TSPX_TCHHSubDef, C_CellB, 1))			
14		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
15		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TSPX_TimadvB))		
16		+RR_hocomp1(750, TCV_chTch1)			
17		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		3.
18		+SendSeqNo_chk			4.
19	post	+ChanRel(TCV_ch)			
		ltree_sendHoCMD			
20		(TCV_Tchtype := INT_TO_BIT((2 + BIT_TO_INT(TSPX_TCHHSubDef)), 5))			
21		[TSPC_PGSM OR TSPC_EGSM]			
22		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_Tchtype, TCV_ts, TSPX_TscC, C_NCC_3, C_BCC_0, C_BCCHcarrierB_ho, 2, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_Omit, TCV_ChMod, Freqchseq_04, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.
23		[TSPC_DCS]			
24		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_Tchtype, TCV_ts, TSPX_TscC, C_NCC_3, C_BCC_0, C_BCCHcarrierB_hod, 15, C_HSN_0, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_20_11d, CellChDes_Omit, TCV_ChMod, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.
		localtree_TchArfcn			
25		[TSPC_PGSM OR TSPC_EGSM]			
26		(TCV_tch_arfcn:= C_arfcnC)			
27		[TSPC_DCS]			
28		(TCV_tch_arfcn:= C_arfcn_tchAd)			

Detailed Comments:

1. Initiate a call and L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from TCHH_FH to TCH/F_FH
3. Last L3-msg shall be retransmitted by the MS with the same sequence number of last L3-msg.
4. Checking of the sending sequence number.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_3_1

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a finely synchronised handover to a synchronised cell, it sends 4 access bursts on the main DCCH and then activates the channel correctly, taking into account the value of any Starting Time information element and correctly calculating the timing advance to use.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltNotZero1, TCV_ts:= TSPX_TmSltNotZero, TCV_Horf:= TSPX_HoRefA, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlD, 5))			
3		+BasicServiceMT(TSPX_MTBscSvcI, C_Full)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TSPX_y), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubB, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TSPX_k*2 + TSPX_y), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 0)			BCCH of cell B is TSPX_k bit periods after cell A
6		+FullRateCh_B_1(C_Synho, TCV_ts, TSPX_TscB, TCV_ChMod, FreqTCH(C_BCCHcarrierB_ho), FreqTCH(C_BCCHcarrierB_hod), TimingAdv(TSPX_k*2 + TSPX_y), C_BABR_0, C_cch_1Comb, C_BPM_3)			
7		+FullRateCh_A_1(C_Ass, TCV_asscmd_ts, TSPX_TscDef, TCV_ChMod, FreqTCHa_hof1, FreqTCHa_hof1d, C_Noarfcn, C_Noarfcn, TimingAdv(TSPX_y), C_BABR_0, C_cch_1Comb, C_BPM_3)			
8		+local_varinit1			
9		+Est_MT_Call_FH(TimingAdv(TSPX_y), C_Full, TCV_maio, TSPX_HSN, TSPX_RANDDef, TCV_asscmd_ts, TSPX_TscDef, TSPX_TCHHSubDef)			1.
10		+local_varinit2			
11		(TCV_Fn1:= OM_ComingFn(TCV_chTch), TCV_Null := OM_SendNextOn(TCV_chTch, TCV_Fn1))			
12	body	L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh('00001'B, TCV_ts, TSPX_TscB, C_NCC, C_BCC,		2.

			TCV_chdescr_arfcn, TCV_chdescr_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), TCV_ChMod, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
13		(TCV_ch := C_FACCHF_B_1, TCV_sacch := C_SACCHF_B_1)			
14		+RR_hocomp3(650)			
15		+Timadv_Pwrlvl_chk(TSPX_k, TSPX_y, TCV_Pwrlvl_ho)			3.
16	post	+ChanRel_end(TCV_ch)			
		local_varinit1			
17		[TSPC_PGSM OR TSPC_EGSM]			
18		(TCV_chdescr_arfcn:= C_arfcnA, TCV_maio := INT_TO_BIT((TSPX_MAIO MOD 12), 6))			
19		[TSPC_DCS]			
20		(TCV_chdescr_arfcn:= C_arfcnAd, TCV_maio := INT_TO_BIT((TSPX_MAIO MOD 9), 6))			
		local_varinit2			
21		[TSPC_PGSM OR TSPC_EGSM]			
22		(TCV_chdescr_arfcn := C_BCCHcarrierB_ho)			
23		[TSPC_DCS]			
24		(TCV_chdescr_arfcn := C_BCCHcarrierB_hod)			

Detailed Comments:
1. IUT enters state U10 with TCH/F_FH in cell A
2. HO from TCH/F_FH of cell A to TCH/F_nonFH in CELL B.
3. To check the power level and timing advance in the L1 head.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_3_2

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a finely synchronised handover to a synchronised cell, it sends 4 access bursts on the main DCCH and then activates the channel correctly, taking into account the value of any Starting Time information element and correctly calculating the timing advance to use.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltNotZero1, TCV_ts:= TSPX_TmSltNotZero, TCV_Horf:= TSPX_HoRefB, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlC, 5))			
3		+BasicServiceMT(TSPX_MTBscSvcI, C_Half)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TSPX_y), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubB, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TSPX_k*2 + TSPX_y), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 0)			BCCH of cell B is TSPX_k bit periods after cell A
6		+HalfRateCh_B_1(TSPX_TCHHSubDef, C_Synho, TCV_ts, TSPX_TscC, TCV_ChMod, FreqTCH(C_BCCHcarrierB_ho), FreqTCH(C_BCCHcarrierB_hod), TimingAdv(TSPX_k*2 + TSPX_y), C_BABR_0, C_cch_1Comb, C_BPM_3)			
7		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Ass, TCV_asscmd_ts, TSPX_TscDef, TCV_ChMod, FreqTCHa_hof1, FreqTCHa_hof1d, C_Noarfcn, C_Noarfcn, TimingAdv(TSPX_y), C_BABR_0, C_cch_1Comb, C_BPM_3)			
8		+localtree_varinit1			
9		+Est_MT_Call_FH(TimingAdv(TSPX_y), C_Half, TCV_maio, TSPX_HSN, TSPX_RANDDef, TCV_asscmd_ts, TSPX_TscDef, TSPX_TCHHSubDef)			1.
10		+localtree_varinit2			
11		(TCV_Fn1:= OM_ComingFn(TCV_chTch), TCV_Null := OM_SendNextOn(TCV_chTch, TCV_Fn1))			
12	body	L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TCV_ts, TSPX_TscC, C_NCC, C_BCC,		2.

			TCV_chdescr_arfcn, TCV_chdescr_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), TCV_ChMod, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
13		(TCV_ch:= OC_SubchOfFacchh(TSPX_TCHHSubDef, C_CellB, 1), TCV_sacch := OC_SubchOfSacchh(TSPX_TCHHSubDef, C_CellB, 1))			
14		+RR_hocomp3(900)			
15		+Timadv_Pwrlvl_chk(TSPX_k, TSPX_y, TCV_Pwrlvl_ho)			3.
16	post	+ChanRel_end(TCV_ch)			
		localtree_varinit1			
17		[TSPC_PGSM OR TSPC_EGSM]			
18		(TCV_chdescr_arfcn:= C_arfcnA, TCV_maio := INT_TO_BIT((TSPX_MAIO MOD 12), 6))			
19		[TSPC_DCS]			
20		(TCV_chdescr_arfcn:= C_arfcnAd, TCV_maio := INT_TO_BIT((TSPX_MAIO MOD 9), 6))			
		localtree_varinit2			
21		(TCV_Tchtype := INT_TO_BIT((2 + BIT_TO_INT(TSPX_TCHHSubDef)), 5))			
22		[TSPC_PGSM OR TSPC_EGSM]			
23		(TCV_chdescr_arfcn := C_BCCHcarrierB_ho)			
24		[TSPC_DCS]			
25		(TCV_chdescr_arfcn := C_BCCHcarrierB_hod)			

Detailed Comments:
1. IUT enters state U10 with TCH/H_FH in cell A
2. HO from TCH/H_FH of cell A to TCH/H_nonFH in CELL B.
3. To check the power level and timing advance in the L1 head.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_4_1

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a finely synchronised handover to a synchronised cell, it sends 4 access bursts on the main DCCH and then activates the channel correctly, taking into account the value of any Starting Time information element and correctly calculating the timing advance to use. To test that the MS correctly retransmits Layer 3 MM or CC messages that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcD, TSPX_MOChRateD)			
3		(TCV_ts:= TSPX_TmSltNotZero, TCV_Cntref:= TSPX_NoOfHoAccA, TCV_Horf:= TSPX_HoRefA, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TSPX_y), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TSPX_k*2 + TSPX_y), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			The BCCH of cell B is k bit periods after BCCH of cell A to simulate the two cells have different propagation delay. The frame numbers of the two cells is different by 100.
6		(TCV_ia_ts:= TSPX_TmSltNotZero)			
7		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TCV_ia_ts, TSPX_TscC, ChMod_sign, FreqSDCCHa_hof2, FreqSDCCHa_hof2d, TimingAdv(TSPX_y), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
8		(TCV_temp := ((TSPX_k*2 + TSPX_y) MOD 256))			
9		+SDCCH8_B_1(TSPX_SDCCH8SubDef, C_Synho, TCV_ts, TSPX_TscA, ChMod_sign, FreqSDCCH8b_hof3, FreqSDCCH8b_hof2d, TimingAdv(TCV_temp), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
10		+Est_MO_Call_init(C_CHSDCCH8_FH, MoblAllc_282, MoblAllc_20_B1, TimingAdv(TSPX_y), TSPX_TscC, 16, 16)			1.
11		(TCV_Fn1:= OM_ComingFn(TCV_ch), TCV_Null := OM_SendNextOn(TCV_ch, TCV_Fn1), TCV_Fn1 := OC_FnInc(TCV_Fn1, 100))			
12	body	+Itree_hoCMD			
13		(TCV_ch := TCV_chTch,			

		TCV_sacch := TCV_sacchTch)			
14		+RR_hocomp3(1500)			
15		+Timadv_Pwrlvl_chk(TSPX_k, TSPX_y, TCV_Pwrlvl_ho)			3.
16		L?DL_DatInSetup (TCV_Mt1 :=DL_DatInSetup.msg. mt)	SetupRcv(SetupInd_01)		4.
17		+SendSeqNo_chk			5.
18	post	+ChanRel_end(TCV_ch)			
		ltree_hoCMD			
19		(TCV_Tchtype := INT_TO_BIT((8 + BIT_TO_INT(TSPX_SDCCH8SubDef)), 5))			
20		[TSPC_PGSM OR TSPC_EGSM]			
21		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_Tchtype, TCV_ts, TSPX_TscA, C_NCC, C_BCC, C_BCCHcarrierB_ho, 16, C_HSN_0, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), Frql_omit, Frql_20_B0, CellChDes_Omit, ChMod_sign_iei, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.
22		[TSPC_DCS]			
23		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_Tchtype, TCV_ts, TSPX_TscA, C_NCC, C_BCC, C_BCCHcarrierB_hod , 16, C_HSN_0, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), Frql_omit, Frql_20_B0d, CellChDes_Omit, ChMod_sign_iei, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		2.

Detailed Comments:

1. To initiate a call and the last L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from SDCCH/8_FH to SDCCH/8_FH
3. To check Timing Advance and Power Level in the next uplink SACCH head
4. Last L3-msg shall be retransmitted by the MS with the same sequence number of last L3-msg.
5. to check the sending sequence number.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_4_2

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a finely synchronised handover to a synchronised cell, it sends 4 access bursts on the main DCCH and then activates the channel correctly, taking into account the value of any Starting Time information element and correctly calculating the timing advance to use. To test that the MS correctly retransmits Layer 3 MM or CC messages that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcE, TSPX_MOChRateE)			
3		(TCV_ts:= C_S0, TCV_Horf:= TSPX_HoRefB, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlB, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TSPX_y), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Synho, TCV_slot, TCV_tsc, TimingAdv(TSPX_k*2 + TSPX_y), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			The BCCH of cell B is k bit periods after BCCH of cell A to simulate the two cells have different propagation delay. The frame numbers of the two cells is different by 100.
6		(TCV_ch1 := OC_SubchOfSdcch4(TSPX_SDCCH4SubDef, C_CellB), TCV_Null := OM_CphMd(TCV_ch1, CphMod_01, TCV_CphKey))			put the handover target channel in ciphering mode
7		(TCV_ia_ts:= TSPX_TmSltNotZero)			
8		+local_set_DCS_Slot			
9		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TCV_ia_ts, TSPX_TscF, ChMod_sign, FreqSDCCHa_hof3, FreqSDCCHa_hof3d, TimingAdv(TSPX_y), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
10		+Est_MO_Call_init(C_CHSDCCH8_FH, MoblAllc_20_A3, MoblAllc_20_A3d, TimingAdv(TSPX_y), TSPX_TscF, 15, 15)			1.
11		(TCV_Fn := OM_ComingFn(TCV_ch), TCV_Null:= OM_SendNextOn(TCV_ch, TCV_Fn), TCV_Fn1 := TCV_Fn, TCV_Fn1 := OC_FnInc(TCV_Fn1, 100))			
12	body	+local_set_arfcn			
13		L!DL_DatRqHoCmd (DL_DatRqHoCmd.msg.strt := OC_StartTime(TCV_Fn, C_StartingTimeHO, 1))	HndOvSnd(TCV_ch, HandOverCmd_nfh(TCV_Tchtype, C_S0, C_BCC, C_NCC,		2.

				C_BCC, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
14			(TCV_ch := TCV_ch1, TCV_sacch := TCV_sacch_B)			
15			+RR_hocomp3(2600)			
16			+Timadv_Pwrlvl_chk(TSPX_k, TSPX_y, TCV_Pwrlvl_ho)			3.
17			L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg. mt)	SetupRcv(SetupInd_01)		4.
18			+SendSeqNo_chk			5.
19	post		+ChanRel_end(TCV_ch)			
		local_set_DCS_Slot				
20		[TSPC_PGSM OR TSPC_EGSM]				
21		[TSPC_DCS]				
22		(TCV_ia_ts := C_S0)				
		local_set_arfcn				
23		(TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubDef)), 5))				
24		[TSPC_PGSM OR TSPC_EGSM]				
25		(TCV_tch_arfcn := C_BCCHcarrierB_ho, TCV_chdescr_arfcn := C_BCCHcarrierB_ho)				
26		[TSPC_DCS]				
27		(TCV_tch_arfcn := C_BCCHcarrierB_hod, TCV_chdescr_arfcn := C_BCCHcarrierB_hod)				

Detailed Comments:

1. To initiate a call and the last L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from SDCCH/8_FH to SDCCH/4_NoFH
3. To check whether the power level and timing advance in the L1 head is correct.
4. Last L3-msg shall be retransmitted by the MS with the same sequence number of last L3-msg.
5. To check the sending sequence number.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_4_3

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a finely synchronised handover to a synchronised cell, it sends 4 access bursts on the main DCCH and then activates the channel correctly, taking into account the value of any Starting Time information element and correctly calculating the timing advance to use. To test that the MS correctly retransmits Layer 3 MM or CC messages that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcF, C_Full)			
3		(TCV_ts:= C_S0, TCV_Horf:= TSPX_HoRefA, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TSPX_y), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TSPX_k*2 + TSPX_y), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			The BCCH of cell B is k bit periods after BCCH of cell A to simulate the two cells have different propagation delay. The frame numbers of the two cells is different by 100.
6		(TCV_ia_ts:= TSPX_TmSltNotZero)			
7		+FullRateCh_A_1_nociph(C_Immass, TCV_ia_ts, TSPX_TscD, TCV_ChMod, FreqTCH(C_arfcnA), FreqTCH(C_arfcnAd_4), TimingAdv(TSPX_y), C_BABR_0, C_cch_1Comb, C_BPM_3)			
8		(TCV_ch := TCV_chTch)			
9		+FullRateCh_B_1(C_Synho, TCV_ts, TSPX_TscB, TCV_ChMod, FreqTCHb_hof8, FreqTCHb_hof8d, TimingAdv(TSPX_k*2 + TSPX_y), C_BABR_0, C_cch_1Comb, C_BPM_3)			
10		+local_set_arfcn			
11		+local_continue			
		local_continue			
12		+Est_MO_Call_init(C_CHTCHF_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(TSPX_y), TSPX_TscD, C_one, C_one)			1.
13		(TCV_Fn1:= OM_ComingFn(TCV_ch), TCV_Null := OM_SendNextOn(TCV_ch, TCV_Fn1), TCV_Fn1 := OC_FnInc(TCV_Fn1, 100))			
14	body	+ltree_hoCMD			2.
15		(TCV_ch := TCV_chTch, TCV_sacch := TCV_sacchTch)			
16		+RR_hocomp3(650)			

17		+Timadv_Pwrlvl_chk(TSPX_k, TSPX_y, TCV_Pwrlvl_ho)			3.
18		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		4.
19		+SendSeqNo_chk			5.
20	post	+ChanRel_end(TCV_ch)			
		ltree_hoCMD			
21		[TSPC_PGSM OR TSPC_EGSM]			
22		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscB, C_NCC, C_BCC, C_BCCHcarrierB_ho, 4, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), Frql_omit, Frql_omit, CellChDes_Omit, TCV_ChMod, Freqchseq_05, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
23		[TSPC_DCS]			
24		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscB, C_NCC, C_BCC, C_BCCHcarrierB_hod , 3, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), Frql_20_B12d, Frql_omit, CellChDes_Omit, TCV_ChMod, Freqchseq_06, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
		local_set_arfcn			
25		[TSPC_PGSM OR TSPC_EGSM]			
26		(TCV_tch_arfcn := C_arfcnA)			
27		[TSPC_DCS]			
28		(TCV_tch_arfcn := C_arfcnAd_4)			

Detailed Comments:

1. To initiate a call and the last L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from TCH/F_NoFH to TCH/F_FH
3. To check the power level and timing advance in L1 head
4. Last L3-msg shall be retransmitted by the MS with the same sequence number of last L3-msg.
5. Checking of the sending sequence number.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_4_4

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a finely synchronised handover to a synchronised cell, it sends 4 access bursts on the main DCCH and then activates the channel correctly, taking into account the value of any Starting Time information element and correctly calculating the timing advance to use. To test that the MS correctly retransmits Layer 3 MM or CC messages that were not acknowledged by Layer 2 before the Handover, after completion of the Handover.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcG, TSPX_MOChRateG)			
3		(TCV_ts:= TSPX_TmSltNotZero, TCV_Horf:= TSPX_HoRefD, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlD, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TSPX_y), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TSPX_k*2 + TSPX_y), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			The BCCH of cell B is k bit periods after BCCH of cell A to simulate the two cells have different propagation delay. The frame numbers of the two cells is different by 100.
6		(TCV_ia_ts:= TSPX_TmSltE)			
7		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TCV_ia_ts, TSPX_TscDef, ChMod_sign, FreqTCH(TSPX_TCHcarrierA_ho), FreqTCH(TSPX_TCHcarrierA_hod), TimingAdv(TSPX_y), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
8		+FullRateCh_B_1(C_Synho, TCV_ts, TSPX_TscA, TCV_ChMod, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TSPX_k*2 + TSPX_y), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
9		+local_set_arfcn1			
10		+Est_MO_Call_init(C_CHSDCCH8_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(TSPX_y), TSPX_TscDef, C_one, C_one)			1.
11		+local_set_arfcn2			
12		(TCV_Fn1:= OM_ComingFn(TCV_ch), TCV_Null := OM_SendNextOn(TCV_ch, TCV_Fn1), TCV_Fn1 := OC_FnInc(TCV_Fn1, 100))			
13	body	L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_nfh(		2.

			'00001'B, TCV_ts, TSPX_TscA, C_NCC, C_BCC, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
14		+local_continue			
		local_continue			
15		(TCV_ch := TCV_chTch, TCV_sacch := TCV_sacchTch)			
16		+RR_hocomp3(650)			
17		+Timadv_Pwrlvl_chk(TSPX_k, TSPX_y, TCV_Pwrlvl_ho)			3.
18		L?DL_DatInSetup (TCV_Mt1 :=DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
19		+SendSeqNo_chk			4.
20	post	+ChanRel_end(TCV_ch)			
		local_set_arfcn2			
21		[TSPC_PGSM OR TSPC_EGSM]			
22		(TCV_chdescr_arfcn := C_BCCHcarrierB_ho, TCV_tch_arfcn := TSPX_TCHcarrierB_ho)			
23		[TSPC_DCS]			
24		(TCV_chdescr_arfcn := C_BCCHcarrierB_hod, TCV_tch_arfcn := TSPX_TCHcarrierB_hod)			
		local_set_arfcn1			
25		[TSPC_PGSM OR TSPC_EGSM]			
26		(TCV_tch_arfcn := TSPX_TCHcarrierA_ho)			
27		[TSPC_DCS]			
28		(TCV_tch_arfcn := TSPX_TCHcarrierA_hod)			

Detailed Comments:

1. To initiate a call and the last L2_frame with the Setup will not be acknowledged by the SS_L2.
2. HO from SDCCH/8_FH to TCH/F_NoFH
3. To check power level and timing advance in the L1 head
4. To check the sending sequence number.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_5_1

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that when the MS is ordered to make a pre-synchronised handover to another cell, it sends 4 access bursts on the main DCCH and then activates the channel correctly and correctly calculates the time to transmit.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcA, C_Full)			
3		(TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubC, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+StartCellB_ho(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(1), 5, 1, 0, 0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k1, 0)			2.
7		+FullRateCh_B_1(C_Synho, TSPX_TmSltC, TSPX_TscC, TCV_ChMod, FreqTCH(TSPX_TCHcarrierB), FreqTCH(TSPX_TCHcarrierBd), TimingAdv(1), C_BABR_0, C_cch_1Comb, C_BPM_3)			
8		(TCV_chTch := C_FACCHF_A_1, TCV_chTch1 := C_FACCHF_B_1, TCV_sacchTch1 := C_SACCHF_B_1)			
9		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubC, C_MaxPwrLvlG, C_MaxPwrLvlD, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			3.
10		(TCV_Fn := OM_ComingFn(TCV_chTch), TCV_Null := OM_SendNextOn(TCV_chTch, TCV_Fn))			
11	body	+local_HoCMD			4.
12		+local_continue			
		local_continue			
13		(TCV_L1Head := OM_GetL1Hd(TCV_sacchTch1), TCV_Cnt := 0)			
14		REPEAT local_hoacc(TCV_chTch1) UNTIL [TCV_Cnt = 4]			

15		(TCV_HoaccPara := OM_GetHoaccPara(TCV_chTch1))			
16		+Check_hoaccTiming(TCV_HoaccPara, '0000000'B)			5.
17		+Check_hoaccPwr(TCV_HoaccPara, TCV_Pwrlvl_ho)			5.
18		L?DL_EstIn CANCEL T_dly1	DLEstInd(TCV_chTch1)		
19		L?DL_DatInHoCom(TCV_Fn1 := DL_DatInHoCom.fn)	HndOvCmpRcv(TCV_chTch1, HandOverCmp_01)		
20	L_0229	[TCV_L1Head.ta = '0000001'B]		(P)	6.
21		+Check_Time(650)			
22		+PostMainLinkRel(TCV_chTch1)			
23	L_0230	[TCV_L1Head.ta <> '0000001'B]		(F)	
24		+Check_Time(650)			
25		+PostMainLinkRel(TCV_chTch1)			
		local_hoacc(ch:LOGICCH)			
26		L?DL_RacInHoacc (TCV_Hrf := DL_RacInHoacc.msg.horf)	HndOvAccRcv(ch, HandOverAcc_01)		
27	L_0231	[TCV_Hrf <> TSPX_HoRefD]		(F)	
28		(TCV_Cnt := TCV_Cnt + 1)			
29	L_0232	[TCV_Hrf = TSPX_HoRefD]		(P)	7.
30		(TCV_Cnt := TCV_Cnt + 1)			
		local_HoCMD			
31		(TCV_Tchtype := '00001'B)			
32		[TSPC_PGSM OR TSPC_EGSM]			
33		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC, TSPX_TscC, C_NCC, C_BCC, C_arfcnB, TSPX_TCHcarrierB, TSPX_HoRefD, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_pre_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
34		[TSPC_DCS]			
35		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC, TSPX_TscC, C_NCC, C_BCC, C_arfcnBd, TSPX_TCHcarrierBd, TSPX_HoRefD, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_pre_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		

Detailed Comments:

1. To setup BCCH, CCCH, SDCCH4 and a full rate traffic channel for cell A.
2. To setup BCCH, CCCH, SDCCH4 and a full rate traffic channel for cell B.
3. To bring the MS into the U10 state.
4. Pre-synch handover without TA IE.

	5. Check access power level and TA. 6. The received timing advance is 1 bit period, pass. 7. The received handover reference is correct.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_5_2

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a pre-synchronised handover to another cell, it sends 4 access bursts on the main DCCH and then activates the channel correctly and correctly calculates the time to transmit. To test that the MS correctly retransmits Layer 3 MM or CC messages that were not acknowledged by Layer 2 before the Handover, after completion of the Handover. To test that the MS correctly reports on the time difference between the cells.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcJ, C_Full)			
3		(TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TSPX_y), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubB, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(TSPX_y), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+StartCellB_ho(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(9), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k1, 0)			2.
7		+FullRateCh_B_1(C_Synho, TSPX_TmSltC, TSPX_TscC, TCV_ChMod, FreqTCH(TSPX_TCHcarrierB), FreqTCH(TSPX_TCHcarrierBd), TimingAdv(9), C_BABR_0, C_cch_1Comb, C_BPM_3)			
8		(TCV_chTch1 := C_FACCHF_B_1, TCV_sacchTch1 := C_SACCHF_B_1)			
9	body	+InitCall(TCV_Service)			
10		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
11		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubB, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(TSPX_y)))		3.
12		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
13		(TCV_Null := OM_NoL2Ack(2, TCV_ch))			
14		+Ciphering_on(TCV_ch)			

15		+localtree1			
		localtree1			
16		L?DL_DatInSetup (TCV_Mt :=DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
17		(TCV_Fn := OM_ComingFn(TCV_ch), TCV_Null := OM_SendNextOn(TCV_ch, TCV_Fn))			
18		+local_hoCMD			
19		+Varinit_fix(C_CellB, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
20		(TCV_L1Head := OM_GetL1Hd(TCV_sacchTch1))			
21		+localtree2(TCV_chTch1)			
22		L?DL_EstIn	DLEstInd(TCV_chTch1)		
23	L_0239	L?DL_DatInHoCom(TCV_Fn1 := DL_DatInHoCom.fn, TCV_Td := DL_DatInHoCom.msg.motdif.value)	HndOvCmpRcv(TCV_chTch1, HandOverCmp_01)	(P)	
24		+CheckMTDif(TCV_Td, TSPX_y, TSPX_k1)			
25	L_0240	[TCV_L1Head.ta <>'0001001'B]		(F)	
26		+Check_Time(650)			
27		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
28		+SendSeqNo_chk			
29		+PostMainLinkRel(TCV_chTch1)			
30	L_0241	[TCV_L1Head.ta = '0001001'B]		(P)	
31		+Check_Time(650)			
32		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
33		+SendSeqNo_chk			
34		+PostMainLinkRel(TCV_chTch1)			
		localtree2(ch:LOGICCH)			
35		(TCV_Cnt := 0)			
36		REPEAT localtree3(ch) UNTIL [TCV_Cnt = 4]			
37		(TCV_HoaccPara := OM_GetHoaccPara(ch))			
38		+Check_hoaccPwr(TCV_HoaccPara, TCV_Pwrlvl_ho)			
		localtree3(ch:LOGICCH)			
39		L?DL_RacInHoacc (TCV_Hrf := DL_RacInHoacc.msg.horf)	HndOvAccRcv(ch, HandOverAcc_01)		
40	L_0244	[TCV_Hrf <> TSPX_HoRefF]		(F)	
41		(TCV_Cnt := TCV_Cnt + 1)			
42	L_0245	[TCV_Hrf = TSPX_HoRefF]		(P)	
43		(TCV_Cnt := TCV_Cnt + 1)			
		local_hoCMD			
44		(TCV_Tchtype := '00001'B)			
45		[TSPC_PGSM OR TSPC_EGSM]			
46		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC, TSPX_TscC, C_NCC, C_BCC, C_arfcnB, TSPX_TCHcarrierB, TSPX_HoRefF, TCV_Pwrlvl_ho, Synchi(C_report_otd, C_pre_synchronised),		

			ChMod_omit, RelTmdDif_omit, TimingAdv_iei(9), CphMod_omit))		
47		[TSPC_DCS]			
48		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC, TSPX_TscC, C_NCC, C_BCC, C_arfcnBd, TSPX_TCHcarrierBd, TSPX_HoRefF, TCV_Pwrlvl_ho, Synchi(C_report_otd, C_pre_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_iei(9), CphMod_omit))		

Detailed Comments:

1. Cell A with BCCH, CCCH, SDCCH4 and a traffic channel.
2. Cell B with BCCH, CCCH, SDCCH4 and a traffic channel. The BCCH of cell B is TSPX_k bit periods after the BCCH of cell A.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_6

Group: GSM_L3_MS_v4220/RR/

Purpose: To test that when the MS is ordered to make a pseudo synchronised handover to another cell, it sends 4 access bursts on the main DCCH and then activates the channel correctly and correctly calculates the time to transmit. To test that the MS correctly reports the time difference between the cells.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcC, C_Full)			
3		(TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TSPX_y2), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(TSPX_y2), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		(TCV_temp := TSPX_y2 - 10)			
7		+StartCellB_ho(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_temp), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k2, 0)			2.
8		+FullRateCh_B_1(C_Ass, TSPX_TmSltC, TSPX_TscC, TCV_ChMod, FreqTCH(TSPX_TCHcarrierB), FreqTCH(TSPX_TCHcarrierBd), TimingAdv(TCV_temp), C_BABR_0, C_cch_1Comb, C_BPM_3)			
9		(TCV_chTch := C_FACCHF_A_1, TCV_chTch1 := C_FACCHF_B_1, TCV_sacchTch1 := C_SACCHF_B_1)			
10		+PreEnterCCstateU10_r01(TimingAdv(TSPX_y2), TSPX_SDCCH4SubDef, 0, 3, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			3.
11		+local_HoCMD			4.
12		+localtree(TCV_chTch1)			
13		+localtree1			
		localtree1			
14		(TCV_Fn := OM_ReturnFn(TCV_chTch))			
15		L?DL_EstIn	DLEstInd(		

			TCV_chTch1)		
16		L?DL_DatInHoCom (TCV_Td := DL_DatInHoCom.msg.motdif.value, TCV_Fn1 := DL_DatInHoCom.fn)	HndOvCmpRcv(TCV_chTch1, HandOverCmp_01)		5.
17		+CheckMTDif(TCV_Td, TSPX_y2, TSPX_k2)			
18		+Check_Time(650)			
19	L_0248	[(BIT_TO_INT(TCV_L1Head.ta) >= (TSPX_y2 - 10 - 2)) OR(BIT_TO_INT(TCV_L1Head.ta) <= (TSPX_y2 - 10 + 2))]		(P)	6.
20		+PostMainLinkRel(TCV_chTch1)			
21	L_0249	[C_Yes]		(F)	
22		+PostMainLinkRel(TCV_chTch1)			
		localtree(ch:LOGICCH)			
23		(TCV_Cnt := 0, TCV_L1Head := OM_GetL1Hd(TCV_sacchTch1))			
24		REPEAT localtree2(ch) UNTIL [TCV_Cnt = 4]			
25		(TCV_HoaccPara := OM_GetHoaccPara(ch))			
26		+Check_hoaccTiming(TCV_HoaccPara, '0000000'B)			
27		+Check_hoaccPwr(TCV_HoaccPara, TCV_Pwrlvl_ho)			
		localtree2(ch:LOGICCH)			
28		L?DL_RacInHoacc (TCV_Hrf := DL_RacInHoacc.msg.horf)	HndOvAccRcv(ch, HandOverAcc_01)		
29	L_0251	[TCV_Hrf <> TSPX_HoRefH]		(F)	
30		(TCV_Cnt := TCV_Cnt + 1)			
31	L_0252	[TCV_Hrf = TSPX_HoRefH]		(P)	
32		(TCV_Cnt := TCV_Cnt + 1)			
		local_HoCMD			
33		(TCV_Tchtype := '00001'B)			
34		[TSPC_PGSM OR TSPC_EGSM]			
35		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC, TSPX_TscC, C_NCC, C_BCC, C_arfcnB, TSPX_TCHcarrierB, TSPX_HoRefH, TCV_Pwrlvl_ho, Synchi(C_report_otd, C_pseudo_synchronised), ChMod_omit, RelTmdDif_01(TSPX_k2), TimingAdv_omit, CphMod_omit))		
36		[TSPC_DCS]			
37		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC, TSPX_TscC, C_NCC, C_BCC, C_arfcnBd, TSPX_TCHcarrierBd, TSPX_HoRefH, TCV_Pwrlvl_ho, Synchi(C_report_otd, C_pseudo_synchronised), ChMod_omit, RelTmdDif_01(TSPX_k2), TimingAdv_omit, CphMod_omit))		

Detailed Comments: 1. Cell A with BCCH, CCCH, SDCCH4 and a traffic channel.

	2. Cell B with BCCH, CCCH, SDCCH4 and a traffic channel. 3. In cell A the timing advance = TSPX_y2. 4. pseudo-synch, rot = 1, nc1, = 0, real time diffrenece = 2*TSPX_k2 + 10., handover referemce = TSPX_HoRefH. 5. The Mobile Time Difference = (2*TSPX_k2 + TSPX_y2) mod 2 097 152 with tolerance of 2. 6. The time advance is correct.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_7
Group: GSM_L3_MS_v4220/RR/
Purpose: To verify that when the MS is ordered to make a non-synchronised handover to another cell and is ordered to report on the time difference between the cells, that it does so correctly.
Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcD, C_Full)			
3		(TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TSPX_y3), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(TSPX_y3), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+StartCellB_ho(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(20), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_9, C_PLMN_2, C_LAC_spe, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k3, 0)			2.
7		+FullRateCh_B_1(C_Ass, TSPX_TmSltC, TSPX_TscC, TCV_ChMod, FreqTCH(TSPX_TCHcarrierB), FreqTCH(TSPX_TCHcarrierBd), TimingAdv(20), C_BABR_0, C_cch_1Comb, C_BPM_3)			
8		(TCV_chTch := C_FACCHF_A_1, TCV_chTch1 := C_FACCHF_B_1, TCV_sacchTch1 := C_SACCHF_B_1)			
9		+PreEnterCCstateU10_r01(TimingAdv(TSPX_y3), TSPX_SDCCH4SubDef, 0, 3, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			3.
10	body	(TCV_Null := OM_SendNextOnHO(TCV_chTch1, 2))			
11		ACTIVATE(RcvHdOvAcc, OtherEventsFail_01)			
12		+local_handoverCMD			4.
13		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_chTch1, TimingAdv(20))		
14		(TCV_L1Head := OM_GetL1Hd(TCV_sacchTch1))			
15		+localtree(TCV_chTch1)			
16		+localtree1			

		localtree1			
17		L?DL_EstIn	DLEstInd(TCV_chTch1)		
18		L?DL_DatInHoCom (TCV_Td := DL_DatInHoCom.msg.motdif.value)	HndOvCmpRcv(TCV_chTch1, HandOverCmp_01)		
19		+CheckMTDif(TCV_Td, TSPX_y3, TSPX_k3)			
20		[TCV_L1Head.ta = '0010100'B]			
21		+PostMainLinkRel(TCV_chTch1)			
22	L_0257	[TCV_L1Head.ta <> '0010100'B]		(F)	
23		+PostMainLinkRel(TCV_chTch1)			
		localtree(ch:LOGICCH)			
24		L?DL_RacInHoacc (TCV_Hrf := DL_RacInHoacc.msg.horf)	HndOvAccRcv(ch, HandOverAcc_01)		
25	L_0258	[TCV_Hrf <> TSPX_HoRefB]		(F)	
26	L_0259	[TCV_Hrf = TSPX_HoRefB]		(P)	
		local_handoverCMD			
27		(TCV_Tchtype := '00001'B)			
28		[TSPC_PGSM OR TSPC_EGSM]			
29		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC, TSPX_TscC, C_NCC, C_BCC, C_arfcnB, TSPX_TCHcarrierB, TSPX_HoRefB, TCV_Pwrlvl_ho, Synchi(C_report_otd, C_non_synchronised) , ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
30		[TSPC_DCS]			
31		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC, TSPX_TscC, C_NCC, C_BCC, C_arfcnBd, TSPX_TCHcarrierBd, TSPX_HoRefB, TCV_Pwrlvl_ho, Synchi(C_report_otd, C_non_synchronised) , ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		

Detailed Comments:

1. Cell A with BCCH, CCCH, SDCCH4 and a traffic channel.
2. Cell B with BCCH, CCCH, SDCCH4 and a traffic channel.
3. In cell A the timing advance = TSPX_y3.
4. non-synch, rot = 1, nci = 0, handover reference = TSPX_HoRefB.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_8
Group: GSM_L3_MS_v4220/RR/
Purpose: To verify the function of timer T3124 and the contents in the message HANDOVER FAILURE
Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcE, TSPX_MTChRateE)			
3		(TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubC, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+CCConfigTCH_pwr(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH(C_arfcn_tchA), FreqTCH(C_arfcn_tchAd), TSPX_MSTxpwrMax)			
6		+StartCellB(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), 5, 1, 0, 0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			2.
7		+FullRateCh_B_1(C_Ass, TSPX_TmSltC, TSPX_TscB, ChMod_speech, FreqTCH(TSPX_TCHcarrierB), FreqTCH(TSPX_TCHcarrierBd), TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
8		(TCV_chTch := C_FACCHF_A_1, TCV_sacchTch := C_SACCHF_A_1, TCV_chTch1 := C_FACCHF_B_1)			
9		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubC, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			3.
10		(TCV_L1Head0 := OM_GetL1Hd(TCV_sacchTch))			
11	body	ACTIVATE(RcvHdOvAcc, OtherEventsFail_01)			
12		+gsmOrDcs			4.
13		+localtree(TCV_chTch1)			
14		+ltree_continue			
		ltree_continue			
15		L?DL_EstIn	DLEstInd(TCV_chTch)		
16		(TCV_L1Head := OM_GetL1Hd(TCV_sacchTch))			
17		(TCV_Fn := OM_ReturnFn(TCV_chTch))			

18		L?DL_DatInHofl (TCV_Fn1 := DL_DatInHofl.fn)	HndOvFlRcv(TCV_chTch, HandOvFail_02)		5.
19		+Check_Time(3000)			
20	L_0260	[TCV_L1Head.mspwrlvl = TCV_L1Head0.mspwrlvl]		(P)	6.
21		+PostMainLinkRel(TCV_chTch)			
22	L_0261	[TCV_L1Head.mspwrlvl <> TCV_L1Head0.mspwrlvl]		(F)	
23		+PostMainLinkRel(TCV_chTch)			
		localtree(ch:LOGICCH)			
24		(TCV_Cnt := 0)			
25		REPEAT localtree1(ch) UNTIL [TCV_Cnt = 3]			
26		(TCV_HoaccPara := OM_GetHoaccPara(ch))			
27		+Check_hoaccPwr(TCV_HoaccPara, '01000'B)			pwrlvl = 8
		localtree1(ch:LOGICCH)			
28		L?DL_RacInHoacc (TCV_Hrf := DL_RacInHoacc.msg.horf)	HndOvAccRcv(ch, HandOverAcc_01)		
29	L_0263	[TCV_Hrf <> TSPX_HoRefC]		(F)	
30		(TCV_Cnt := TCV_Cnt + 1)			
31	L_0264	[TCV_Hrf = TSPX_HoRefC]		(P)	
32		(TCV_Cnt := TCV_Cnt + 1)			
		gsmOrDcs			
33		(TCV_Tchtype := '00001'B)			
34		[TSPC_PGSM OR TSPC_EGSM]			
35		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC, TSPX_TscB, C_NCC, C_BCC, C_arfcnB, TSPX_TCHcarrierB, TSPX_HoRefC, TCV_Pwrlvl_ho, Synchi(C_report_otd, C_non_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
36		[TSPC_DCS]			
37		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC, TSPX_TscB, C_NCC, C_BCC, C_arfcnBd, TSPX_TCHcarrierBd, TSPX_HoRefC, TCV_Pwrlvl_ho, Synchi(C_report_otd, C_non_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		

Detailed Comments:

1. To setup BCCH, CCCH, SDCCH4 and a full rate traffic channel for cell A.
2. To setup BCCH, CCCH, SDCCH4 and a full rate traffic channel for cell B.
3. To bring the MS into the U10 state.
4. Non-synchronised handover, power level = 8.
5. On old channel.
6. Power level is the old one (TSPX_MSTxpwrMax)

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_5_9
Group: GSM_L3_MS_v4220/RR/
Purpose: To verify the function of timer T3124 and the contents in the message HANDOVER FAILURE
Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcF, TSPX_MTChRateF)			
3		(TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+CCConfigTCH_pwr(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH(C_arfcn_tchA), FreqTCH(C_arfcn_tchAd), TSPX_MSTxpwrMax)			
6		+StartCellB(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			
7		+FullRateCh_B_1(C_Ass, TSPX_TmSltC, TSPX_TscC, ChMod_speech,FreqTCH(TSPX_TCHcarrierB), FreqTCH(TSPX_TCHcarrierBd), TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
8		(TCV_chTch := C_FACCHF_A_1, TCV_sacchTch := C_SACCHF_A_1, TCV_chTch1 := C_FACCHF_B_1)			
9		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubDef, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			3.
10		(TCV_L1Head0 := OM_GetL1Hd(TCV_sacchTch))			
11	body	ACTIVATE(RcvHdOvAcc, OtherEventsFail_01)			
12		+gsmOrDcs			4.
13		+localtree(TCV_chTch1)			
14		+ltree_continue			
		ltree_continue			
15		L?DL_EstIn	DLEstInd(TCV_chTch)		

16		(TCV_L1Head := OM_GetL1Hd(TCV_sacchTch))			
17		(TCV_Fn := OM_ReturnFn(TCV_chTch))			
18		L?DL_DatInHofl (TCV_Fn1 := DL_DatInHofl.fn)	HndOvFlRcv(TCV_chTch, HandOvFail_02)		5.
19		+Check_Time(3000)			
20	L_0267	[TCV_L1Head.mspwrlvl = TCV_L1Head0.mspwrlvl]		(P)	6.
21		+PostMainLinkRel(TCV_chTch)			
22	L_0268	[TCV_L1Head.mspwrlvl <> TCV_L1Head0.mspwrlvl]		(F)	
23		+PostMainLinkRel(TCV_chTch)			
		localtree(ch:LOGICCH)			
24		(TCV_Cnt := 0)			
25		REPEAT localtree1(ch) UNTIL [TCV_Cnt = 2]			
26		(TCV_HoaccPara := OM_GetHoaccPara(ch))			
27		+Check_hoaccPwr(TCV_HoaccPara, TCV_Pwrlvl_ho)			
		localtree1(ch:LOGICCH)			
28		L?DL_RacInHoacc (TCV_Hrf := DL_RacInHoacc.msg.horf)	HndOvAccRcv(ch, HandOverAcc_01)		
29	L_0270	[TCV_Hrf <> TSPX_HoRefC]		(F)	
30		(TCV_Cnt := TCV_Cnt + 1)			
31	L_0271	[TCV_Hrf = TSPX_HoRefC]		(P)	
32		(TCV_Cnt := TCV_Cnt + 1)			
		gsmOrDcs			
33		(TCV_Tchtype := '00001'B)			
34		[TSPC_PGSM OR TSPC_EGSM]			
35		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC, TSPX_TscC, C_NCC, C_BCC, C_arfcnB, TSPX_TCHcarrierB, TSPX_HoRefC, TCV_Pwrlvl_ho, Synchi(C_report_otd, C_non_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
36		[TSPC_DCS]			
37		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC, TSPX_TscC, C_NCC, C_BCC, C_arfcnBd, TSPX_TCHcarrierBd, TSPX_HoRefC, TCV_Pwrlvl_ho, Synchi(C_report_otd, C_non_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		

Detailed Comments:

1. To setup BCCH, CCCH, SDCCH4 and a full rate traffic channel for cell A.
2. To setup BCCH, CCCH, SDCCH4 and a receiving only full rate traffic channel for cell B.
3. To bring the MS into the U10 state.
4. Non-synchronised handover, power level = 8.
5. On old channel.

6. Power level is the old one (TSPX_MSTxpwrMax)

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_6_1
Group: GSM_L3_MS_v4220/RR/
Purpose: To verify that the MS, after receiving a Frequency Redefinition message, starts using the new frequencies and hopping sequence at the time indicated in the message.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(700)			
2		+test1			
3		+test2			
4		+test3			
5		+test4			
6		+test5			
7		+test6			
8		[TSPC_DualRate]			
9		+test7			
10		+test8			
11		+test9			
12		[NOT TSPC_DualRate]			
		test1			
13		+sdcch8(Freq_rg18, Freq_rd18, TSPX_TmSltA, TSPX_TscA, Ca2_g01, Ca3_g01, Ca2_d01, Ca3_d01, Ma1_g01, Ma2_g01, Ma3_g01, 6, 7, 91, ChDescrp_sdcch8_fh(TSPX_TmSltA, TSPX_TscA, TSPX_SDCCH8SubA, '001001'B, '000000'B))			1. R=1, K=1
		test2			
14		+sdcch8(Freq_rg19, Freq_rd19, TSPX_TmSltB, TSPX_TscB, Ca2_g02, Ca3_g02, Ca2_d02, Ca3_d02, Ma1_g02, Ma2_g02, Ma3_g02, 2, 1, 42000, ChDescrp_sdcch8_fh(TSPX_TmSltB, TSPX_TscB, TSPX_SDCCH8SubB, '000100'B, '000000'B))			R=1, K=2
		test3			
15		+sdcch8(Freq_rg20, Freq_rd20, TSPX_TmSltC, TSPX_TscC, Ca2_g03, Ca3_g03, Ca2_d03, Ca3_d03, Ma1_g03, Ma2_g03, Ma3_g03, 1, 3, 1000, ChDescrp_sdcch8_fh(TSPX_TmSltC, TSPX_TscC, TSPX_SDCCH8SubC, '000010'B, '000000'B))			R=1, K=3
		test4			
16		+tchf(Freq_rg21, Freq_rd21, TSPX_TmSltD, TSPX_TscD, Ca2_g04, Ca3_g04, Ca2_d04, Ca3_d04, Ma1_g04, Ma2_g04, Ma3_g04, 2, 3, 91, ChDescrp_tchf_fh(TSPX_TmSltD, TSPX_TscD, '000110'B, '000000'B))			2. R=2, K=1
		test5			
17		+tchf(Freq_rg22, Freq_rd22, TSPX_TmSltE, TSPX_TscE, Ca2_g05, Ca3_g05, Ca2_d05, Ca3_d05, Ma1_g05, Ma2_g05, Ma3_g05, 4, 2, 42000, ChDescrp_tchf_fh(TSPX_TmSltE, TSPX_TscE, '000101'B, '000000'B))			R=2, K=2
		test6			
18		+tchf(Freq_rg23, Freq_rd23, TSPX_TmSltF, TSPX_TscF, Ca2_g06, Ca3_g06, Ca2_d06, Ca3_d06, Ma1_g06, Ma2_g06, Ma3_g06, 6, 4, 15000, ChDescrp_tchf_fh(TSPX_TmSltF, TSPX_TscF, '001000'B, '000000'B))			R=2, K=3
		test7			
19		+tchh(Freq_rg24, Freq_rd24, TSPX_TmSltG, TSPX_TscG, Ca2_g07, Ca3_g07, Ca2_d07, Ca3_d07, Ma1_g07, Ma2_g07, Ma3_g07, 7, 4, 91, ChDescrp_tchh_fh(TSPX_TmSltG, TSPX_TscG, TSPX_TCHHSubDef, '000011'B, '000000'B))			3. R=3, K=1

		test8			
20		+tchh(Freq_rg25, Freq_rd25, TSPX_TmSltA, TSPX_TscA, Ca2_g08, Ca3_g08, Ca2_d08, Ca3_d08, Ma1_g08, Ma2_g08, Ma3_g08, 2, 5, 42000, ChDescrp_tchh_fh(TSPX_TmSltA, TSPX_TscA, TSPX_TCHHSubDef, '000111'B, '000000'B))			R=3, K=2
		test9			
21		+tchh(Freq_rg26, Freq_rd26, TSPX_TmSltDef, TSPX_TscDef, Ca2_g09, Ca3_g09, Ca2_d09, Ca3_d09, Ma1_g09, Ma2_g09, Ma3_g09, 5, 3, 4000, ChDescrp_tchh_fh(TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, '000001'B, '000000'B))			R=3, K=3
		test(Ca2, Ca3:CCHD; Ma1, Ma2, Ma3: MA; Maio2, Maio3, tm:INTEGER; Chd: CHD)			
22		+ltree_immass(Ma1, Chd)			
23		+Check(TCV_cchd1, Ma1, tm, Chd)			
24		+ltree_freqRedef(Ca2, Ma2, Maio2, tm, Chd)			
25		+ltree_freqRedef(Ca3, Ma3, Maio3, tm, Chd)			
26		+ltree_chkAndrelease(Ca3, Ma3, tm, Chd)			
		ltree_freqRedef(ca: CCHD; ma: MA; maio, tm:INTEGER; chd: CHD)			
27		[NOT TCV_Res]			
28		[TCV_Res]			
29		+ltree_fredef(ca, ma, maio, tm, chd)			
30		+Check(ca, ma, tm, chd)			
		ltree_chkAndrelease(ca: CCHD; ma: MA; tm:INTEGER; chd: CHD)			
31		[NOT TCV_Res]			
32		(TCV_Null := OM_StopCell(C_CellA))			
33		[TCV_Res]			
34		+PostMainLinkRel(TCV_chTch)			
35		(TCV_Null := OM_StopCell(C_CellA))			
		Check(ca: CCHD; ma: MA; tm:INTEGER; chd: CHD)			
36		(TCV_Res := OM_FHCHK(TCV_sacchTch, ca, ma, chd, tm, TCV_Fn))			
37	L_0274	[NOT TCV_Res]		(F)	
38		+PostMainLinkRel(TCV_chTch)			
39	L_0275	[TCV_Res]		(P)	
		ltree_fredef(cchd: CCHD; ma:MA; maio, tm:INTEGER; chd: CHD)			
40		(TCV_Fn := OM_ComingFn(TCV_chTch), TCV_chd1 := chd, TCV_chd1.maio := INT_TO_BIT(maio, 6), TCV_Strt := OC_StartTime(TCV_Fn, tm, 0), TCV_Null := OM_SendNextOn(TCV_chTch, TCV_Fn), TCV_Null := OM_FreqDef(TCV_Strt, ma, TCV_chTch, TCV_chd1, cchd, TCV_Fn))			
41		L!DL_DatRqFrqre	FrqRedfSnd(TCV_chTch, FreqRedef_01(TCV_chd1, ma, TCV_Strt, cchd))		
		sdcch8(FqParag, FqParad:FRQPARA; Tmslt:SN; Tsc:TSC; Cag2, Cag3, Cad2, Cad3:CCHD; Ma1, Ma2, Ma3: MA; Maio2, Maio3, tm:INTEGER; Chd: CHD)			
42		+ltree_idleupdated(FqParag, FqParad, Cag2, Cag3, Cad2, Cad3)			
43		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubA, C_Immass, Tmslt, Tsc, ChMod_sign, FqParag, FqParad, TimingAdv(30), C_BABR_0,			

		C_cch_1nonComb, C_BPM_3)			
44		(TCV_chTch := TCV_ch, TCV_sacchTch := TCV_sacch8)			
45		+test(TCV_CAgsm, TCV_CAdcs, Ma1, Ma2, Ma3, Maio2, Maio3, tm, Chd)			
		tchf(FqParag, FqParad:FRQPARA; Tmslt:SN; Tsc:TSC; Cag2, Cag3, Cad2, Cad3:CCHD; Ma1, Ma2, Ma3: MA; Maio2, Maio3, tm:INTEGER; Chd: CHD)			
46		+ltree_idleupdated(FqParag, FqParad, Cag2, Cag3, Cad2, Cad3)			
47		+FullRateCh_A_1_nociph(C_Immass, Tmslt, Tsc, ChMod_sign, FqParag, FqParad, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
48		+test(TCV_CAgsm, TCV_CAdcs, Ma1, Ma2, Ma3, Maio2, Maio3, tm, Chd)			
		tchh(FqParag, FqParad:FRQPARA; Tmslt:SN; Tsc:TSC; Cag2, Cag3, Cad2, Cad3:CCHD; Ma1, Ma2, Ma3: MA; Maio2, Maio3, tm:INTEGER; Chd: CHD)			
49		+ltree_idleupdated(FqParag, FqParad, Cag2, Cag3, Cad2, Cad3)			
50		+HalfRateCh_A_1_nociph(TSPX_TCHHSubDef, C_Immass, Tmslt, Tsc, ChMod_sign, FqParag, FqParad, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
51		+test(TCV_CAgsm, TCV_CAdcs, Ma1, Ma2, Ma3, Maio2, Maio3, tm, Chd)			
		ltree_immass(ma1:MA; chd:CHD)			
52		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
53		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
54		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
55		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_02(TCV_Rr, TCV_Fn, TimingAdv(30), chd, ma1))		
56		L?DL_EstInPgRes(TCV_Fn := DL_EstInPgRes.fn)	PagingRes(PagingRes_01)		
57		ACTIVATE(OtherEventsFail)			Restore Normal default
		ltree_idleupdated(FqPg, FqPd:FRQPARA; Cag2, Cag3, Cad2, Cad3:CCHD)			
58		(TCV_CAgsm := Ca1_g01, TCV_freq := FqPg, TCV_CAgsm.rfl := TCV_freq.flst.fl, TCV_CAdcs := Ca1_g01, TCV_freq := FqPd, TCV_CAdcs.rfl := TCV_freq.flst.fl)			
59		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, TCV_CAgsm, TCV_CAdcs, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
60		[TSPC_PGSM OR TSPC_EGSM]			
61		(TCV_cchd1 := TCV_CAgsm, TCV_CAgsm := Cag2, TCV_CAdcs := Cag3)			4.
62		[TSPC_DCS]			
63		(TCV_cchd1 := TCV_CAdcs, TCV_CAgsm :=			5.

		Cad2, TCV_CAdcs := Cad3)			

Detailed Comments:

1. To perform test on SDCCH channel.
2. To perform test on TCH/F channel.
3. To perform test on TCH/H channel.
4. TCV_CAgsm, TCV_CAdcs contain two cell allocation values for GSM.
5. TCV_CAgsm, TCV_CAdcs contain two cell allocation values for DCS.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_7_1

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS, in an RR connected state, acknowledging a CHANNEL MODE MODIFY message by sending a CHANNEL MODE MODIFY ACKNOWLEDGEMENT message specifying and switching to the correct mode.

- the new mode if that mode is supported
- the old mode if the new mode is not supported.

This shall be verified for the channel modes

- signalling only
- speech full rate
- data 9.6 Kb/s
- data 4.8 Kb/s full rate
- data 2.4 Kb/s full rate.

Default: OtherEventsFail

Comments: apply only to the MS supporting TCH/F.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+gsmOrDcs			
4		+FullRateCh_A_1_nociph(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH(C_arfcn_tchA), FreqTCH(C_arfcn_tchAd), TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
5		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_05)		
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r13(TCV_Rr, TCV_Fn, TCV_chd1, TimingAdv(30)))		2.
9		L?DL_EstInPgRes	PagingRes(PagingRes_r02)		
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11	body	L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(ChMod_speech, TCV_chd1))		
12		[TSPC_Serv_TS11 OR TSPC_Serv_TS12]			
13	L_0276	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_speech, TCV_chd1))	(P)	
14		+localtree(ChMod_speech)			
15		[(NOT TSPC_Serv_TS11) AND(NOT TSPC_Serv_TS12)]			
16	L_0277	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch,	(P)	

			ChmomoAck_01(ChMod_sign, TCV_chd1))		
17		+localtree(ChMod_sign)			
		localtree(chm:CHMOD)			
18		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(ChMod_12k, TCV_chd1))		
19		[TSPC_96Data]			
20	L_0278	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_12k, TCV_chd1))	(P)	
21		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(ChMod_6k, TCV_chd1))		
22		[TSPC_48DataF]			
23		+localtree2			
24		[NOT TSPC_48DataF]			
25	L_0279	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_12k, TCV_chd1))	(P)	
26		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(ChMod_3k, TCV_chd1))		
27		[TSPC_24DataF]			
28	L_0280	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_3k, TCV_chd1))	(P)	
29		+localtree1			
30		[NOT TSPC_24DataF]			
31	L_0281	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_12k, TCV_chd1))	(P)	
32		+localtree1			
33		[NOT TSPC_96Data]			
34	L_0282	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(chm, TCV_chd1))	(P)	
35		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(ChMod_6k, TCV_chd1))		
36		[TSPC_48DataF]			
37		+localtree2			
38		[NOT TSPC_48DataF]			
39	L_0283	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(chm, TCV_chd1))	(P)	
40		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(ChMod_3k, TCV_chd1))		
41		[TSPC_24DataF]			

42	L_0284	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_3k, TCV_chd1))	(P)	
43		+localtree1			
44		[NOT TSPC_24DataF]			
45	L_0285	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(chm, TCV_chd1))	(P)	
46		+localtree1			
		localtree1			
47		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(ChMod_sign, TCV_chd1))		
48	L_0286	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_sign, TCV_chd1))	(P)	
49		+PostMainLinkRel(TCV_chTch)			
		localtree2			
50	L_0287	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_6k, TCV_chd1))	(P)	
51		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(ChMod_3k, TCV_chd1))		
52		[TSPC_24DataF]			
53	L_0288	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_3k, TCV_chd1))	(P)	
54		+localtree1			
55		[NOT TSPC_24DataF]			
56	L_0289	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_6k, TCV_chd1))	(P)	
57		+localtree1			
		gsmOrDcs			
58		[TSPC_PGSM OR TSPC_EGSM]			
59		(TCV_chd1 := ChDescrp_nfh('00001'B, TSPX_TmSltDef, TSPX_TscDef, C_arfcn_tchA))			
60		[TSPC_DCS]			
61		(TCV_chd1 := ChDescrp_nfh('00001'B, TSPX_TmSltDef, TSPX_TscDef, C_arfcn_tchAd))			

Detailed Comments: 1. Default system infirmations for RR testing.
2. TCH/F channel.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_7_2

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS, in an RR connected state, acknowledges a CHANNEL MODE MODIFY message by sending a CHANNEL MODE MODIFY ACKNOWLEDGEMENT message specifying and switches to the correct mode

- the new mode if that mode is supported
- the old mode if the new mode is not supported.

This shall be verified for the channel modes

- signalling only
- speech half rate
- data 4.8 Kb/s half rate
- data 2.4 Kb/s half rate

Default: OtherEventsFail

Comments: apply only to the MS supporting TCH/H.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+gsmOrDcs			
4		+HalfRateCh_A_1_nociph(TSPX_TCHHSubA, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(C_arfcn_tchA), FreqTCH(C_arfcn_tchAd), TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
5		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_05)		
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r14(TCV_Rr, TCV_Fn, TCV_chd1, TimingAdv(30)))		2.
9		L?DL_EstInPgRes	PagingRes(PagingRes_r02)		
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_02(ChMod_speech, TCV_chd1))		
12	L_0290	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_02(ChMod_speech, TCV_chd1))	(P)	
13		[TSPC_HalfRateSpeech]			
14	L_0291	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_02(ChMod_speech, TCV_chd1))	(P)	
15		+localtree(ChMod_speech)			

16		[NOT TSPC_HalfRateSpeech]			
17	L_0292	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_02(ChMod_sign, TCV_chd1))	(P)	
18		+localtree(ChMod_sign)			
		localtree(chm:CHMOD)			
19		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_02(ChMod_6k, TCV_chd1))		
20		[TSPC_48DataH]			
21		+localtree2			
22		[NOT TSPC_48DataH]			
23	L_0293	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_02(chm, TCV_chd1))	(P)	
24		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_02(ChMod_3k, TCV_chd1))		
25		[TSPC_24DataH]			
26	L_0294	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_02(ChMod_3k, TCV_chd1))	(P)	
27		+localtree1			
28		[NOT TSPC_24DataH]			
29	L_0295	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_02(chm, TCV_chd1))	(P)	
30		+localtree1			
		localtree1			
31		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_02(ChMod_sign, TCV_chd1))		
32	L_0296	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_02(ChMod_sign, TCV_chd1))	(P)	
33		+PostMainLinkRel(TCV_chTch)			
		localtree2			
34	L_0297	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_02(ChMod_6k, TCV_chd1))	(P)	
35		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_02(ChMod_3k, TCV_chd1))		
36		[TSPC_24DataH]			
37	L_0298	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_02(ChMod_3k, TCV_chd1))	(P)	
38		+localtree1			

39		[NOT TSPC_24DataH]			
40	L_0299	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_02(ChMod_6k, TCV_chd1))	(P)	
41		+localtree1			
		gsmOrDcs			
42		[TSPC_PGSM OR TSPC_EGSM]			
43		(TCV_chd1 := ChDescrp_tchh_nfh(TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubA, C_arfcn_tchA))			
44		[TSPC_DCS]			
45		(TCV_chd1 := ChDescrp_tchh_nfh(TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubA, C_arfcn_tchAd))			

Detailed Comments: 1. Default system infirmations for RR testing.
2. TCH/H channel

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_8_1
Group: GSM_L3_MS_v4220/RR/
Purpose: To verify that the MS starts ciphering when it receives a CIPHERING MODE COMMAND message with Cipher Mode Setting = "Start Ciphering". To verify that it continues to use the old cipher key after it receives an AUTHENTICATION REQUEST whilst in ciphered mode.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcH, TSPX_MOChRateH)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_05, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_02, BcchFreqLst_01, BcchFreqLst_04, BcchFreqLst_48, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNA, TSPX_RANDA, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4	body	[TSPC_Feat_A51]			
5		(TCV_CphAlg := '000'B)			
6		+InitCall(TCV_Service)			
7		+localtree			2.
8		[TSPC_Feat_A52]			
9		(TCV_CphAlg := '001'B)			
10		+InitCall(TCV_Service)			
11		+localtree			3.
12		[NOT TSPC_Feat_A52]			
13		[NOT TSPC_Feat_A51]			
14		[TSPC_Feat_A52]			
15		(TCV_CphAlg := '001'B)			
16		+InitCall(TCV_Service)			
17		+localtree			3.
18		[NOT TSPC_Feat_A52]			
		localtree			
19		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
20		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
21		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
22		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
23		ACTIVATE(OtherEventsFail)			Restore Normal default
24		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNA, TSPX_RANDA))		
25		L?DL_DatInAuthRes	AuthRes(AuthResponse)		
26		(TCV_CphMd.algid := TCV_CphAlg, TCV_Null := OM_CphMdChg(TCV_ch, TCV_CphMd, TCV_CphKey))			
27		L!DL_DatRqCphmCmd (DL_DatRqCphmCmd.msg.cphms.algid := TCV_CphAlg)	CphCmd(TCV_ch, CphModeCmd_01)		4.

28	L_0300	L?DL_DatInCphmCom	CphCom(CphModeCmp_01)	(P)	
29		L?DL_DatInSetup	SetupRcv(SetupInd_01)		
30		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNA, TSPX_RANDA))		
31	L_0301	L?DL_DatInAuthRes	AuthRes(AuthResponse)	(P)	
32		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. 1 cell with default parameters except Radio-Link-Timeout = 64.
2. Test the A5/1 algorithm.
3. Test the A5/2 algorithm.
4. Ciphering mode setting = "Start ciphering".

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_8_2
Group: GSM_L3_MS_v4220/RR/
Purpose: To verify that the MS does not start ciphering when it receives a CIPHERING MODE COMMAND message with Cipher Mode Setting = "No Ciphering".
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcC, TSPX_MOChRateC)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_05, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_02, BcchFreqLst_01, BcchFreqLst_04, BcchFreqLst_48, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4	body	+InitCall(TCV_Service)			
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubA, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
8		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNDef, TSPX_RANDDef))		
11		L?DL_DatInAuthRes	AuthRes(AuthResponse)		
12		L!DL_DatRqCphmCmd	CphCmd(TCV_ch, CphModeCmd_02)		2.
13		L?DL_DatInCphmCom	CphCom(CphModeCmp_01)		
14	L_0302	L?DL_DatInSetup	SetupRcv(SetupInd_01)	(P)	
15		+PostMainLinkRel(TCV_ch)			

Detailed Comments:
1. 1 cell with default parameters except Radio-Link-Timeout = 64.
2. Ciphering mode setting = "no ciphering".

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_8_3
Group: GSM_L3_MS_v4220/RR/
Purpose: To verify that the MS uses the stored cipher key when it receives a CIPHERING MODE COMMAND without a preceding authentication procedure.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcB, TSPX_MOChRateB)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_05, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_02, BcchFreqLst_01, BcchFreqLst_04, BcchFreqLst_48, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubB, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4	body	+InitCall(TCV_Service)			
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubB, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
8		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		(TCV_CphMd.algid := TSPX_CphAlgA)			2.
11		(TCV_Null := OM_CphMdChg(TCV_ch, TCV_CphMd, TCV_CphKey))			
12		L!DL_DatRqCphmCmd (DL_DatRqCphmCmd.msg.cphms.algid := TSPX_CphAlgA)	CphCmd(TCV_ch, CphModeCmd_01)		3.
13	L_0303	L?DL_DatInCphmCom	CphCom(CphModeCmp_01)	(P)	
14	L_0304	L?DL_DatInSetup	SetupRcv(SetupInd_01)	(P)	
15		+PostMainLinkRel(TCV_ch)			

Detailed Comments:
1. 1 cell with default parameters except Radio-Link-Timeout = 64.
2. The cphering algorithm is chosen arbitrarily but controllable (TSPX_CphAlgA).
3. Ciphering mode setting = "Start ciphering", old stored Kc. (generated by TSPX_Ki and TPSX_RANDDef)

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_8_4

Group: GSM_L3_MS_v4220/RR/

Purpose:

1 To verify that when the MS is in the "not ciphered" mode and receives the CIPHERING MODE COMMAND message with Ciphering Mode Setting information element set to "start ciphering", the MS uses the cipher key stored in the SIM to start ciphering and deciphering with the algorithm indicated by the "algorithm identifier" field and that the MS responds with a CIPHERING MODE COMPLETE message.

2 To verify that the MS is ready to transmit the CIPHERING MODE COMPLETE message before 500ms after the end of the CIPHERING MODE COMMAND message.

3 To verify that when the MS receives an ASSIGNMENT COMMAND message containing a Cipher Mode Setting IE after receipt of a CIPHERING MODE COMMAND message, the MS shall perform the assignment, use the commanded mode and/or algorithm on the new channel, and not change the ciphering key.

4 To verify that when the MS receives a HANDOVER COMMAND message containing a Cipher Mode Setting IE after receipt of a CIPHERING MODE COMMAND message, the MS shall perform the handover, use the commanded mode and/or algorithm on the new channel, and not change the ciphering key

5 To verify that when the MS is in the "ciphered" mode and receives the CIPHERING MODE COMMAND message with Cipher Mode Setting IE set to "no ciphering", the MS loads the cipher key stored in the SIM into the ME, stops ciphering and deciphering and, responds with a CIPHERING MODE COMPLETE message.

6 To verify that the MS responds to an AUTHENTICATION REQUEST message with an AUTHENTICATION RESPONSE message and continues to use the cipher key obtained from the previous authentication procedure.

7 To verify that when the MS is in the "not ciphered" mode and receives the CIPHERING MODE COMMAND message with Ciphering Mode Setting information element set to "no ciphering", the does not start ciphering or deciphering, but does respond with a CIPHERING MODE COMPLETE message.

8 To verify that when the MS receives a HANDOVER COMMAND message and the handover fails, the MS sends a HANDOVER FAILURE message on the old channel using the old ciphering mode and (if ciphered) the old algorithm and old key.

9 To verify that when the MS receives an ASSIGNMENT COMMAND message and the assignment fails, the MS sends an ASSIGNMENT FAILURE message on the old channel using the old ciphering mode and (if ciphered) the old algorithm and old key.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNB, TSPX_RANDB, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		(TCV_ch1 := OC_SubchOfSdcch4(TSPX_SDCCH4SubB, C_CellA), TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+ltree_body			
		ltree_body			
5		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		

6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		+ltree_ImmAss			
9		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		+localtree1			
12		+localtree2			
13		+localtree3			
14		[(TSPC_Feat_A51 AND(NOT TSPC_Feat_A52)) OR((NOT TSPC_Feat_A51) AND TSPC_Feat_A52)]			
15		+PostMainLinkRel(TCV_ch)			
16		[TSPC_Feat_A51 AND TSPC_Feat_A52]			
17		+localtree4			
		localtree1			
18		(TCV_Null := OM_CphMdChg(TCV_ch, CphMod_04(TSPX_CphAlgA), TCV_CphKey))			
19		L!DL_DatRqCphmCmd (DL_DatRqCphmCmd.msg.cphms.algid := TSPX_CphAlgA)	CphCmd(TCV_ch, CphModeCmd_01)		
20		START T_dly1(500)			
21	L_0305	?TIMEOUT T_dly1		(F)	
22		+PostMainLinkRel(TCV_ch)			
23		L?DL_DatInCphmCom CANCEL T_dly1	CphCom(CphModeCmp_01)		
24		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNB, TSPX_RANDB))		2.
25		L?DL_DatInAuthRes	AuthRes(AuthResponse)		
26		+ltree_HndOv0			
27		+handoverAcc(TCV_ch1)			
28		L?DL_EstIn	DLEstInd(TCV_ch1)		
29	L_0306	L?DL_DatInHoCom	HndOvCmpRcv(TCV_ch1, HandOverCmp_01)	(P)	
		localtree2			
30		(TCV_ch := OC_SubchOfSdcch4(TSPX_SDCCH4SubC, C_CellA), TCV_Null := OM_CphMd(TCV_ch, CphMod_04(TSPX_CphAlgB), TCV_CphKey))			
31		+ltree_Asgn1			
32		+AssCh_complete(TCV_ch1,TCV_ch,TCV_AssCmd)			3.
33		L!DL_DatRqCphmCmd START T_dly1(500)	CphCmd(TCV_ch, CphModeCmd_02)		4.
34	L_0307	?TIMEOUT T_dly1		(F)	
35		+PostMainLinkRel(TCV_ch)			
36		L?DL_DatInCphmCom CANCEL T_dly1	CphCom(CphModeCmp_01)		
37		(TCV_ch1 := OC_SubchOfSdcch4(TSPX_SDCCH4SubDef, C_CellA), TCV_CphKey := OC_CphKeyGen(TSPX_Ki, TSPX_RANDB), TCV_Null := OM_CphMd(TCV_ch1, CphMod_04(TSPX_CphAlgB), TCV_CphKey))			
38		+ltree_HndOv1			5.
39		+handoverAcc(TCV_ch1)			
40		L?DL_EstIn	DLEstInd(TCV_ch1)		
41	L_0308	L?DL_DatInHoCom	HndOvCmpRcv(TCV_ch1,	(P)	

			HandOverCmp_01)			
42		L!DL_DatRqIdRq	IDReq(TCV_ch1, IDRequest_01('0010'B))			
43		L?DL_DatInIdRes	IDRes(IDResponse_01)			
44		(TCV_ch := OC_SubchOfSdcch4(TSPX_SDCCH4SubA, C_CellA), TCV_Null := OM_CphMd(TCV_ch, CphMod_04(TSPX_CphAlgB), TCV_CphKey))				
45		+ltree_HndOv2				6.
46		+handoverAcc(TCV_ch)				
47		L?DL_EstIn	DLEstInd(TCV_ch)			
48		L?DL_DatInHoCom	HndOvCmpRcv(TCV_ch, HandOverCmp_01)			
	localtree3					
49	(TCV_ch1 := OC_SubchOfSdcch4(TSPX_SDCCH4SubB, C_CellA), TCV_Null := OM_CphMd(TCV_ch1, CphMod_02, TCV_CphKey))					
50	+ltree_Asgn2					
51	+AssCh_complete(TCV_ch,TCV_ch1,TCV_AssCmd)					
52	L!DL_DatRqCphmCmd START T_dly1(500)		CphCmd(TCV_ch1, CphModeCmd_02)			
53	L_0309	?TIMEOUT T_dly1		(F)		
54		+PostMainLinkRel(TCV_ch1)				
55	L?DL_DatInCphmCom CANCEL T_dly1		CphCom(CphModeCmp_01)			
56	L!DL_DatRqAuthRq		AuthReq(TCV_ch1, AuthRequest(TSPX_CKSNC, TSPX_RANDC))			8.
57	L?DL_DatInAuthRes		AuthRes(AuthResponse)			
58	(TCV_ch := OC_SubchOfSdcch4(TSPX_SDCCH4SubC, C_CellA), TCV_Null := OM_CphMd(TCV_ch, CphMod_04(TSPX_CphAlgD), TCV_CphKey))					
59	+ltree_HndOv3					9.
60	+handoverAcc(TCV_ch)					
61	L?DL_EstIn		DLEstInd(TCV_ch)			
62	L_0310	L?DL_DatInHoCom	HndOvCmpRcv(TCV_ch, HandOverCmp_01)	(P)		
63	(TCV_ch1 := OC_SubchOfSdcch4(TSPX_SDCCH4SubB, C_CellA))					
64	+ltree_HndOv0					10.
65	L?DL_EstIn		DLEstInd(TCV_ch)			
66	L_0311	L?DL_DatInHofl	HndOvFlRcv(TCV_ch, HandOvFail_01)	(P)		
67	L!DL_DatRqIdRq		IDReq(TCV_ch, IDRequest_01('0010'B))			
68	L?DL_DatInIdRes		IDRes(IDResponse_01)			
69	+ltree_Asgn2					
70	+AssCh_failure(TCV_ch,TCV_AssCmd,TRUE)					

		localtree4			
71		(TCV_Null := OM_CphMd(TCV_ch1, CphMod_04(TSPX_CphAlgE), TCV_CphKey))			
72		+ltree_Asgn3			
73		+AssCh_complete(TCV_ch,TCV_ch1,TCV_AssCmd)			
74		(TCV_ch := OC_SubchOfSdcch4(TSPX_SDCCH4SubA, C_CellA), TCV_Null := OM_CphMd(TCV_ch, CphMod_04(TSPX_CphAlgE), TCV_CphKey))			
75		+ltree_HndOv4			
76		+handoverAcc(TCV_ch)			
77		L?DL_EstIn	DLEstInd(TCV_ch)		
78	L_0312	L?DL_DatInHoCom	HndOvCmpRcv(TCV_ch, HandOverCmp_01)	(P)	
79		+PostMainLinkRel(TCV_ch)			
		handoverAcc(ch:LOGICCH)			
80		L?DL_RacInHoacc	HndOvAccRcv(ch, HandOverAcc_01)		
81		L?DL_RacInHoacc	HndOvAccRcv(ch, HandOverAcc_01)		
82		L?DL_RacInHoacc	HndOvAccRcv(ch, HandOverAcc_01)		
83		L?DL_RacInHoacc	HndOvAccRcv(ch, HandOverAcc_01)		
		ltree_ImmAss			
84		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubA, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
		ltree_Asgn1			
85		(TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubC)), 5))			
86		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, 9, TCV_chdescr_arfcn, CellChDes_Omit, ChMod_omit, StartingTm_omit, CphMod_04iei(TSPX_CphAlgB)))			
		ltree_Asgn2			
87		(TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubB)), 5))			
88		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, 9, TCV_chdescr_arfcn, CellChDes_Omit, ChMod_omit, StartingTm_omit, CphMod_02iei))			
		ltree_Asgn3			
89		(TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubB)), 5))			
90		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, 9, TCV_chdescr_arfcn, CellChDes_Omit, ChMod_omit, StartingTm_omit, CphMod_04iei(TSPX_CphAlgE)))			
		ltree_HndOv0			
91		(TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubB)), 5))			
92		[TSPC_PGSM OR TSPC_EGSM]			
93		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC, C_arfcnA, C_arfcnA,		

			TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_02iei))		
94		[TSPC_DCS]			
95		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC, C_arfcnAd, C_arfcnAd, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_02iei))		
		ltree_HndOv1			
96		(TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubDef)), 5))			
97		[TSPC_PGSM OR TSPC_EGSM]			
98		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch1, HandOverCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC, C_arfcnA, C_arfcnA, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_04iei(TSPX_CphAlgC)))		
99		[TSPC_DCS]			
100		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch1, HandOverCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC, C_arfcnAd, C_arfcnAd, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_04iei(TSPX_CphAlgC)))		
		ltree_HndOv2			
101		(TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubA)), 5))			
102		[TSPC_PGSM OR TSPC_EGSM]			
103		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch1, HandOverCmd_nfh(		

			TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC, C_arfcnA, C_arfcnA, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_04iei(TSPX_CphAlgC)))		
104		[TSPC_DCS]				
105		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch1, HandOverCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC, C_arfcnAd, C_arfcnAd, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_04iei(TSPX_CphAlgC)))		
		ltree_HndOv3				
106		(TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubC)), 5))				
107		[TSPC_PGSM OR TSPC_EGSM]				
108		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch1, HandOverCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC, C_arfcnA, C_arfcnA, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_04iei(TSPX_CphAlgD)))		
109		[TSPC_DCS]				
110		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch1, HandOverCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC, C_arfcnAd, C_arfcnAd, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_04iei(TSPX_CphAlgD)))		
		ltree_HndOv4				

111		(TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubA)), 5))			
112		[TSPC_PGSM OR TSPC_EGSM]			
113		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch1, HandOverCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC, C_arfcnA, C_arfcnA, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_04iei(TSPX_CphAlgE)))		
114		[TSPC_DCS]			
115		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch1, HandOverCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC, C_arfcnAd, C_arfcnAd, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_04iei(TSPX_CphAlgE)))		

Detailed Comments:

1. 1 cell, default parameters.
2. New ciphering key sequence number and new ciphering key L.
3. New SDCCH4 subchannel different from the one in use, start ciphering.
4. Load new key L, no ciphering.
5. New SDCCH4 subchannel different from the one in use, start ciphering.
6. New SDCCH4 subchannel different from the one in use, start ciphering.
7. New SDCCH4 subchannel different from the one in use, no ciphering.
8. To generate new ciphering key M.
9. New SDCCH4 subchannel different from the one in use, start ciphering.
10. New SDCCH4 subchannel different from the one in use, no ciphering, new channel not activated.
11. New SDCCH4 subchannel different from the one in use, no ciphering, new channel not activated.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_8_5

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS supplies its IMEISV in the CIPHERING MODE COMPLETE message when it receives a CIPHERING MODE COMMAND message with a Cipher Response bit set to 'IMEISV shall be included'. To verify that the MS does not supply any Mobile Identity IE in the CIPHERING MODE COMPLETE message when it receives a CIPHERING MODE COMMAND message with a Cipher Response bit set to 'IMEISV shall not be included'.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubB, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
4		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
5		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
6		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubB, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		2.
7		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
8		ACTIVATE(OtherEventsFail)			Restore Normal default
9	body	L!DL_DatRqCphmCmd	CphCmd(TCV_ch, CphModeCmd_02)		
10	L_0313	L?DL_DatInCphmCom	CphCom(CphModeCmp_02)	(P)	
11		L!DL_DatRqCphmCmd	CphCmd(TCV_ch, CphModeCmd_03)		
12	L_0314	L?DL_DatInCphmCom	CphCom(CphModeCmp_03)	(P)	
13		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_11_1

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that if the RF power capability or any other capability indicated in a Classmark IE of the MS is changed during a call, the change is communicated on the DCCH to the network. To verify that if the RF power capability or any other capability indicated in a Classmark IE of the MS is changed in idle mode, the out of date capabilities are not communicated to the network during RR connection establishment.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcB, TSPX_MOChRateB)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubC, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5	body	(TCV_Null := OO_AddPwrAmp())			
6		+InitCall(TCV_Service)			
7		+subtree1			
8		(TCV_Null := OO_RemvPwrAmp())			
9		+subtree2			
		subtree1			
10		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
11		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
12		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubC, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
13	L_0315	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_02)	(P)	2.
14		ACTIVATE(OtherEventsFail)			Restore Normal default
15		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
16		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
17		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
18		L!DL_DatRqAlert	AlertSnd(TCV_ch, Alerting_01(TCV_TI))		
19		+continue			
		continue			
20		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
21		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
22		L!DL_DatRqConn	ConnSnd(TCV_chTch,		

			Connect_02(TCV_TI))		
23		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
		subtree2			
24	L_0316	L?DL_UdatInClmChn	ClassChgURcv(ClassChange_01)	(P)	5.
25		(TCV_Null := OO_AddPwrAmp())			
26	L_0317	L?DL_UdatInClmChn	ClassChgURcv(ClassChange_02)	(P)	6.
27		+PostMainLinkRel(TCV_chTch)			
28		+localtree3			
		localtree3			
29		(TCV_Null := OO_RemvPwrAmp())			
30		START T_dly(12000)			
31		?TIMEOUT T_dly			
32		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
33		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
34		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
35		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubC, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
36	L_0318	L?DL_EstInPgRes	PagingRes(PagingRes_03(TSPX_CKSNDef))	(P)	7.
37		ACTIVATE(OtherEventsFail)			Restore Normal default
38		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4.
2. The power capability is the one with RF amplification.
3. To setup a physical channel as full rate traffic channel for full rate bearer capability.
4. To setup a physical channel as half rate traffic channel for half rate bearer capability.
5. The power capability is the original one without RF amplification.
6. The power capability is the one with RF amplification.
7. The power capability is the original one without RF amplification.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_11_2

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that if the network requests the MS to supply all its classmark information then this information is communicated on the DCCH to the network.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_TmsiOff, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOn(C_SIMneedRmv)			
4		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
5		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
6		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubA, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
7		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_05)		
8		ACTIVATE(OtherEventsFail)			Restore Normal default
9		L!DL_DatRqClmEnq START T_dly1(300)	ClassMkEnq_01(TCV_ch)		
10	L_0319	L?DL_DatInClmChn CANCEL T_dly1	ClassChgDRcv(ClassChange_03)	(P)	2.
11		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1, C_PLMN_1, C_LAC_1))		
12		+PostMainLinkRel(TCV_ch)			
13	L_0320	?TIMEOUT T_dly1		(F)	3.
14		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To change the IMSI attach/detach flag to 1.
2. The expected CLASSMARK CHANGE message received before 300 ms after the CLASSMARK ENQUIRY, pass.
3. The expected message not received with 300 ms, fail.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_12_1
Group: GSM_L3_MS_v4220/RR/
Purpose: To verify that the MS is able to correctly release an SDCCH after having received a CHANNEL RELEASE message.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+Varinit_tch(C_arfcn_tchA, C_arfcn_tchAd)			
4		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubB, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(TCV_tch_arfcn), FreqTCH(TCV_tch_arfcn), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
5	body	+subtree1			
6		+subtree2			
		subtree1			
7		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
8		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
9		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
10		+local_immass			
11		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
12		ACTIVATE(OtherEventsFail)			Restore Normal default
13		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
14		L?DL_RelIn	DLRelInd_01		
15		START T_dly(3000)			
16	L_0321	L?OTHERWISE		(F)	
17	L_0322	?TIMEOUT T_dly		(P)	
		subtree2			
18		START T_dly(12000)			
19		?TIMEOUT T_dly			
20		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
21		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
22		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
23		+local_immass			
24	L_0323	L?DL_EstInPgRes	PagingRes(PagingRes_01)	(P)	
25		ACTIVATE(OtherEventsFail)			Restore Normal default
26		+PostMainLinkRel(TCV_ch)			
		local_immass			
27		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r02(		

			TCV_Rr, TCV_Fn, TSPX_SDCCH8SubB, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), TCV_tch_arfcn))		
Detailed Comments:		1. No any L 2 messages.			

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_12_2
Group: GSM_L3_MS_v4220/RR/
Purpose: To verify that the MS is able to correctly release a SDCCH after having received a CHANNEL RELEASE message, even if the SS does not L2 acknowledge the L2 DISC frame.
Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+Varinit_tch(C_arfcn_tchA, C_arfcn_tchAd)			
4		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubG, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(TCV_tch_arfcn), FreqTCH(TCV_tch_arfcn), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
5	body	+subtree1			
6		+subtree2			
		subtree1			
7		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
8		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
9		+local_immass			
10		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
11		(TCV_Res := OM_NoUAforDISC(TCV_ch))			
12		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
13		L?DL_RelIn	DLRelInd_01		
14		L?DL_RelIn	DLRelInd_01		
15		START T_dly(2000)			
16		?TIMEOUT T_dly			
17		(TCV_Res := OM_ResumUAforDISC(TCV_ch))			
18		START T_dly(3000)			
19	L_0324	L?OTHERWISE		(F)	
20	L_0325	?TIMEOUT T_dly		(P)	
21		L!MDL_RelRq	MDLRelReq(TCV_ch)		Local end release
		subtree2			
22		START T_dly(12000)			
23		?TIMEOUT T_dly			
24		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
25		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
26		+local_immass			
27	L_0326	L?DL_EstInPgRes	PagingRes(PagingRes_01)	(P)	
28		+PostMainLinkRel(TCV_ch)			

		local_immass			
29		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r02(TCV_Rr, TCV_Fn, TSPX_SDCCH8SubG, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), TCV_tch_arfcn))		
Detailed Comments:		1. Use TSPX_SDCCH8SubG.			

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_12_3
Group: GSM_L3_MS_v4220/RR/
Purpose: To verify that the MS is able to correctly release a full-rate TCH after having received a CHANNEL RELEASE message.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+gsmOrDcs			
4		+FullRateCh_A_1_nociph(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH(C_arfcn_tchA), FreqTCH(C_arfcn_tchAd), TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
5	body	+subtree1			
6		+subtree2			
		subtree1			
7		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
8		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
9		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
10		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r13(TCV_Rr, TCV_Fn, TCV_chd1, TimingAdv(30)))		
11		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
12		ACTIVATE(OtherEventsFail)			Restore Normal default
13		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
14		L?DL_RelIn	DLRelInd_01		
15		START T_dly(3000)			
16	L_0327	L?OTHERWISE		(F)	
17	L_0328	?TIMEOUT T_dly		(P)	
		subtree2			
18		START T_dly(12000)			
19		?TIMEOUT T_dly			
20		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
21		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
22		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
23		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r13(TCV_Rr, TCV_Fn, TCV_chd1, TimingAdv(30)))		
24	L_032	L?DL_EstInPgRes	PagingRes(	(P)	

	9		PagingRes_01)		
25		ACTIVATE(OtherEventsFail)			Restore Normal default
26		+PostMainLinkRel(TCV_chTch)			
		gsmOrDcs			
27		[TSPC_PGSM OR TSPC_EGSM]			
28		(TCV_chd1 := ChDescrp_nfh('00001'B, TSPX_TmSltDef, TSPX_TscDef, C_arfcn_tchA))			
29		[TSPC_DCS]			
30		(TCV_chd1 := ChDescrp_nfh('00001'B, TSPX_TmSltDef, TSPX_TscDef, C_arfcn_tchAd))			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_12_4
Group: GSM_L3_MS_v4220/RR/
Purpose: To verify that the MS is able to correctly release a TCH/F after having received a CHANNEL RELEASE message, even if the SS does not L2 acknowledge the L2 DISC frame.
Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+local_set_channel_descr			
4		+FullRateCh_A_1_nociph(C_Ass, TSPX_TmSltC, TSPX_TscC, ChMod_speech, FreqTCH(C_arfcn_tchA), FreqTCH(C_arfcn_tchAd), TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
5	body	+subtree1			
6		+Varinit_tch(C_arfcn_tchA, C_arfcn_tchAd)			
7		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH(TCV_tch_arfcn), FreqTCH(TCV_tch_arfcn), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
8		+subtree2			
		subtree1			
9		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
10		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
11		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r13(TCV_Rr, TCV_Fn, TCV_chd1, TimingAdv(30)))		
12		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
13		(TCV_Res := OM_NoUAforDISC(TCV_chTch))			
14		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
15		L?DL_RelIn	DLRelInd_01		
16		L?DL_RelIn	DLRelInd_01		
17		START T_dly(2000)			
18		?TIMEOUT T_dly			
19		(TCV_Res := OM_ResumUAforDISC(TCV_chTch))			
20		START T_dly(3000)			
21	L_0330	L?OTHERWISE		(F)	
22	L_0331	?TIMEOUT T_dly		(P)	
23		L!MDL_RelRq	MDLRelReq(TCV_ch)		local end release

		subtree2			
24		START T_dly(12000)			
25		?TIMEOUT T_dly			
26		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
27		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
28		+local_immass			
29	L_0332	L?DL_EstInPgRes	PagingRes(PagingRes_01)	(P)	
30		+PostMainLinkRel(TCV_ch)			
		local_set_channel_descr			
31		[TSPC_PGSM OR TSPC_EGSM]			
32		(TCV_chd1 :=ChDescrp_nfh('00001'B, TSPX_TmSltDef, TSPX_TscDef, C_arfcn_tchA))			
33		[TSPC_DCS]			
34		(TCV_chd1 := ChDescrp_nfh('00001'B, TSPX_TmSltDef, TSPX_TscDef, C_arfcn_tchAd))			
		local_immass			
35		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r02(TCV_Rr, TCV_Fn, TSPX_SDCCH8SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), TCV_tch_arfcn))		
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_13_1

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS, after receiving an ASSIGNMENT COMMAND message with a starting time and channel descriptions both for before and after the starting time, and ready to access before the indicated time, performs correctly the assignment using the description for before the time, and eventually starts using the frequency parameters for after the time at the time indicated in the message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_203_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubA, C_Immass, TSPX_TmSltA, TSPX_TscA, ChMod_sign, FreqSDCCH8_rg1, FreqSDCCH8_rd1, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			2.
4		+local_set_channel_for_ass			3.
5	body	+ltree_body			
		ltree_body			
6		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
7		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
8		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
9		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r27(TCV_Rr, TCV_Fn, TSPX_TmSltA, TSPX_TscA, TimingAdv(30)))		4.
10		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
11		ACTIVATE(OtherEventsFail)			Restore Normal default
12		(TCV_Fn := OM_ComingFn(TCV_ch), TCV_Null := OM_SendNextOn(TCV_ch, TCV_Fn), TCV_Strt := OC_StartTime(TCV_Fn, C_StrT_1000, 1))			5.
13		(TCV_AssCmd := AsgnCmd_r14(TSPX_TmSltB, TSPX_TscB, TCV_Strt), TCV_Null := OM_FreqDef(TCV_Strt, MoblAllc_r06, TCV_WorkingCh, ChDescrp_fh(TSPX_Chtp1, TSPX_TmSltB, TSPX_TscB, INT_TO_BIT(TSPX_Maio2, 6), INT_TO_BIT(TSPX_Hsn2, 6)), TCV_cchd1, TCV_Fn))			
14		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
15		+local_hoppingCHK			6.
16		+PostMainLinkRel(TCV_WorkingCh)			
		local_hoppingCHK			
17		(TCV_Res := OM_FHCHK(TCV_WorkingCh, TCV_cchd1, MoblAllc_r06, ChDescrp_fh(			

		TSPX_Chtp1, TSPX_TmSltB, TSPX_TscB, INT_TO_BIT(TSPX_Maio2, 6), INT_TO_BIT(TSPX_Hsn2, 6)), C_StrT_1000, TCV_Fn))			
18	L_0333	[TCV_Res]		(P)	
19	L_0334	[NOT TCV_Res]		(F)	
		local_set_channel_for_ass			
20		(TCV_n := BIT_TO_INT(TSPX_Chtp1))			
21		[TCV_n = 1]			
22		+FullRateCh_A_1_nociph(C_Ass, TSPX_TmSltB, TSPX_TscB, ChMod_sign, Freq_rg2, Freq_rd2,TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			7.
23		(TCV_WorkingCh := TCV_chTch)			
24		+Set_CellChDescr(C_CellA)			
25		[(TCV_n = 2) OR (TCV_n = 3)]			
26		+HalfRateCh_A_1_nociph(OC_LeastBits(TSPX_Chtp1, 1), C_Ass, TSPX_TmSltB, TSPX_TscB, ChMod_sign, Freq_rg2, Freq_rd2, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			8.
27		(TCV_WorkingCh := TCV_chTch)			
28		+Set_CellChDescr(C_CellA)			
29		[(TCV_n >= 8) AND (TCV_n <= 15)]			
30		+SDCCH8_A_2_nociph(OC_LeastBits(TSPX_Chtp1, 3), C_Ass, TSPX_TmSltB, TSPX_TscB, ChMod_r01, Freq_rg2, Freq_rd2, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			9.
31		(TCV_WorkingCh := TCV_ch)			
32		+Set_CellChDescr(C_CellA)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH.
2. To set a physical channel as hopping SDCCH8 channel, hopping parameters defined by PIXIT.
3. To setup the before time and after time channels for ASSIGNMENT COMMAND.
4. To assign the hopping SDCCH8 channel.
5. To calculate the STARTING TIME.
6. To check whether the after time frequency hopping is correct at the RF burst level.
7. If the selected channel type is TCH/F, setup a physical channel as full rate channel for before time.
8. If the selected channel type is TCH/H, setup a physical channel as half rate channel for before time.
9. If the selected channel type is SDCCH8, setup a physical channel as SDCCH8 channel for before time.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_13_2

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS, after receiving an ASSIGNMENT COMMAND message with a starting time and channel descriptions both for before and after the starting time, performs correctly the assignment using the description for after the time if the indicated time has already elasped when the Mobile Station is ready to trasmit.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_203_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubB, C_Immass, TSPX_TmSltB, TSPX_TscB, ChMod_sign, FreqSDCCH8_rg2, FreqSDCCH8_rd2, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			2.
4		+local_set_channelForAss			3.
5	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r28(TCV_Rr, TCV_Fn, TSPX_TmSltB, TSPX_TscB, TimingAdv(30)))		4.
9		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		(TCV_Fn := OM_ComingFn(TCV_ch), TCV_Null := OM_SendNextOn(TCV_ch, TCV_Fn))			5.
12		(TCV_AssCmd := AsgnCmd_r15(TSPX_TmSltC, TSPX_TscC, TSPX_TmSltC, TSPX_TscC, OC_StartTime(TCV_Fn, 5, 1)))			
13		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
14		+local_continue			6.
15		+PostMainLinkRel(TCV_chTch)			
		local_continue			
16		(TCV_Res := OM_FHCHK(TCV_chTch, TCV_cchd1, MoblAllc_r09, ChDescrp_fh(TSPX_Chtp2, TSPX_TmSltC, TSPX_TscC, INT_TO_BIT(TSPX_Maio5, 6), INT_TO_BIT(TSPX_Hsn5, 6)), 5, TCV_Fn))			
17	L_0335	[TCV_Res]		(P)	
18	L_0336	[NOT TCV_Res]		(F)	

		local_set_channelForAss			
19		(TCV_n := BIT_TO_INT(TSPX_Chtp2))			
20		[TCV_n = 1]			
21		+FullRateCh_A_1_nociph(C_Ass, TSPX_TmSltC, TSPX_TscC, ChMod_r02, Freq_rg4, Freq_rd4, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			7.
22		+Set_CellChDescr(C_CellA)			
23		[(TCV_n = 2) OR (TCV_n = 3)]			
24		+HalfRateCh_A_1_nociph(OC_LeastBits(TSPX_Chtp2, 1), C_Ass, TSPX_TmSltC, TSPX_TscC, ChMod_r02, Freq_rg4, Freq_rd4, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			8.
25		+Set_CellChDescr(C_CellA)			
26		[(TCV_n >= 8) AND (TCV_n <= 15)]			
27		+SDCCH8_A_2_nociph(OC_LeastBits(TSPX_Chtp2, 3), C_Ass, TSPX_TmSltC, TSPX_TscC, ChMod_r02, Freq_rg4, Freq_rd4, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			9.
28		+Set_CellChDescr(C_CellA)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH.
2. To set a physical channel as hopping SDCCH8 channel, hopping parameters defined by PIXIT.
3. To setup the after time channels for ASSIGNMENT COMMAND.
4. To assign the hopping SDCCH8 channel.
5. To calculate the STARTING TIME.
6. To check whether the after time frequency hopping is correct at the RF burst level.
7. If the selected channel type is TCH/F, setup a physical channel as full rate channel for after time.
8. If the selected channel type is TCH/H, setup a physical channel as half rate channel for after time.
9. If the selected channel type is SDCCH8, setup a physical channel as SDCCH8 channel for after time.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_13_3

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS, after receiving a FREQUENCY REDEFINITION message and then an ASSIGNMENT COMMAND message with a starting time and channel descriptions both for before and after the starting time, failling the assignment and returning on the old channel, and ready to access before the time indicated in the FREQUENCY REDEFINITION, resumes transmission on the channels used at the time of the reception of the FREQUENCF REDEFINITION message and eventually starts using the frequency parameters at the time indicated in the FREQUENCY REDEFINITION message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_203_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+local_channel_setup			2.
4	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
5		L?DL_RacInChRq(TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r29(TCV_Rr, TCV_Fn, TSPX_TmSltD, TSPX_TscD, TimingAdv(30)))		
8		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		+localtree			
		localtree			
11		(TCV_Fn := OM_ComingFn(TCV_WorkingCh), TCV_Strt := OC_StartTime(TCV_Fn, 5000, 0), TCV_chd1 := ChDescrp_fh(TSPX_Chtp3, TSPX_TmSltD, TSPX_TscD, INT_TO_BIT(TSPX_Maio7, 6), INT_TO_BIT(TSPX_Hsn6, 6)), TCV_Null := OM_FreqDef(TCV_Strt, MoblAllc_r11, TCV_WorkingCh, TCV_chd1, TCV_cchd1, TCV_Fn), TCV_Null := OM_SendNextOn(TCV_WorkingCh, TCV_Fn))			3.
12		L!DL_DatRqFrqre	FrqRedfSnd(TCV_WorkingCh, FreqRedef_01(TCV_chd1, MoblAllc_r11, TCV_Strt, CellChDes_Omit))		4.
13		(TCV_AssCmd := AsgnCmd_r16(TSPX_TmSltE, TSPX_TscE, OC_StartTime(TCV_Fn, 4000, 1)))			
14		+AssCh_failure(TCV_WorkingCh, TCV_AssCmd, TRUE)			
15		+ltree_check			6.
16		+PostMainLinkRel(TCV_WorkingCh)			
		ltree_check			

17		[TCV_WorkingCh = TCV_ch]			
18		(TCV_Res := OM_FHCHK(TCV_WorkingCh, TCV_cchd1, MoblAllc_r11, ChDescrp_fh(TSPX_Chtp4, TSPX_TmSltD, TSPX_TscD, INT_TO_BIT(TSPX_Maio7, 6), INT_TO_BIT(TSPX_Hsn6, 6)), 5000, TCV_Fn))			
19	L_0337	[TCV_Res]		(P)	
20	L_0338	[NOT TCV_Res]		(F)	
		local_channel_setup			
21		(TCV_n := BIT_TO_INT(TSPX_Chtp3))			
22		[TCV_n = 1]			
23		+FullRateCh_A_1_nociph(C_Immass, TSPX_TmSltD, TSPX_TscD, ChMod_sign, Freq_rg5, Freq_rd5, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			7.
24		(TCV_WorkingCh := TCV_chTch)			
25		+Set_CellChDescr(C_CellA)			
26		[(TCV_n = 2) OR (TCV_n = 3)]			
27		+HalfRateCh_A_1_nociph(OC_LeastBits(TSPX_Chtp3, 1), C_Immass, TSPX_TmSltD, TSPX_TscD, ChMod_sign, Freq_rg5, Freq_rd5, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			8.
28		(TCV_WorkingCh := TCV_chTch)			
29		+Set_CellChDescr(C_CellA)			
30		[(TCV_n >= 8) AND (TCV_n <= 15)]			
31		+SDCCH8_A_1_nociph(OC_LeastBits(TSPX_Chtp3, 3), C_Immass, TSPX_TmSltD, TSPX_TscD, ChMod_sign, Freq_rg5, Freq_rd5, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			9.
32		(TCV_WorkingCh := TCV_ch)			
33		+Set_CellChDescr(C_CellA)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH.
2. To set two physical channels, one as hopping channel for immediate assignment another one as hopping channel for after time channel, parameters defined by PIXIT.
3. To calculate the starting time for frequency redifinition.
4. To send FREQUENCY REDIFINITION message.
5. To calculate the starting time and send the ASSIGNMENT COMMAND on special frame TCV_Fn. The assigned channel not activated in the tester.
6. To check whether the after time frequency hopping is correct at the RF burst level.
7. If the selected channel type is TCH/F, setup a physical channel as full rate channel for immediate assignment.
8. If the selected channel type is TCH/H, setup a physical channel as half rate channel for immediate assignment.
9. If the selected channel type is SDCCH8, setup a physical channel as SDCCH8 channel for immediate assignment.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_13_4

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS, after receiving a FREQUENCY REDEFINITION message and then an ASSIGNMENT COMMAND message with a starting time and channel descriptions both for before and after the starting time, failing the assignment and returning on the old channel, and ready to access after the time indicated in the FREQUENCY REDEFINITION, resumes transmission using the frequency parameters indicated in the FREQUENCY REDEFINITION message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_203_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+local_channel_setup			2.
4	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r30(TCV_Rr, TCV_Fn, TSPX_TmSltF, TSPX_TscF, TimingAdv(30)))		
8		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		+local_continue			
		local_continue			
11		(TCV_Fn := OM_ComingFn(TCV_WorkingCh), TCV_Strt := OC_StartTime(TCV_Fn, 214, 0), TCV_chd1 := ChDescrp_fh(TSPX_Chtp5, TSPX_TmSltF, TSPX_TscF, INT_TO_BIT(TSPX_Maio11, 6), INT_TO_BIT(TSPX_Hsn10, 6)), TCV_Null := OM_FreqDef(TCV_Strt, MoblAllc_r15, TCV_WorkingCh, TCV_chd1, TCV_cchd1, TCV_Fn), TCV_Null := OM_SendNextOn(TCV_WorkingCh, TCV_Fn))			
12		L!DL_DatRqFrqre	FrqRedfSnd(TCV_WorkingCh, FreqRedef_01(TCV_chd1, MoblAllc_r15, TCV_Strt, CellChDes_Omit))		4.
13		(TCV_AssCmd := AsgnCmd_r17(TSPX_TmSltF, TSPX_TscF, OC_StartTime(TCV_Fn, 5000, 1)))			
14		+AssCh_failure(TCV_WorkingCh, TCV_AssCmd, TRUE)			5.
15		+PostMainLinkRel(TCV_WorkingCh)			
		local_channel_setup			
16		(TCV_n := BIT_TO_INT(TSPX_Chtp5))			
17		[TCV_n = 1]			

18		+FullRateCh_A_1_nociph(C_Immass, TSPX_TmSltF, TSPX_TscF, ChMod_sign, Freq_rg7, Freq_rd7, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			7.
19		(TCV_WorkingCh := TCV_chTch)			
20		+Set_CellChDescr(C_CellA)			
21		[(TCV_n = 2) OR (TCV_n = 3)]			
22		+HalfRateCh_A_1_nociph(OC_LeastBits(TSPX_Chtp5, 1), C_Immass, TSPX_TmSltF, TSPX_TscF, ChMod_sign, Freq_rg7, Freq_rd7, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			8.
23		(TCV_WorkingCh := TCV_chTch)			
24		+Set_CellChDescr(C_CellA)			
25		[(TCV_n >= 8) AND (TCV_n <= 15)]			
26		+SDCCH8_A_1_nociph(OC_LeastBits(TSPX_Chtp5, 3), C_Immass, TSPX_TmSltF, TSPX_TscF, ChMod_sign, Freq_rg7, Freq_rd7, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			9.
27		(TCV_WorkingCh := TCV_ch)			
28		+Set_CellChDescr(C_CellA)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH.
2. To set two physical channels, one as hopping channel for immediate assignment another one as hopping channel for after time channel, parameters defined by PIXIT. Note: Although the test case is designed for TCH/F, TCH/H & SDCCH8 the TSPX_Chtp5 should be assigned to an SDCCH8 subchannel because calculations of starting time T1 for TCH/F & TCH/H have not yet been carried out. No simple values can be defined.
3. To calculate the starting time for frequency redifinition.
4. To send FREQUENCY REDIFINITION message.
5. To calculate the starting time and send the ASSIGNMENT COMMAND on special frame TCV_Fn. The assigned channel not activated in the tester.
6. The expected ASSIGNMENT FAILURE message received on the channel defined by frequency redifinition.
7. If the selected channel type is TCH/F, setup a physical channel as full rate channel for immediate assignment.
8. If the selected channel type is TCH/H, setup a physical channel as half rate channel for immediate assignment.
9. If the selected channel type is SDCCH8, setup a physical channel as SDCCH8 channel for immediate assignment.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_13_5

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS, after receiving a HANDOVER COMMAND message with a starting time and channel descriptions both for before and after the starting time, and ready to access before the time performs correctly the handover using the description for before the time, and then starts using the frequency parameters for after the time at the the time indicated in the message.

Default: OtherEventsFail, RcvHdOvAcc

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_203_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+StartCellB(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, C_S0, TSPX_TscDef, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_1, C_NCC)			2.
4		+local_channel_setup			3.
5	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r31(TCV_Rr, TCV_Fn, TSPX_TmSltG, TSPX_TscG, TimingAdv(30)))		
9		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		+localcontinue			
		localcontinue			
12		(TCV_Null := OM_StartMsrReport(TCV_sacch8))			
13		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_02)		
14		(TCV_Null := OM_StopMsrReport(TCV_sacch8), TCV_Fn := OM_ComingFn(TCV_ch), TCV_Strt := OC_StartTime(TCV_Fn, C_StrT_1000, 1), TCV_Null := OM_SendNextOn(TCV_ch, TCV_Fn), TCV_chd1 := ChDescrp_fh(TSPX_Chtp7, TSPX_TmSltDef, TSPX_TscDef, INT_TO_BIT(TSPX_Maio15, 6), INT_TO_BIT(TSPX_Hsn15, 6)), TCV_Null := OM_FreqDef(TCV_Strt, MoblAllc_r18iei, TCV_chTch, TCV_chd1, TCV_cchd1, TCV_Fn),TCV_Null:=OM_SendNextOnHO(TCV_chTch,2))			4.
15		ACTIVATE(RcvHdOvAcc, OtherEventsFail_01)			
16		+local_sendHoCMD			
17		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_chTch,		

			TimingAdv(30))		
18		L?DL_RacInHoacc	HndOvAccRcv(TCV_chTch, HandOverAcc_01)		
19		L?DL_EstIn	DLEstInd(TCV_chTch)		
20		L?DL_DatInHoCom	HndOvCmpRcv(TCV_chTch, HandOverCmp_01)		
21		+ltree_Hoppingcheck			
22		+PostMainLinkRel(TCV_chTch)			
		ltree_Hoppingcheck			
23		(TCV_Res := OM_FHCHK(TCV_chTch, TCV_cchd1, MoblAllc_r18iei, TCV_chd1, C_StrT_1000, TCV_Fn))			
24	L_0339	[TCV_Res]		(P)	
25	L_0340	[NOT TCV_Res]		(F)	
		local_sendHoCMD			
26		[TSPC_PGSM OR TSPC_EGSM]			
27		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd(C_NCC, C_BCC_1, TSPX_TmSltDef, TSPX_TscDef, TCV_cchd1, TCV_Strt, C_BCCHcarrierB_ho, TSPX_Chtp7, TSPX_Maio15, TSPX_Maio16, TSPX_Hsn15, TSPX_Hsn16, TSPX_HoRefA, TSPX_PwrlvlA, Synchi(C_not_report_otd, C_non_synchronised), ChMod_r03, MoblAllc_r18iei, MoblAllc_r20, CphMod_02iei))		
28		[TSPC_DCS]			
29		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd(C_NCC, C_BCC_1, TSPX_TmSltDef, TSPX_TscDef, TCV_cchd1, TCV_Strt, C_BCCHcarrierB_hod, TSPX_Chtp7, TSPX_Maio15, TSPX_Maio16, TSPX_Hsn15, TSPX_Hsn16, TSPX_HoRefA, TSPX_PwrlvlA, Synchi(C_not_report_otd, C_non_synchronised), ChMod_r03, MoblAllc_r18iei, MoblAllc_r20, CphMod_02iei))		
		local_channel_setup			
30		(TCV_n := BIT_TO_INT(TSPX_Chtp7))			
31		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubB, C_Immass, TSPX_TmSltG, TSPX_TscG, ChMod_sign, Freq_rg9, Freq_rd9, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			

32		[TCV_n = 1]			
33		+FullRateCh_B_1_nociph(C_Asynho, TSPX_TmSltDef, TSPX_TscDef, ChMod_r03, Freq_rg11, Freq_rd11, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			5.
34		+Set_CellChDescr(C_CellB)			
35		[(TCV_n = 2) OR (TCV_n = 3)]			
36		+HalfRateCh_B_1_nociph(OC_LeastBits(TSPX_Chtp7, 1), C_Asynho, TSPX_TmSltDef, TSPX_TscDef, ChMod_r03, Freq_rg11, Freq_rd11, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			6.
37		+Set_CellChDescr(C_CellB)			
38		[(TCV_n >= 8) AND (TCV_n <= 15)]			
39		+SDCCH8_B_1_nociph(OC_LeastBits(TSPX_Chtp7, 3), C_Asynho, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, Freq_rg11, Freq_rd11, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			7.
40		+Set_CellChDescr(C_CellB)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH for cell A.
2. To setup a physical channel as BCCH, CCCH for cell B.
3. To setup a physical channel as SDCCH8 in cell A and setup 2 physical channels as hopping channels in cell B.
4. To get the future frame number for sending HANDOVER COMMAND and calculate starting time.
5. If the required channel is full rate channel, setup the before time full rate hopping channel.
6. If the required channel is half rate channel, setup the before time half rate hopping channel.
7. If the required channel is SDCCH8 channel, setup the before time SDCCH8 hopping channel.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_13_6

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS, after receiving a HANDOVER COMMAND message with a starting time and channel descriptions both for before and after the starting time, and ready to tramsmit after the indicated time, performs correctly the handover using the description for after the time.

Default: OtherEventsFail, RcvHdOvAcc

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_203_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+StartCellB(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, C_S0, TSPX_TscB, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_1, C_NCC)			2.
4		+local_config_channel			3.
5	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r32(TCV_Rr, TCV_Fn, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30)))		
9		+local_continue			
		local_continue			
10		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
11		ACTIVATE(OtherEventsFail)			Restore Normal default
12		(TCV_Fn := OM_ComingFn(TCV_ch), TCV_Strt := OC_StartTime(TCV_Fn, 5, 1), TCV_Null := OM_SendNextOn(TCV_ch, TCV_Fn), TCV_Null:=OM_SendNextOnHO(TCV_chTch, 2))			4.
13		ACTIVATE(RcvHdOvAcc, OtherEventsFail_01)			
14		+local_sendHoCMD			
15		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_chTch, TimingAdv(30))		
16		L?DL_RacInHoacc	HndOvAccRcv(TCV_chTch, HandOverAcc_01)		
17		L?DL_EstIn	DLEstInd(TCV_chTch)		
18		L?DL_DatInHoCom	HndOvCmpRcv(TCV_chTch, HandOverCmp_01)		
19		+PostMainLinkRel(TCV_chTch)			
		local_sendHoCMD			

20		[TSPC_PGSM OR TSPC_EGSM]			
21		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd(C_NCC, C_BCC_1, TSPX_TmSltB, TSPX_TscB, TCV_cchd1, TCV_Strt, C_BCCHcarrierB_ho, TSPX_Chtp8, TSPX_Maio18, TSPX_Maio19, TSPX_Hsn18, TSPX_Hsn19, TSPX_HoRefA, TSPX_PwrlvlB, Synchi(C_not_report_otd, C_non_synchronised), ChMod_r04, MoblAllc_r22, MoblAllc_r23, CphMod_omit))		
22		[TSPC_DCS]			
23		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd(C_NCC, C_BCC_1, TSPX_TmSltB, TSPX_TscB, TCV_cchd1, TCV_Strt, C_BCCHcarrierB_hod, TSPX_Chtp8, TSPX_Maio18, TSPX_Maio19, TSPX_Hsn18, TSPX_Hsn19, TSPX_HoRefA, TSPX_PwrlvlB, Synchi(C_not_report_otd, C_non_synchronised), ChMod_r04, MoblAllc_r22, MoblAllc_r23, CphMod_omit))		
		local_config_channel			
24		(TCV_n := BIT_TO_INT(TSPX_Chtp8))			
25		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubC, C_Immass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, Freq_rg12, Freq_rd12, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
26		[TCV_n = 1]			
27		+FullRateCh_B_1_nociph(C_Asynho, TSPX_TmSltB, TSPX_TscB, ChMod_r04, Freq_rg13, Freq_rd13, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			5.
28		+Set_CellChDescr(C_CellB)			
29		[(TCV_n = 2) OR (TCV_n = 3)]			
30		+HalfRateCh_B_1_nociph(OC_LeastBits(TSPX_Chtp8, 1), C_Asynho, TSPX_TmSltB, TSPX_TscB, ChMod_r04, Freq_rg13, Freq_rd13, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			6.
31		+Set_CellChDescr(C_CellB)			
32		[(TCV_n >= 8) AND (TCV_n <= 15)]			
33		+SDCCH8_B_1_nociph(OC_LeastBits(TSPX_Chtp8, 3), C_Asynho, TSPX_TmSltB, TSPX_TscB, ChMod_sign, Freq_rg13, Freq_rd13, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			7.
34		+Set_CellChDescr(C_CellB)			

Detailed Comments: 1. To setup a physical channel as BCCH, CCCH for cell A.

	2. To setup a physical channel as BCCH, CCCH for cell B. 3. To setup a physical channel as SDCCH8 in cell A and setup a physical channel as hopping channels in cell B. 4. To get the future frame number for sending HANDOVER COMMAND and calculate starting time. 5. If the required channel is full rate channel, setup the after time full rate hopping channel. 6. If the required channel is half rate channel, setup the after time half rate hopping channel. 7. If the required channel is SDCCH8 channel, setup the after time SDCCH8 hopping channel.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_13_7

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS, after receiving a FREQUENCY REDEFINITION message and then an HANDOVER COMMAND message with a starting time and channel descriptions both for before and after the starting time, failing the handover, and ready to access the old channel before the time indicated in the FREQUENCY REDEFINITION, resumes transmission on the channels used at the time of the reception of the FREQUENCY REDEFINITION message and eventually starts using the new frequency parameters at the time indicated in the FREQUENCY REDEFINITION message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_203_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+local_channel_setup			3.
4		+StartCellB(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TSPX_TmSltD, TSPX_TscD, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_1, C_NCC)			2.
5	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r33(TCV_Rr, TCV_Fn, TSPX_TmSltC, TSPX_TscC, TimingAdv(30)))		
9		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		+localtree			
		localtree			
12		(TCV_Fn := OM_ComingFn(TCV_WorkingCh), TCV_Strt := OC_StartTime(TCV_Fn, 5000, 0), TCV_chd1 := ChDescrp_fh(TSPX_Chtp9, TSPX_TmSltC, TSPX_TscC, INT_TO_BIT(TSPX_Maio21, 6), INT_TO_BIT(TSPX_Hsn20, 6)), TCV_Null := OM_FreqDef(TCV_Strt, MoblAllc_r25, TCV_WorkingCh, TCV_chd1, TCV_cchd1, TCV_Fn), TCV_Null := OM_SendNextOn(TCV_WorkingCh, TCV_Fn))			4.
13		L!DL_DatRqFrqre	FrqRedfSnd(TCV_WorkingCh, FreqRedef_01(TCV_chd1, MoblAllc_r25, TCV_Strt, CellChDes_Omit))		5.

14		(TCV_Strt := OC_StartTime(TCV_Fn, 4000, 1))			6.
15		+local_handoverCMD			
16		L?DL_EstIn	DLEstInd(TCV_WorkingCh)		
17		L?DL_DatInHofl	HndOvFlRcv(TCV_WorkingCh, HandOvFail_02)		
18		+local_HoppingCHK			7.
19		+PostMainLinkRel(TCV_WorkingCh)			
		local_HoppingCHK			
20		(TCV_Res := OM_FHCHK(TCV_WorkingCh, TCV_cchd1, MoblAllc_r25, TCV_chd1, 5000, TCV_Fn))			
21	L_0341	[TCV_Res]		(P)	
22	L_0342	[NOT TCV_Res]		(F)	
		local_handoverCMD			
23		[TSPC_PGSM OR TSPC_EGSM]			
24		L!DL_DatRqHoCmd	HndOvSnd(TCV_WorkingCh, HandOverCmd(C_NCC, C_BCC_1, TSPX_TmSltD, TSPX_TscD, CellChDes_20_Bman, TCV_Strt, C_BCCHcarrierB_ho, TSPX_Chtp10, TSPX_Maio22, TSPX_Maio23, TSPX_Hsn22, TSPX_Hsn23, TSPX_HoRefA, TSPX_PwrlvlC, Synchi(C_not_report_otd, C_non_synchronised), ChMod_r05, MoblAllc_r26, MoblAllc_r27, CphMod_omit))		
25		[TSPC_DCS]			
26		L!DL_DatRqHoCmd	HndOvSnd(TCV_WorkingCh, HandOverCmd(C_NCC, C_BCC_1, TSPX_TmSltD, TSPX_TscD, CellChDes_201_Bd, TCV_Strt, C_BCCHcarrierB_hod, TSPX_Chtp10, TSPX_Maio22, TSPX_Maio23, TSPX_Hsn22, TSPX_Hsn23, TSPX_HoRefA, TSPX_PwrlvlC, Synchi(C_not_report_otd, C_non_synchronised), ChMod_r05, MoblAllc_r26, MoblAllc_r27, CphMod_omit))		
		local_channel_setup			
27		(TCV_n := BIT_TO_INT(TSPX_Chtp9))			
28		[TCV_n = 1]			
29		+FullRateCh_A_1_nociph(C_Immass, TSPX_TmSltC, TSPX_TscC, ChMod_sign,			7.

		Freq_rg14, Freq_rd14, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
30		(TCV_WorkingCh := TCV_chTch)			
31		+Set_CellChDescr(C_CellA)			
32		[(TCV_n = 2) OR (TCV_n = 3)]			
33		+HalfRateCh_A_1_nociph(OC_LeastBits(TSPX_Chtp9, 1), C_Immass, TSPX_TmSltC, TSPX_TscC, ChMod_sign, Freq_rg14, Freq_rd14, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			8.
34		(TCV_WorkingCh := TCV_chTch)			
35		+Set_CellChDescr(C_CellA)			
36		[(TCV_n >= 8) AND (TCV_n <= 15)]			
37		+SDCCH8_A_1_nociph(OC_LeastBits(TSPX_Chtp9, 3), C_Immass, TSPX_TmSltC, TSPX_TscC, ChMod_sign, Freq_rg14, Freq_rd14, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			9.
38		(TCV_WorkingCh := TCV_ch)			
39		+Set_CellChDescr(C_CellA)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH.
2. To set a physical channel as BCCH, CCCH for Cell B.
3. To setup a physical channel for immediate assignmemnt.
4. To calculate the starting time for frequency redifinition.
5. To send FREQUENCY REDIFINITION message.
6. To calculate the starting time and send the HANDOVER COMMAND on special frame TCV_Fn.
   The assigned channel not activated in the tester.
7. To check whether the after time frequency hopping is correct at the RF burst level.
8. If the selected channel type is TCH/F, setup a physical channel as full rate channel for immediate assignment.
9. If the selected channel type is TCH/H, setup a physical channel as half rate channel for immediate assignment.
10. If the selected channel type is SDCCH8, setup a physical channel as SDCCH8 channel for immediate assignment.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_13_8

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS, after receiving a FREQUENCY REDEFINITION message and then an HANDOVER COMMAND message with a starting time and channel descriptions both for before and after the starting time, failing the handover, and returning on the old channel, and ready to access after the time indicated in the FREQUENCY REDEFINITION, resumes transmission using the new frequency parameters indicated in the FREQUENCY REDEFINITION message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_203_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+local_channel_setup			3.
4		+StartCellB(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, C_S0, TSPX_TscE, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_1, C_NCC)			2.
5	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r34(TCV_Rr, TCV_Fn, TSPX_TmSltD, TSPX_TscD, TimingAdv(30)))		
9		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		+local_continue			
		local_continue			
12		(TCV_Fn := OM_ComingFn(TCV_WorkingCh), TCV_Strt := OC_StartTime(TCV_Fn, 265, 0), TCV_chd1 := ChDescrp_fh(TSPX_Chtp11, TSPX_TmSltD, TSPX_TscD, INT_TO_BIT(TSPX_Maio25, 6), INT_TO_BIT(TSPX_Hsn24, 6)), TCV_Null := OM_FreqDef(TCV_Strt, MoblAllc_r29, TCV_WorkingCh, TCV_chd1, TCV_cchd1, TCV_Fn), TCV_Null := OM_SendNextOn(TCV_WorkingCh, TCV_Fn))			4.
13		L!DL_DatRqFrqre	FrqRedfSnd(TCV_WorkingCh, FreqRedef_01(TCV_chd1, MoblAllc_r29, TCV_Strt, CellChDes_Omit))		5.
14		(TCV_Strt := OC_StartTime(TCV_Fn, 5000, 1))			6.

15		+local_sendHoCMD			
16		L?DL_EstIn	DLEstInd(TCV_WorkingCh)		
17		L?DL_DatInHofl	HndOvFlRcv(TCV_WorkingCh, HandOvFail_02)		7.
18		+PostMainLinkRel(TCV_WorkingCh)			
		local_sendHoCMD			
19		[TSPC_PGSM OR TSPC_EGSM]			
20		L!DL_DatRqHoCmd	HndOvSnd(TCV_WorkingCh, HandOverCmd(C_NCC, C_BCC_1, TSPX_TmSltE, TSPX_TscE, CellChDes_20_B, TCV_Strt, C_BCCHcarrierB_ho, TSPX_Chtp12, TSPX_Maio26, TSPX_Maio27, TSPX_Hsn26, TSPX_Hsn27, TSPX_HoRefA, TSPX_PwrlvlD, Synchi(C_not_report_otd, C_non_synchronised), ChMod_r06, MoblAllc_r30iei, MoblAllc_r31iei, CphMod_omit))		
21		[TSPC_DCS]			
22		L!DL_DatRqHoCmd	HndOvSnd(TCV_WorkingCh, HandOverCmd(C_NCC, C_BCC_1, TSPX_TmSltE, TSPX_TscE, CellChDes_201_Bdiei, TCV_Strt, C_BCCHcarrierB_hod, TSPX_Chtp12, TSPX_Maio26, TSPX_Maio27, TSPX_Hsn26, TSPX_Hsn27, TSPX_HoRefA, TSPX_PwrlvlD, Synchi(C_not_report_otd, C_non_synchronised), ChMod_r06, MoblAllc_r30iei, MoblAllc_r31iei, CphMod_omit))		
		local_channel_setup			
23		(TCV_n := BIT_TO_INT(TSPX_Chtp11))			
24		[TCV_n = 1]			
25		+FullRateCh_A_1_nociph(C_Immass, TSPX_TmSltD, TSPX_TscD, ChMod_sign, Freq_rg15, Freq_rd15, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			7.
26		(TCV_WorkingCh := TCV_chTch)			
27		+Set_CellChDescr(C_CellA)			
28		[(TCV_n = 2) OR (TCV_n = 3)]			
29		+HalfRateCh_A_1_nociph(OC_LeastBits(TSPX_Chtp11, 1), C_Immass, TSPX_TmSltD, TSPX_TscD, ChMod_sign, Freq_rg15, Freq_rd15, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			8.
30		(TCV_WorkingCh := TCV_chTch)			

31		+Set_CellChDescr(C_CellA)			
32		[(TCV_n >= 8) AND (TCV_n <= 15)]			
33		+SDCCH8_A_1_nociph(OC_LeastBits(TSPX_Chtp11, 3), C_Immass, TSPX_TmSltD, TSPX_TscD, ChMod_sign, Freq_rg15, Freq_rd15, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			9.
34		(TCV_WorkingCh := TCV_ch)			
35		+Set_CellChDescr(C_CellA)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH.
2. To set a physical channel as BCCH, CCCH for Cell B.
3. To setup a physical channel for immediate assignmemnt. Note: Although the test case is designed for TCH/F, TCH/H & SDCCH8 the TSPX_Chtp11 should be assigned to an SDCCH8 subchannel because calculations of starting time T1 for TCH/F & TCH/H have not yet been carried out. No simple values can be defined.
4. To calculate the starting time for frequency redifinition.
5. To send FREQUENCY REDIFINITION message.
6. To calculate the starting time and send the HANDOVER COMMAND on special frame TCV_Fn.
   The assigned channel not activated in the tester.
7. The HANDOVER FAILURE message received on the channel using the after time hopping parameters of the frequency redifinetion message.
8. If the selected channel type is TCH/F, setup a physical channel as full rate channel for immediate assignment.
9. If the selected channel type is TCH/H, setup a physical channel as half rate channel for immediate assignment.
10. If the selected channel type is SDCCH8, setup a physical channel as SDCCH8 channel for immediate assignment.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_13_9

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS, after receiving an IMMEDIATE ASSIGNMENT message with a starting time and channel descriptions both for before and after the starting time, and ready to access before the indicated time, performs correctly the assignment using the description for before the time, and then starts using the frequency parameters for after the time indicated in the message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_203_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+local_channel_setup			2.
4	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
5		L?DL_RacInChRq(TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn1 := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		+local_continue			
		local_continue			
8		(TCV_Fn := OM_ComingFn(TCV_agch), TCV_n := 0 - 867, TCV_Strt := OC_StartTime(TCV_Fn, TCV_n, 0), TCV_Fn := TCV_Strt.fn, TCV_Strt := OC_StartTime(TCV_Fn, TSPX_Tm3, 1), TCV_chd1 := ChDescrp_fh(TSPX_Chtp13, TSPX_TmSltF, TSPX_TscF, INT_TO_BIT(TSPX_Maio28, 6), INT_TO_BIT(TSPX_Hsn28, 6)), TCV_Null := OM_FreqDef(TCV_Strt, MoblAllc_r32, TCV_WorkingCh, ChDescrp_fh(TSPX_Chtp13, TSPX_TmSltF, TSPX_TscF, INT_TO_BIT(TSPX_Maio28, 6), INT_TO_BIT(TSPX_Hsn28, 6)), TCV_cchd1, TCV_Fn), TCV_Null := OM_SendNextOn(TCV_agch, TCV_Fn))			3. 4.
9		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r35(TCV_Rr, TCV_Fn1, TSPX_TmSltF, TSPX_TscF, TimingAdv(30), TCV_Strt))		
10		L?DL_EstInPgRes	PgRes(TCV_WorkingCh, PagingRes_01)		
11		ACTIVATE(OtherEventsFail)			Restore Normal default
12		+ltree_aftertime_hopping_check			5.
13		+PostMainLinkRel(TCV_WorkingCh)			
		ltree_aftertime_hopping_check			
14		(TCV_Res := OM_FHCHK(TCV_WorkingCh, TCV_cchd1, MoblAllc_r32, ChDescrp_fh(TSPX_Chtp13, TSPX_TmSltF, TSPX_TscF, INT_TO_BIT(TSPX_Maio28, 6), INT_TO_BIT(TSPX_Hsn28, 6)), 5000, TCV_Fn))			
15	L_0343	[TCV_Res]		(P)	
16	L_034	[NOT TCV_Res]		(F)	

	4				
		local_channel_setup			
17		(TCV_n := BIT_TO_INT(TSPX_Chtp13))			
18		[TCV_n = 1]			
19		+FullRateCh_A_1_nociph(C_Immass, TSPX_TmSltF, TSPX_TscF, ChMod_sign, Freq_rg16, Freq_rd16, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			7.
20		(TCV_WorkingCh := TCV_chTch)			
21		+Set_CellChDescr(C_CellA)			
22		[(TCV_n = 2) OR (TCV_n = 3)]			
23		+HalfRateCh_A_1_nociph(OC_LeastBits(TSPX_Chtp13, 1), C_Immass, TSPX_TmSltF, TSPX_TscF, ChMod_sign, Freq_rg16, Freq_rd16, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			8.
24		(TCV_WorkingCh := TCV_chTch)			
25		+Set_CellChDescr(C_CellA)			
26		[(TCV_n >= 8) AND (TCV_n <= 15)]			
27		+SDCCH8_A_1_nociph(OC_LeastBits(TSPX_Chtp13, 3), C_Immass, TSPX_TmSltF, TSPX_TscF, ChMod_sign, Freq_rg16, Freq_rd16, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			9.
28		(TCV_WorkingCh := TCV_ch)			
29		+Set_CellChDescr(C_CellA)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH.
2. To setup a physical channel for immediate assignmemnt.
3. To calculate the starting time for frequency parameters change.
4. To inform the test system to change frequency parameters after starting time.
5. To check whether the MS transimitting on the after time frequncy parameters.
6. If the selected channel type is TCH/F, setup a physical channel as full rate channel for immediate assignment before time.
7. If the selected channel type is TCH/H, setup a physical channel as half rate channel for immediate assignment before time.
8. If the selected channel type is SDCCH8, setup a physical channel as SDCCH8 channel for immediate assignment before time.

Test Case Dynamic Behaviour

Test Case Name: TC_26_6_13_10

Group: GSM_L3_MS_v4220/RR/

Purpose: To verify that the MS, after receiving an IMMEDIATE ASSIGNMENT message with a starting time and channel descriptions both for before and after the starting time, performs correctly the assignment using the frequencies and hopping sequence for after the time if the indicated time has already elasped when the MObile Station is ready to transmit.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_203_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+local_channel_setup			2.
4		(TCV_Fn := OM_ComingFn(TCV_agch), TCV_Strt := OC_StartTime(TCV_Fn, 5, 1), TCV_Null := OM_SendNextOn(TCV_agch, TCV_Fn))			3.
5		START T_dly(2800)			4.
6		?TIMEOUT T_dly			
7	body	L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
8		L?DL_RacInChRq(TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
9		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
10		+local_continue			
		local_continue			
11		(TCV_chd1 := ChDescrp_fh(TSPX_Chtp14, TSPX_TmSltG, TSPX_TscG, INT_TO_BIT(TSPX_Maio30, 6), INT_TO_BIT(TSPX_Hsn30, 6)))			3.
12		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r36(TCV_Rr, TCV_Fn, TSPX_TmSltG, TSPX_TscG, TimingAdv(30), TCV_Strt))		
13		L?DL_EstInPgRes	PgRes(TCV_WorkingCh, PagingRes_01)		4.
14		ACTIVATE(OtherEventsFail)			Restore Normal default
15		+ltree_hopping_check			
16		+PostMainLinkRel(TCV_WorkingCh)			
		ltree_hopping_check			
17		(TCV_Res := OM_FHCHK(TCV_WorkingCh, TCV_cchd1, MoblAllc_r33, ChDescrp_fh(TSPX_Chtp14, TSPX_TmSltG, TSPX_TscG, INT_TO_BIT(TSPX_Maio30, 6), INT_TO_BIT(TSPX_Hsn30, 6)), 5000, TCV_Fn))			
18	L_0345	[TCV_Res]		(P)	
19	L_0346	[NOT TCV_Res]		(F)	
		local_channel_setup			

20		(TCV_n := BIT_TO_INT(TSPX_Chtp14))			
21		[TCV_n = 1]			
22		+FullRateCh_A_1_nociph(C_Immass, TSPX_TmSltG, TSPX_TscG, ChMod_sign, Freq_rg17, Freq_rd17, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			5.
23		(TCV_WorkingCh := TCV_chTch)			
24		+Set_CellChDescr(C_CellA)			
25		[(TCV_n = 2) OR (TCV_n = 3)]			
26		+HalfRateCh_A_1_nociph(OC_LeastBits(TSPX_Chtp14, 1), C_Immass, TSPX_TmSltG, TSPX_TscG, ChMod_sign, Freq_rg17, Freq_rd17, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			6.
27		(TCV_WorkingCh := TCV_chTch)			
28		+Set_CellChDescr(C_CellA)			
29		[(TCV_n >= 8) AND (TCV_n <= 15)]			
30		+SDCCH8_A_1_nociph(OC_LeastBits(TSPX_Chtp14, 3), C_Immass, TSPX_TmSltG, TSPX_TscG, ChMod_sign, Freq_rg17, Freq_rd17, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			7.
31		(TCV_WorkingCh := TCV_ch)			
32		+Set_CellChDescr(C_CellA)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH.
2. To setup a physical channel for immediate assignmemnt.
3. To calculate the starting time for immediate assignment.
4. To paging response message received on the channel with after time frequency parameters.
5. If the selected channel type is TCH/F, setup a physical channel as full rate channel for immediate assignment after time.
6. If the selected channel type is TCH/H, setup a physical channel as half rate channel for immediate assignment after time.
7. If the selected channel type is SDCCH8, setup a physical channel as SDCCH8 channel for immediate assignment after time.

Test Group MM

Test Case Dynamic Behaviour					
Test Case Name:	TC_26_7_1				
Group:	GSM_L3_MS_v4220/MM/				
Purpose:	To verify that the MS is able to receive and acknowledge a new TMSI by means of an explicit TMSI reallocation procedure. To verify that the MS has stored the TMSI in a non-volatile memory. The implicit reallocation procedure is tested in section 26.7.4.1.				
Default:	OtherEventsFail				
Comments:	Initial Conditions of MS: The MS has valid TMSI(TMSI_01), CKSN and Kc.It is "idle updated" on cell B. The initial conditions will be arrived in procedures of PREAMBLE. Required SIM card: default Foreseen final state of the MS: The MS has valid TMSI(TMSI_01), CKSN and Kc.It is "idle updated" on cell A.				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_2, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+StartCellA(C_E_neighbourdefault, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_02, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			
4		+PreEstRRC_MM(MiTmsi_01, TCV_cksn, TimingAdv(30))			
5		+Ciphering_on(TCV_ch)			
6	body	+ltree_body			
		ltree_body			
7		+TmsiReallocation(MiTmsi_02, C_MCC_1, C_PLMN_1, C_LAC_2)			1)
8		+ChanRel(TCV_ch)			
9		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			no ciphering
10		+MM_PwrOrSimOff(C_SIMIn)			
11		START T_dly(10000)			
12		?TIMEOUT T_dly			
13		+MM_PwrOrSimOn(C_SIMIn)			
14		+WaitForInService			
15		+CCCH_group_Paging_group(TCV_Ccd0B, TSPX_IMSI)			
16		+SelectPagingCh(C_CellB)			
17		+PreEstRRC_MM(MiTmsi_02, TCV_cksn, TimingAdv(30))			2)
18		+ChanRel(TCV_ch)			
19		+ltree_switchcell			
		ltree_switchcell			
20		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			4)
21		+MM_LUP_tmsirealloc(MiTmsi_01, MiTmsi_02, C_MCC_1, C_PLMN_1, C_LAC_2, TCV_lac,			5)

		C_normal_updating, TCV_cksn, TimingAdv(30))			
22		+ChanRel(TCV_ch)			
23		+CCCH_group_Paging_group(TCV_Ccd0A, TSPX_IMSI)			
24		+SelectPagingCh(C_CellA)			
25		+Wait(C_T_Wait1)			
26		+PreEstRRC_MM(MiTmsi_01, TCV_cksn, TimingAdv(30))			6)
27		+ChanRel_end(TCV_ch)			

Detailed Comments:
1) new TMSI from PIXIT, test body starts from here.
2) RR connection with the new tmsi 2.
3) Initialisation of variables for cell A.
4) The RF level of cell B is lowered until the MS selects cell A and starts the Location Update.
5) Location Update from cell B to A.
6) RR connection with the new tmsi 1

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_2_1

Group: GSM_L3_MS_v4220/MM/

Purpose: 1) To check that a Mobile Station correctly responds to an Authentication(TCV_ch) Request message by
sending an Authentication(TCV_ch) Response message with the SRES information field set to the same
value as the one produced by the authentication algorithm in the network.
2) To check that a Mobile Station indicates in a Paging Response message the ciphering key sequence number which was allocated to it through the authentication procedure.

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has valid TMSI, CKSN(CKSN1) and Kc.It is "idle updated" on the cell.

Foreseen final state of the MS:
The MS has valid TMSI(TMSI1), CKSN and Kc.It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+PreEstRRC_MM(MiTmsi_01, TCV_cksn, TimingAdv(30))			
4		(TCV_cksn:= TSPX_CKSNB)			
5	body	+Authentication(TCV_ch, TCV_cksn, TSPX_RANDDef)			
6		+ChanRel(TCV_ch)			
7		+Wait(C_T_Wait1)			
8		+PreEstRRC_MM(MiTmsi_01, TCV_cksn, TimingAdv(30))			1)
9	post	+ChanRel_end(TCV_ch)			

Detailed Comments:
1) RR-Establishment with a new CKSN

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_2_2

Group: GSM_L3_MS_v4220/MM/

Purpose: 1) To check that ,after reception of an Authentication Reject message, the Mobile Station:
1.1 does not perform normal location updating
1.2 does not perform periodic location updating
1.3 does not respond to paging with TMSI
1.4 rejects any request from CM entity for MM connection except for emergency call
1.5 does not perform IMSI detach if deactivated

2) To check that, after reception of an Authentication Reject message the Mobile Station, if it supports speech, accepts a request for an emergency call by sending a CHANNEL REQUEST message with the establishment cause set to "emergency call" and includes an IMEI as mobile identity in the CM SERVICE REQUEST message.

3) To check that, after reception of an Authentication Reject message and after having been deactivated and reactivated, the MS performs location updating using its IMSI as mobile identity and indicates deleted LAI and CKSN.

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has valid TMSI, CKSN2 and Kc.It is "idle updated" on cell B.

Foreseen final state of the MS:
The MS has valid TMSI, CKSN1 and Kc.It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(700)			
2		+BasicServiceMO(TSPX_MOBscSvcG, TSPX_MOChRateG)			
3		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_2, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+StartCellA(C_E_neighbourdefault, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_02, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			
5	body	+PreEstRRC_MM(MiTmsi_01, TCV_cksn, TimingAdv(30))			
6		+Authentication(TCV_ch, TCV_cksn, TSPX_RANDDef)			
7		L!DL_DatRqAuthRej	AuthRejSnd(TCV_ch, AuthReject_01)		
8		+ChanRel(TCV_ch)			
9		+MM_no_paging(MiTmsi_01, 3000, TCV_Ccd0B, C_CellB)			2)
10		START T_dly(15000)			
11		?TIMEOUT T_dly			
12		+local_continue			
		local_continue			
13		+MM_no_cmservices(3000)			3)
14		+MM_check_ecall1(TimingAdv(30), MiImei_01)			
15		+ltree_switchcelltoA			
16		+NoReaction(30000)			5)

17		+ltree_continue1			
		ltree_continue1			
18		+NoReaction(420000)			6)
19		+ImsiDetachNoReaction(3000, C_SIMneedRmv)			7)
20		+MM_PwrOrSimOn(C_SIMneedRmv)			
21		+MM_LUPauth2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TCV_cksn, TSPX_RANDA, TimingAdv(30))			8)
22	post	+ChanRel_end(TCV_ch)			
		ltree_switchcelltoA			
23		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNA, TSPX_RANDA, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			9)

Detailed Comments:

1) Initial condition: CKSN2, which has the default value.
2) Check of purpose 1.3
3) Check of purpose 1.4
4) Check of purpose 1.4, emergency call
5) Check of purpose 1.1
6) Check of purpose 1.2
7) Check of purpose 1.5
8) Check normal location update after SimOutIn or SwitchOnOff or PowerOnOff.
9) CKSN1 takes the value different from the default value.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_3_1

Group: GSM_L3_MS_v4220/MM/

Purpose: 1) To verify that the MS sends identity information as requested by the system in the following cases: IMSI and TMSI are requested in non-ciphered mode, IMEI is requested in ciphered mode.
2) To verify that the MS sends its IMEI, when requested to do so, in non-ciphered mode.
3) To verify that the MS sends its IMEISV, when requested to do so, in non-ciphered mode.

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has valid TMSI. It is "idle updated" on the cell.

Foreseen final state of the MS:
The MS has valid TMSI. It is "idle updated" on the cell.
In the 11.10 there are two test sequences. In TTCN they are combined to only one.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+PreEstRRC_MM(MiTmsi_01, TCV_cksn, TimingAdv(30))			
4		+IdentityRequest(C_IMSI, MiImsi_01)			
5		+IdentityRequest(C_TMSI, MiTmsi_01)			
6		+Ciphering_on(TCV_ch)			
7		+IdentityRequest(C_IMEI, MiImei_01)			
8		+Ciphering_off(TCV_ch)			
9		+IdentityRequest(C_IMEI, MiImei_01)			
10		+IdentityRequest(C_IMEISV, MiImeisv_01)			
11	post	+ChanRel_end(TCV_ch)			

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_3_2

Group: GSM_L3_MS_v4220/MM/

Purpose: To check that the MS behaves correctly when activated with an IMSI of length less than the maximum length. In this condition, the MS shall:

1. perform location updating
2. answer to paging with IMSI
3. give the correct IMSI when asked by an IDENTITY REQUEST
4. attempt CM connection establishment when requested to
5. attempt call re-establishment when needed
6. attempt IMSI detach when needed
7. erase its TMSI when the IMSI is sent by the network in a LOCATION UPDATING ACCEPT or a TMSI REALLOCATION COMMAND message.

Default: OtherEventsFail

Comments: Initial Conditions of MS:

The MS has no valid TMSI. It is "idle updated" on the cell. The IMSI has the value '001011234'.

This test case shall use the SIM Card 2 with 'IMSI=001011234' and HPLMN_search_period=6min.

Foreseen final state of the MS:

The MS has no valid TMSI. It is "idle updated" on the cell.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcB, TSPX_MTChRateB)			
3		(TCV_Null := OO_SIM2Ins())			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Imsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, C_shortIMSI, C_Test_fh, C_NCCP_2)			0.
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6	body	+ltree_body			
		ltree_body			
7		+ltree_check_idreqimsi			1.
8		+ltree_EnterCCstateU10			
9		(TCV_Null := OM_Deactivate(TCV_chTch, TCV_sacchTch), TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
10		+ltree_check_reestablish_imsi			2.
11		+ltree_check_tmsi_imsi			3.
12		+ImsiDetach(MiImsi_31, TimingAdv(30), C_SIMIn)			4.
13		+ltree_check_luppoweron			5.
14		+ltree_check_luplacchange			6.
15		+ltree_check_cmservreqimsi			7.
16		+ChanRel_end(TCV_ch)			
		ltree_check_idreqimsi			
17		+PreEstRRC_MM(MiImsi_31, TCV_cksn, TimingAdv(30))			
18		+IdentityRequest(C_IMSI, MiImsi_31)			

		ltree_check_reestablish_imsi			
19		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_10)		
20		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
21		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
22	L_0347	L?DL_EstInCmreRq	CmreReq_02(MiImsi_31, LocAreaId_anyomit, TSPX_CKSNDef)	(P)	
23		ACTIVATE(OtherEventsFail)			Restore Normal default
24		+TmsiReallocation(MiTmsi_01, C_MCC_1, C_PLMN_1, C_LAC_1)			
25		+ChanRel(TCV_ch)			
		ltree_check_tmsi_imsi			
26		+Wait(C_T_Wait1)			
27		+PreEstRRC_MM(MiTmsi_01, TCV_cksn, TimingAdv(30))			
28		+Authentication(TCV_ch, TCV_cksn, TSPX_RANDDef)			
29		+TmsiReallocation(MiImsi_31, C_MCC_1, C_PLMN_1, C_LAC_1)			
30		+ChanRel(TCV_ch)			
31		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
		ltree_check_luppoweron			
32		+MM_PwrOrSimOn(C_SIMIn)			
33		+MM_LUP2(MiTmsi_01iei, MiImsi_31, C_MCC_1, C_PLMN_1, TCV_lac, TCV_lac, C_imsi_attach, TCV_cksn, TimingAdv(30))			
		ltree_check_luplacchange			
34		(TCV_lac:= C_LAC_2)			
35		(TCV_slot := C_S0, TCV_tsc := C_BCC)			
36		+StartCellA_1(C_E_default, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), 5, 1, 0, 1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, TCV_lac, BcchFreqLst_01, BcchFreqLst_01, BcchFreqLst_48, BcchFreqLst_48, C_noRestablishment)			
37		+WaitForInService			
38		+MM_LUP_imsi(C_T_Wait1stChReq, MiImsi_31iei, MiTmsi_01, C_MCC_1, C_PLMN_1, C_LAC_1, TCV_lac, C_normal_updating, TimingAdv(30))			
		ltree_check_cmservreqimsi			
39		+BasicServiceMO(TSPX_MOBscSvcF, TSPX_MOChRateF)			
40		+InitCall(TCV_Service)			
41		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
42		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
43		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
44	L_034	L?DL_EstInCmsRq	CMSerReq(	(P)	

	8		CMServiceReq_30(MiImsi_31))		
45		ACTIVATE(OtherEventsFail)			Restore Normal default
		ltree_EnterCCstateU10			
46		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNDef, TSPX_RANDDef))		
47		L?DL_DatInAuthRes(TCV_Sres := DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
48		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, TSPX_RANDDef), TCV_Setup_mt.sig := Signal_01)			
49	L_0349	[NOT TCV_Res]		(I)	
50		+PostMainLinkRel(TCV_ch)			
51		[TCV_Res]			
52		+Ciphering_on(TCV_ch)			
53		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		
54		L?DL_DatInCallCo(TCV_CallCfm := DL_DatInCallCo.msg)	CallCfm(CallConfirm_01(TI_01))		
55		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
56		+ltree_assignTch			
57		L?DL_DatInAlert	AlertRcv(AlertingInd_02(TI_01))		
58		(TCV_Null := OO_HookOff())			
59		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
60		+ltree_assignTch			
		ltree_assignTch			
61		+CCAssignTCH(TSPX_TmSltDef, TSPX_TscDef)			
62		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
63		+CCstatuschk_02(TCV_chTch, C_U10, TI_02, TI_01)			

Detailed Comments:

0. Initial condition: no valid TMSI.
1. check of purpose 2. and 3.
2. check of purpose 5.
3. check of purpose 7.
4. check of purpose 6.
5. check of purpose 1.
6. check of purpose 1. and 7.
7. check of purpose 4.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_1

Group: GSM_L3_MS_v4220/MM/

Purpose: To test the behaviour of the MS if the network accepts the location updating of the MS. For the network response three different cases are identified:

1) TMSI is allocated,
2) Location updating accept contains neither TMSI nor IMSI,
3) Location updating accept contains IMSI.

Default: OtherEventsFail

Comments: Initial Conditions of MS:

The MS has valid TMSI(TMSI1), CKSN(CKSN1) and Kc.It is "idle updated" on cell A.

Foreseen final state of the MS:

The MS has no valid TMSI and no CKSN. It is "idle updated" on cell B.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+StartCellB(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			
4	body	+ltree_switchcelltoB_LupPag1			
5		+ltree_switchcelltoA_LupPag			
6		+ltree_switchcelltoB_LupPag2			
		ltree_switchcelltoB_LupPag1			
7		+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
8		+MM_LUP2(MiTmsi_02iei, MiTmsi_01, C_MCC_1, C_PLMN_1, C_LAC_1, C_LAC_2, C_normal_updating, TCV_cksn, TimingAdv(30))			
9		+WaitForInService			
10		+CCCH_group_Paging_group(TCV_Ccd0B, TSPX_IMSI)			
11		+SelectPagingCh(C_CellB)			
12		+PreEstRRC_MM(MiTmsi_02, TCV_cksn, TimingAdv(30))			
13		+ChanRel(TCV_ch)			
		ltree_switchcelltoA_LupPag			
14		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
15		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_09)		
16		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
17		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn,		

			TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
18		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_31(MiTmsi_02, C_MCC_1, C_PLMN_1, C_LAC_2, C_normal_updating, TCV_cksn))		
19		ACTIVATE(OtherEventsFail)			Restore Normal default
20	L_0350	L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1, C_PLMN_1, C_LAC_1))	(P)	
21		+ChanRel(TCV_ch)			
22		+WaitForInService			
23		+CCCH_group_Paging_group(TCV_Ccd0A, TSPX_IMSI)			
24		+SelectPagingCh(C_CellA)			
25		+PreEstRRC_MM(MiTmsi_02, TCV_cksn, TimingAdv(30))			
26		+ChanRel(TCV_ch)			
		ltree_switchcelltoB_LupPag2			
27		+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
28		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_09)		
29		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
30		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
31		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_31(MiTmsi_02, C_MCC_1, C_PLMN_1, C_LAC_1, C_normal_updating, TCV_cksn))		
32		ACTIVATE(OtherEventsFail)			Restore Normal default
33	L_0351	L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiImsi_01iei, C_MCC_1, C_PLMN_1, C_LAC_2))	(P)	
34		+ChanRel(TCV_ch)			
35		+WaitForInService			
36		+MM_no_paging(MiTmsi_02, 5000, TCV_Ccd0B, C_CellB)			
37		+PreEstRRC_MM(MiImsi_01, TCV_cksn, TimingAdv(30))			
38		+ChanRel_end(TCV_ch)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_2_1

Group: GSM_L3_MS_v4220/MM/

Purpose: To test the behaviour of the MS if the network rejects the location updating of the MS with the cause "IMSI unknown in HLR", "illegal MS" or "Illegal ME".

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has no valid TMSI and no CKSN. It is "idle updated" on cell A.
Foreseen final state of the MS:
The MS has valid TMSI and no CKSN. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(2100)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+StartCellB(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			
4	body	+execution1			
5		+execution2			
6		+execution3			
		execution1			
7		+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
8		+MM_LupRej(C_rc_imsiunknownhlr, C_normal_updating, TimingAdv(30))			
9		+ltree_main			
10		+ChanRel(TCV_ch)			
		execution2			
11		+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
12		+MM_LupRej(C_rc_illegal_ms, C_normal_updating, TimingAdv(30))			
13		+ltree_main			
14		+ChanRel(TCV_ch)			
		execution3			
15		+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
16		+MM_LupRej(C_rc_illegal_me, C_normal_updating, TimingAdv(30))			
17		+ltree_main			
18	post	+ChanRel_end(TCV_ch)			
		ltree_main			

19		+BasicServiceMO(TSPX_MOBscSvcD, TSPX_MOChRateD)			
20		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
21		+NoReaction(30000)			
22		+NoReaction(420000)			
23		+MM_no_paging(MiImsi_01, 3000, TCV_Ccd0A, C_CellA)			
24		+MM_no_paging(MiTmsi_01, 3000, TCV_Ccd0A, C_CellA)			
25		+MM_no_cmservices(3000)			
26		+MM_check_ecall1(TimingAdv(30), MiImei_01)			
27		+ImsiDetachNoReaction(3000, C_SIMneedRmv)			
28		+MM_PwrOrSimOn(C_SIMneedRmv)			
29		+MM_LUPauth2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_2_2_1

Group: GSM_L3_MS_v4220/MM/

Purpose: To test the behaviour of the MS if the network rejects the location updating of the MS with the cause "PLMN not allowed".

Default: OtherEventsFail

Comments: Initial conditions for the Mobile Station:
- The MS has a valid TMSI. It is "idle updated" on cell C.
- The MS is in manual mode for PLMN selection.

Final state of the Mobile Station:
Idle Updated with TMSI on cell C. The MS is in automatic mode for PLMN selection.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(720)			
2		+BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		+IdleUpdated(C_E_default, C_CellC, C_SCH_C, C_BCCH_C_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellC, C_MCC_1, C_PLMN_1, C_LAC_3, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_07, BcchFreqLst_07, BcchFreqLst_15, BcchFreqLst_15, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnC, C_arfcnCd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4	body	+MM_PwrOrSimOff(C_SIMIn)			
5		(TCV_Null := OM_StopCell(C_CellC))			
6		+Start_2cells(C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_1, C_LAC_2)			
7		+Varinit_fix(C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
8		+MM_PwrOrSimOn(C_SIMIn)			
9		+ltree_continue			
10		+RmvForbiddenPlmn1(C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			8)
		ltree_continue			
11		(TCV_Null := OO_PLMNselModeMan())			
12		(TCV_Null := OO_SelPLMN(C_PLMN_2))			
13		+MM_LupRej(C_rc_plmn_not, C_normal_updating, TimingAdv(30))			1)
14		+NoReaction(420000)			2)
15		+ImsiDetachNoReaction(3000, C_SIMneedRmv)			3)
16		+MM_PwrOrSimOn(C_SIMneedRmv)			
17		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
18		+ltree_continue1			
		ltree_continue1			
19		+NoReaction(60000)			4)

20		+MM_check_ecall1(TimingAdv(30), MiImsi_01)			5)
21		(TCV_Service := TSPX_MOBscSvcA)			
22		+MM_no_cmservices(3000)			5)
23		+MM_PwrOrSimOff(C_SIMIn)			
24		(TCV_Null := OM_StopCell(C_CellA), TCV_Null := OM_StopCell(C_CellB))			
25		+ltree_continue2			
		ltree_continue2			
26		+StartCellC(C_E_default, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), 0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_3, C_noRestablishment, C_BCC, C_NCC)			6)
27		+Varinit_fix(C_CellC, C_LAC_3, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnC, C_arfcnCd, TCV_Ccd0C, TSPX_IMSI)			
28		+MM_PwrOrSimOn(C_SIMIn)			
29		(TCV_Null := OO_PLMNselModeAuto())			
30		+MM_LUP2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TimingAdv(30))			7)

Detailed Comments:

1) Location updating rejected with cause = PLMN not allowed.

MS shall

2) not perform periodic updating
3) not perform IMSI detach when switched off.
4) not perform normal location updating after switching to a new LAC in the same PLMN and when that PLMN is not selected manually.
5) reject any request from CM entity for MM connection other than for emergency call.
6) switch to a new cell with a new PLMN.
7) perform normal location updating after entering in a new PLMN.
8. The PLMN of cell B is supressed on the SIM's forbidden PLMN list.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_2_2_2

Group: GSM_L3_MS_v4220/MM/

Purpose: To test the behaviour of the MS if the network rejects the location updating of the MS with the cause "PLMN not allowed".

Default: OtherEventsFail

Comments: Initial conditions for the Mobile Station:

- The MS has a valid TMSI. It is "idle updated" on cell C.
- The MS is in manual mode for PLMN selection.

Final state of the Mobile Station:

Idle Updated with TMSI on cell C. The MS is in automatic mode for PLMN selection.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(720)			
2		+IdleUpdated(C_E_default, C_CellC, C_SCH_C, C_BCCH_C_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellC, C_MCC_1, C_PLMN_1, C_LAC_3, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_07, BcchFreqLst_07, BcchFreqLst_15, BcchFreqLst_15, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnC, C_arfcnCd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		(TCV_Null := OM_StopCell(C_CellC))			
5		+Start_2cells(C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_1, C_LAC_2)			
6		+Varinit_fix(C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
7		+MM_PwrOrSimOn(C_SIMIn)			
8		+ltree_continue			
9		+RmvForbiddenPlmn1(C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			3)
		ltree_continue			
10		(TCV_Null := OO_PLMNselModeMan())			
11		(TCV_Null := OO_SelPLMN(C_PLMN_2))			
12		+MM_LupRej(C_rc_plmn_not, C_normal_updating, TimingAdv(30))			1)
13		(TCV_Null := OO_SelPLMN(C_PLMN_2))			2)
14		+MM_LupInit(MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TimingAdv(30))			
15		+ChanRel(TCV_ch)			
16		+MM_PwrOrSimOff(C_SIMIn)			
17		(TCV_Null := OM_StopCell(C_CellA), TCV_Null := OM_StopCell(C_CellB))			
18		+ltree_continue2			
		ltree_continue2			
19		+StartCellC(C_E_default, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), 0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_3, C_noRestablishment, C_BCC, C_NCC)			
20		+Varinit_fix(C_CellC, C_LAC_3,			

		TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnC, C_arfcnCd, TCV_Ccd0C, TSPX_IMSI)			
21		+MM_PwrOrSimOn(C_SIMIn)			
22		(TCV_Null := OO_PLMNselModeAuto())			
23		+MM_LUP2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TimingAdv(30))			
Detailed Comments:		1) Location updating rejected with cause = PLMN not allowed. 2) MS shall perform normal location updating after switching to a new LAC in the same PLMN and when that PLMN is selected manually. 3) The PLMN of cell B is supressed on the SIM's forbidden PLMN list.			

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_2_3

Group: GSM_L3_MS_v4220/MM/

Purpose: To test the behaviour of the MS if the network rejects the location updating of the MS with the cause "Location Area not allowed".

To test that the MS deletes the list of forbiden location areas after switching of the MS.

Default: OtherEventsFail

Comments: Initial conditions for the Mobile Station:

- The MS has a valid TMSI. It is "idle updated" on cell A.

Initial conditions for the Mobile Station:

- The MS has a valid TMSI. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(720)			
2		+BasicServiceMO(TSPX_MOBscSvcE, TSPX_MOChRateE)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+StartCellB(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			
5		+ltree_body			
		ltree_body			
6		+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
7		+MM_LupRej(C_rc_LAnotallowed, C_normal_updating, TimingAdv(30))			1)
8		+NoReaction(420000)			2)
9		+MM_no_paging(MiTmsi_01, 3000, TCV_Ccd0B, C_CellB)			3)
10		+MM_no_cmservices(3000)			4)
11		+MM_check_ecall1(TimingAdv(30), MiImsi_01)			4)
12		+ImsiDetachNoReaction(3000, C_SIMIn)			5)
13		+MM_PwrOrSimOn(C_SIMIn)			
14		+MM_LupRej2(C_rc_LAnotallowed, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TimingAdv(30))			6)
15		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
16		+MM_LUPauth1(MiTmsi_01iei, C_MCC_1, C_PLMN_1, TCV_lac, C_normal_updating, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			7)

17			+ChanRel_end(TCV_ch)			

Detailed Comments:

1) Reject of Location Updating with the cause Location Area is not allowed.

MS shall

2) not perform periodic updating
3) not perform paging with TMSI
4) reject any request from CM entity for MM connection other than for emergency call.
5) not perform IMSI detach when switched off.
6) delete list of forbidden LAs after switch off and perform normal location updating
7) perform normal location updating after entering in a new Location Area.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_2_4_1

Group: GSM_L3_MS_v4220/MM/

Purpose: Test purpose 1

To test that on receipt of a rejection using the Roaming cause code, the MS ceases trying to update on that cell, that this situation continues for at least one periodic location interval period, and that the corresponding list is re-set by powering down the MS (the requirement in TS GSM 04.08 is that the list shall be retained for at least 12 hours. This aspect is not formally tested).

Default: OtherEventsFail

Comments: This testcase includes only the procedure of part 1 of GSM 11.10-1, section 26.7.4.2.4.

Initial Conditions of MS:

The MS has valid TMSI, CKSN and Kc.It is "idle updated" on cell B.

Foreseen final state of the MS:

The MS has no valid TMSI and no CKSN. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(720)			
2		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_ci_cellB, C_MCC_1, C_PLMN_2, C_LAC_2, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+StartCellA(C_E_neighbourdefault, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_1, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			
4	body	+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
5		+MM_LupRej(C_rc_roamingnot, C_normal_updating, TimingAdv(30))			1)
6		+NoReaction(420000)			2)
7		+MM_PwrOrSimOff(C_SIMIn)			3)
8		+MM_PwrOrSimOn(C_SIMIn)			3)
9		(TCV_Null := OO_PLMNselModeAuto())			
10		+MM_LupInit2(C_normal_updating, TimingAdv(30))			3)
11	L_0352	L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1, C_PLMN_2, TCV_lac))	(P)	4)
12		+ChanRel_end(TCV_ch)			

Detailed Comments:

1) Reject of Location Updating with the cause Roaming is not allowed.

MS shall

2) not perform periodic updating

3) reset the list of "forbidden location areas for national roaming" when powered down.

4) Location Updating Accept with LAI belonging to PLMN2 and without Mobile Identity.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_2_4_2

Group: GSM_L3_MS_v4220/MM/

Purpose: Test purpose 2

To test that if no cell is available, the MS does not answer to paging with TMSI, rejects a request from CM entity except other than emergency calls.

Default: OtherEventsFail

Comments: This testcase includes only the procedure of part 2 of GSM 11.10-1, section 26.7.4.2.4.

Initial Conditions of MS:

The MS has valid TMSI, CKSN and Kc.It is "idle updated" on cell B.

Foreseen final state of the MS:

The MS has no valid TMSI and no CKSN. It is in the "limited service" state on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(360)			
2		+BasicServiceMO(TSPX_MOBscSvcH, TSPX_MOChRateH)			
3		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_ci_cellB, C_MCC_1, C_PLMN_2, C_LAC_2, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+StartCellA(C_E_neighbourdefault, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_1, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			
5		+ltree_body			
		ltree_body			
6		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
7		+MM_LupRej(C_rc_roamingnot, C_normal_updating, TimingAdv(30))			1)
8		+Varinit_fix(C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
9		+MM_LupRej(C_rc_roamingnot, C_normal_updating, TimingAdv(30))			2)
10		+NoReaction(120000)			4)
11		+MM_no_paging(MiImsi_01, 3000, TCV_Ccd0B, C_CellB)			3b)
12		+Varinit_fix(C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
13		+MM_no_paging(MiImsi_01, 3000, TCV_Ccd0A, C_CellA)			3a)
14		+MM_no_cmservices(3000)			4)
15		+MM_check_ecall1(TimingAdv(30), MiImsi_01)			4)
16		+ChanRel_end(TCV_ch)			

Detailed Comments:	1) Reject of Location Updating with the cause Roaming is not allowed. MS shall 2) perform normal updating when a new location area is entered. 3a) not respond to paging with TMSI in cell a. 3b) not respond to paging with TMSI in cell b. 4) reject any request from CM entity for MM connection other than for emergency call.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_2_4_3

Group: GSM_L3_MS_v4220/MM/

Purpose: Test purpose 3

To test that at least 6 entries can be held in the list of "forbidden location areas for roaming" (the requirement in TS GSM 04.08 is to store at least 10 entries. This is not fully tested by the this procedure).

Default: OtherEventsFail

Comments: This testcase includes only the procedure of part 3 of GSM 11.10-1, section 26.7.4.2.4.

Initial Conditions of MS:

The MS has valid TMSI, CKSN and Kc.It is "idle updated" on cell B.

Foreseen final state of the MS:

The MS has no valid TMSI and no CKSN. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(1020)			
2		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_ci_cellB, C_MCC_1, C_PLMN_2, C_LAC_2, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+StartCellA(C_E_neighbourdefault, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_1, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			
4	body	+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
5		+MM_LupRej(C_rc_roamingnot, C_normal_updating, TimingAdv(30))			1)
6		+Varinit_fix(C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
7		+MM_LupRej(C_rc_roamingnot, C_normal_updating, TimingAdv(30))			2)
8		+Varinit_fix(C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
9		+ChgLAC_A(C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_3)			
10		+MM_LupRej(C_rc_roamingnot, C_normal_updating, TimingAdv(30))			2)
11		+ltree_continue			
		ltree_continue			
12		+Varinit_fix(C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
13		+ChgLAC_B(C_TxInt_5, C_Max_1, C_NECI_0,			

		C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_4)			
14		+MM_LupRej(C_rc_roamingnot, C_normal_updating, TimingAdv(30))			2)
15		+ChgLAC_A(C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_5)			
16		+MM_LupRej(C_rc_roamingnot, C_normal_updating, TimingAdv(30))			2)
17		+Varinit_fix(C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
18		+ChgLAC_B(C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_6)			
19		+MM_LupRej(C_rc_roamingnot, C_normal_updating, TimingAdv(30))			2)
20		START T_dly(420000)			3)
21	L_0353	?TIMEOUT T_dly		P	

Detailed Comments:

1) Reject of Location Updating with the cause Roaming is not allowed.

MS shall

2) perform normal updating when a new location area is entered.

3) not perform periodic updating

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_2_4_4

Group: GSM_L3_MS_v4220/MM/

Purpose: Test purpose 4

To test that if a cell of the Home PLMN is available then the MS returns to it in preference to any other available cell.

Default: OtherEventsFail

Comments: This testcase includes only the procedure of part 4 of GSM 11.10-1, section 26.7.4.2.4.

Initial Conditions of MS:

The MS has valid TMSI, CKSN and Kc.It is "idle updated" on cell A.

Foreseen final state of the MS:

The MS has no valid TMSI and no CKSN. It is "idle updated" on cell C.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(960)			
2		+IdleUpdated(73, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_2, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_01, BcchFreqLst_48, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+StartCellB_1(C_E_default, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_2, C_noRestablishment, C_BCC, C_NCC)			
4		+StartCellC(C_E_neighbourdefault, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_3, C_noRestablishment, C_BCC, C_NCC)			
5		+MM_LUPper(C_MCC_1, C_PLMN_2, TCV_lac, C_periodic_updating, TimingAdv(30))			1)
6		+ChgLAI_C(1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_Home, C_LAC_3)			2)
7		+MM_LUPperrej(C_rc_roamingnot, C_periodic_updating, TimingAdv(30))			3)
8		+Varinit_fix(C_CellC, C_LAC_3, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnC, C_arfcnCd, TCV_Ccd0C, TSPX_IMSI)			
9		+MM_LUP3(C_MCC_1, C_PLMN_Home, TCV_lac, C_normal_updating, TimingAdv(30))			4)

Detailed Comments:

1) Periodic Updating in cell A.

2) LAI change to HPLMN

3) Reject of Periodic Location Updating with the cause Roaming is not allowed in cell A.

4) MS shall periodically search to Home PLMN.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_2_4_5
Group: GSM_L3_MS_v4220/MM/
Purpose: Test purpose 5
To test that if the SIM is removed the list of "forbidden location areas for roaming" is cleared.
Default: OtherEventsFail
Comments: This testcase includes only the procedure of part 5 of GSM 11.10-1, section 26.7.4.2.4.

Initial Conditions of MS:
The MS has valid TMSI, CKSN and Kc.It is "idle updated" on cell B.

Foreseen final state of the MS:
The MS has no valid TMSI and no CKSN. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_SIMRmv]			
2		START T_guard(720)			
3		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_ci_cellB, C_MCC_1, C_PLMN_2, C_LAC_2, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+StartCellA(C_E_neighbourdefault, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_1, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			
5	body	+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
6		+MM_LupRej(C_rc_roamingnot, C_normal_updating, TimingAdv(30))			1)
7		+NoReaction(420000)			
8		+ImsiDetachNoReaction(3000, C_SIMneedRmv)			2)
9		+MM_PwrOrSimOn(C_SIMneedRmv)			3)
10		+MM_LupInit2(C_normal_updating, TimingAdv(30))			4)
11	L_0354	L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1, C_PLMN_1, TCV_lac))	(P)	no MI
12		+ChanRel_end(TCV_ch)			
13	L_0355	[NOT TSPC_SIMRmv]		I	

Detailed Comments:
1) Reject of Location Updating in cell B.
2) The SIM is removed.
3) The SIM is inserted
4) The MS shall reset the list of "forbidden location areas for roaming" when SIM is removed.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_3_1

Group: GSM_L3_MS_v4220/MM/

Purpose: To verify that when during the RR connection establishment phase of a location updating procedure, channel requests are not answered by the network, after expiry of T3213 (= 4s in Phase 2) and when the cell reselection procedure is finished the complete procedure is repeated if still necessary.

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has valid TMSI, CKSN and Kc.It is "idle updated" on cell B.

Foreseen final state of the MS:
The MS has no valid TMSI and no CKSN. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, TSPX_MaxRetrans, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_2, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+StartCellA(C_E_neighbourdefault, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, TSPX_MaxRetrans, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_02, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			
4	body	+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			1)
5		(TCV_Cnt:=0)			
6		REPEAT ltree_ra UNTIL [TCV_Cnt = (TSPX_MaxRetrans+1)]			2)
7		START T_dly(4000)			
8	L_0356	?TIMEOUT T_dly		(P)	3)
9		+MM_LUP_imsi(5000, MiMsi_omit, MiTmsi_01, C_MCC_1, C_PLMN_1, C_LAC_2, TCV_lac, C_normal_updating, TimingAdv(30))			4) 4+0s to 4+5 s
10		(TCV_Null := OM_ChangeRFOf2Cells(C_CellB, C_E_default, C_CellA, C_E_suitable))			5)
11		(TCV_Cnt := 0)			
12		REPEAT ltree_ra UNTIL [TCV_Cnt = (TSPX_MaxRetrans+1)]			6)
13		(TCV_Null := OM_ChangeRFOf2Cells(C_CellB, C_E_default, C_CellA, C_E_default))			7)
14		+NoReaction(15000)			8)
		ltree_ra			
15		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn, TCV_Cnt := TCV_Cnt + 1)	ChReq(ChRequest_09)		

Detailed Comments:
1) MS shall selects cell A, cell B is not available.
2) To send Max-Restrans+1 times Channel Requests in cell A
3) MS shall not try to establish a connection during a period of a cell reselection (4 seconds).
4) MS shall perform a normal location updating procedure as it is neccessary. The time difference

	between the channel request in the test step and the last channel request in the ltree_ra shall be in the range of 4-9 seconds. 5) Cell A is not available and MS shall switches to cell B 6) To send Max-Restrans+1 times Channel Requests in cell B 7) Cell A is available again. 8) MS shall not repeat the complete procedure if the original cause of the normal location updating procedure has disappeared.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_3_2

Group: GSM_L3_MS_v4220/MM/

Purpose: To verify that the MS performs normal location updating procedures when its attempt counter is smaller than 4. To check that the MS does not perform the IMSI detach procedure when "idle not updated". To verify that when "idle not updated" the MS can perform an emergency call. To verify that when "idle not updated" the MS uses requests from CM layer other than emergency call as triggering of a normal location updating procedure. To verify that the MS performs a normal location updating procedure if it enters a new cell while being "idle not updated".

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has valid TMSI and CKSN. It is "idle updated" on cell A.

Foreseen final state of the MS:
The MS has valid TMSI and CKSN. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(540)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+StartCellB(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			
4	body	+ltree_confreq1			1.
5		+ltree_confreq2			2.
6		+ltree_confreq3			3.
7		+ltree_confreq4			4.
8		+ltree_confreq6			6.
		ltree_confreq1			
9		+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
10		+MM_LupRej2(C_rc_protocolerror, MiTmsi_01, C_MCC_1, C_PLMN_1, C_LAC_1, C_normal_updating, TCV_cksn, TimingAdv(30))			1.1.
11		+NoReaction(C_T_T3211min)			1.2.
12		+MM_LupAndStop(MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TimingAdv(30), TCV_sacch_B, C_T_T3211min)			In cell B
13		+MM_LupInit(MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TimingAdv(30))			1.2.
14		+ChanRel(TCV_ch)			
15		+NoReaction(C_T_T3211min)			
16		+MM_LUPauth2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			1.2.

17		+ChanRel(TCV_ch)			
		ltree_confreq2			
18		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
19		+MM_LupRej2(C_rc_conditIEerror, MiTmsi_01, C_MCC_1, C_PLMN_1, C_LAC_2, C_normal_updating, TCV_cksn, TimingAdv(30))			1.2.
20		(TCV_Null:=OM_PgFill(C_CellA, PgReqTp1_30(MiTmsi_01)))			
21		START T_dly(8000)			
22		(TCV_Null := FALSE)			
23		REPEAT ltree_wait_or_Rej UNTIL [TCV_Null]			
24		+ltree2_send			
		ltree2_send			
25		(TCV_Null:=OM_PgFill(C_CellA, PgReqTp1Norm))			
26		START T_dly(12000)			
27		(TCV_Null := FALSE)			
28		REPEAT ltree_wait_or_Rej UNTIL [TCV_Null]			
29		+ImsiDetachNoReaction(30000, C_SIMneedRmv)			
30		+MM_PwrOrSimOn(C_SIMneedRmv)			
31		+MM_LUPauth2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			
32		+ChanRel(TCV_ch)			
		ltree_wait_or_Rej			
33	L_0357	?TIMEOUT T_dly		(P)	
34		(TCV_Null := TRUE)			
35		+MM_LupRej(C_rc_conditIEerror, C_normal_updating, TimingAdv(30))			2.3.
		ltree_confreq3			
36		+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
37		+MM_LupAuthRpt(MiTmsi_01, C_MCC_1, C_PLMN_1, C_LAC_1, C_normal_updating, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			
38		+NoReaction(3000)			
39		+MM_check_ecall2(MiImsi_01, C_cksn_nokey, TimingAdv(30))			3.2.
40		START T_dly(15000)			
41	L_0358	?TIMEOUT T_dly		(F)	3.3.
42		+ChanRel(TCV_ch)			
43		+MM_LUPauth2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			
44		+ChanRel(TCV_ch)			
		ltree_confreq4			
45		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
46		+MM_LupAndStop(MiTmsi_01, C_MCC_1, C_PLMN_1, C_LAC_2, C_normal_updating,			In cell A

		TCV_cksn, TimingAdv(30), TCV_sacch, 0)			
47		+BasicServiceMO(TSPX_MOBscSvcD, TSPX_MOChRateD)			
48		+InitCall(TCV_Service)			
49		+MM_LUP2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TimingAdv(30))			4.2.
50		START T_dly(10000)			4.3.
51	L_0359	?TIMEOUT T_dly		(P)	
52		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		4.3.
53		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
54		CANCEL T_dly			
55		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
56	L_0360	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_32(MiTmsi_01, C_cksn_nokey))	(P)	
57		ACTIVATE(OtherEventsFail)			Restore Normal default
58		+ChanRel(TCV_ch)			
		ltree_confreq6			
59		+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
60		+MM_LupInit(MiTmsi_01, C_MCC_1, C_PLMN_1, C_LAC_1, C_normal_updating, C_cksn_nokey, TimingAdv(30))			
61		L!DL_DatRqChRel START T_dly(C_T_T3211plus)	ChRel(TCV_ch, ChRelease_01)		
62	L_0361	L?DL_RelIn	DLRelInd_01	(P)	
63		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
64		+MM_LUPauth2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			6.2.
65		+ChanRel_end(TCV_ch)			
66	L_0362	?TIMEOUT T_dly		(F)	6.1.

Detailed Comments:

1. Test of Conformance Requirement 1
1.1. Reject of Location Updating in cell B
1.2. MS shall wait the period of T3211 and restart the normal location updating procedure when the attempt counter is smaller than 4.
1.3. Radio Link Failure extends the period of delay for next location updating.

2. Test of Conformance Requirement 2
2.1. Reject of Location Updating in cell A
2.2. MS shall not asnwer to paging
2.3. All of location updating requests shall be rejected.
2.4. MS shall not perform the IMSI detach procedure.

3. Test of Conformance Requirement 3
3.1. Failure during Location Updating in cell A
3.2. MS shall support emergency call.

	3.3. MS shall wait at most 15 sec. for location updating 4. Test of Conformance Requirement 4 4.1. Failure during Location Updating Procedure 4.2. MS shall use a request from Cm entity other than emergency call as a trigger for a normal location updating procedure. 4.3. After the Location Updating Procedure MS cann (optional) start automaticly the Cm entity service again. It is optional and shall observed only 15 sec. 6. Test of Conformance Requirement 6 6.1. Failure during Location Updating Procedure 6.2. MS shall start the location Updating Procedure as soon as it enters a new cell.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_3_3

Group: GSM_L3_MS_v4220/MM/

Purpose: To verify that the MS performs normal location updating after T3212 expiry when its attempt counter has reached value 4 and that the MS reset its attempt counter after a timer T3212 expiry.
To verify that the MS still follows the MM-IDLE ATTEMPTING TO UPDATE state requirements after its attempt counter has reached value 4.
To verify that the attempt counter is reset in the cases where it has to be done.

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has valid TMSI and CKSN.It is "idle updated" on cell B.

Foreseen final state of the MS:
The MS has valid TMSI and CKSN. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(1800)			
2		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_2, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+StartCellA(C_E_neighbourdefault, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_02, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			
4	body	+ltree_part1			1.
5		+ltree_part2			2.
6		+ltree_part3			3.
		ltree_part1			
7		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
8		+MM_LupRej2(C_rc_congestion, MiTmsi_01, C_MCC_1, C_PLMN_1, C_LAC_2, C_normal_updating, TCV_cksn, TimingAdv(30))			
9		+NoReaction(C_T_T3211min)			
10		+MM_LupAndStop(MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TimingAdv(30), TCV_sacch, C_T_T3211min)			
11		+MM_LupAuthRpt(MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TSPX_RANDDef, TimingAdv(30))			
12		+ltree_part1_2			
		ltree_part1_2			
13		+NoReaction(C_T_T3211min)			
14		+MM_LupInit(MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TimingAdv(30))			
15		+ChanRel(TCV_ch)			1.1.
16		+NoReaction(C_T_T3212min)			
17		+MM_LupRej2(C_rc_networkfailure, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating,			1.2.

		C_cksn_nokey, TimingAdv(30))			
18		+NoReaction(C_T_T3211min)			
19		+MM_LUPauth2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			1.3.
20		+ChanRel(TCV_ch)			
		ltree_part2			
21		+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
22		+ltree_increaseATcounter(C_LAC_1, TCV_sacch_B)			
23		+MM_LupRej2(C_rc_notidentified, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TimingAdv(30))			2.1.
24		+MM_check_ecall2(MiImsi_01, C_cksn_nokey, TimingAdv(30))			2.2.
25		+ImsiDetachNoReaction(3000, C_SIMIn)			7.
26		+MM_PwrOrSimOn(C_SIMneedRmv)			
27		+MM_LUPauth2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			
28		+ChanRel(TCV_ch)			
		ltree_part3			
29		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
30		+ltree_increaseATcounter(C_LAC_2, TCV_sacch)			
31		+MM_LupAndStop(MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TimingAdv(30), TCV_sacch, 0)			3.1.
32		+BasicServiceMO(TSPX_MOBscSvcF, TSPX_MOChRateF)			
33		+InitCall(TCV_Service)			
34		+MM_LupInit(MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TimingAdv(30))			3.2.
35		+ChanRel(TCV_ch)			
36		+ltree_part3_2			
		ltree_part3_2			
37		+NoReaction(C_T_T3211min)			
38		+MM_LUPauth2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			3.3.
39		+ChanRel(TCV_ch)			
40		START T_dly(10000)			
41		+ltree_part3_3			
42		+ltree_part3_4			
		ltree_part3_3			
43		?TIMEOUT T_dly			
44		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_02)		
45		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
46		L!DL_UdatRqImmass	ImmAss(TCV_agch,		

			ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
47	L_0363	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_32(MiTmsi_01,TCV_cksn))	(P)	
48		ACTIVATE(OtherEventsFail)			Restore Normal default
49		L!DL_DatRqCmsRej	CMSerRej(TCV_ch, CMServiceRej_30(C_rc_networkfailure))		
50		+ChanRel(TCV_ch)			
		ltree_part3_4			
51		+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
52		+ltree_increaseATcounter(C_LAC_1, TCV_sacch_B)			
53		+MM_LupRej2(C_rc_invalidmaninfo, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TimingAdv(30))			3.4.
54		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			3.5.
55		+MM_LupAndStop(MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TimingAdv(30), TCV_sacch, C_T_T3211min)			
56		+MM_LUPauth2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			3.6.
57	post	+ChanRel_end(TCV_ch)			
		ltree_increaseATcounter(lac:OCTETSTRING; par: LOGICCH)			
58		+MM_LupRej2(C_rc_notidentified, MiTmsi_01, C_MCC_1, C_PLMN_1, lac, C_normal_updating, TCV_cksn, TimingAdv(30))			
59		+NoReaction(C_T_T3211min)			
60		+MM_LupAndStop(MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TimingAdv(30), par, C_T_T3211min)			
61		+MM_LupInit(MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, C_normal_updating, C_cksn_nokey, TimingAdv(30))			
62		+ChanRel(TCV_ch)			
63		+NoReaction(C_T_T3211min)			

Detailed Comments:

1. To test conformance requirement 1
1.1. Attempt counter is now equal to 4
1.2. MS shall perform location updating after T3212
1.3. MS shall initiate location updating procedure, if the updating was unsuccessful.

2.) To test conformance requirement 2
2.1. Attempt counter is now equal to 4
2.2. MS shall perform request for emergency call
2.3. MS shall not perform an IMSI detach procedure

3.) To test conformance requirement 3 and 4
3.1. Attempt counter is now equal to 4
3.2. MS shall use a request from CM entity for MM connection for a service other than emergency

	call as a trigger for a normal location updating procedure. 3.3. MS shall reset the attempt counter after successful location updating procedure. 3.4. Attempt counter is now equal to 4 3.5. MS shall perform the normal location updating procedure after entering a new cell. 3.6. If the location updating procedure unssuccessful, the MS shall trigger the location updating after T3211 again.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_3_4

Group: GSM_L3_MS_v4220/MM/

Purpose: To verify that in the case when the attempt counter is smaller than 4 and the broadcast LAI is equal to the stored LAI, the MS is in the MM-IDLE state and NORMAL SERVICE substate. To verify that timer T3211 is stopped after a MM connection establishment. To verify that the MS uses the T3211 timer. and that it enters the MM-IDLE state and NORMAL SERVICE substate when its attempt counter reaches value

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has valid TMSI and CKSN.It is "idle updated" on cell B.

Foreseen final state of the MS:
The MS has valid TMSI and CKSN. It is "idle updated" on cell B.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(2700)			
2		+IdleUpdated(C_E_neighbourdefault, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_2, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_01, BcchFreqLst_48, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+ltree_confreq1			1.
4		+ltree_confreq2			2.
5		+ltree_confreq3			3.
6		+ltree_confreq4			4.
7		+ltree_confreq5			5.
8		+ltree_confreq6			6.
		ltree_confreq1			
9		(TCV_tmp := (C_T_T3212 + 45000))			
10		+MM_LUPperrej2(C_rc_networkfailure, MiTmsi_01, TCV_tmp, C_MCC_1, C_PLMN_1, TCV_lac, C_periodic_updating, TCV_cksn, TimingAdv(30))			1.1.
11		+BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
12		+InitCM(TCV_Service)			
13		+ltree_cmservinit(MiTmsi_01, TCV_cksn)			1.2.
14		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
15		+ltree_CMmsg			
16		+ChanRel(TCV_ch)			
17		(TCV_n := 2 * C_T_T3211min)			
18		+NoReaction(TCV_n)			1.3.
19		+ImsiDetach(MiTmsi_01, TimingAdv(30), C_SIMneedRmv)			
		ltree_confreq2			
20		+ltree_MOCall			
21		+NoReaction(TCV_n)			2.3.
22		+ImsiDetach(MiTmsi_01, TimingAdv(30), C_SIMneedRmv)			
23		+MM_PwrOrSimOn(C_SIMneedRmv)			
24		+MM_LupInit(MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_imsi_attach, TCV_cksn, TimingAdv(30))			
25		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1,		

			C_PLMN_1, TCV_lac))		
26		+ChanRel(TCV_ch)			
		ltree_confreq3			
27		(TCV_tmp := (C_T_T3212 + 15000))			
28		+MM_LUPperrej3(MiTmsi_01, TCV_tmp, C_MCC_1, C_PLMN_1, TCV_lac, C_periodic_updating, TCV_cksn,TimingAdv(30))			3.1.
29		+NoReaction(C_T_T3211min)			
30		+MM_LupRej2(C_rc_networkfailure, MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_periodic_updating, TCV_cksn, TimingAdv(30))			3.1.
31		+NoReaction(C_T_T3211min)			
32		+MM_LupAndStop(MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_periodic_updating, TCV_cksn, TimingAdv(30), TCV_sacch_B, C_T_T3211min)			3.1.
33		+ltree_luprej3(MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_periodic_updating, TCV_cksn, TimingAdv(30))			3.1.
34		+MM_LUPperauth(MiImsi_01, MiTmsi_01iei, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normalOrperiodic, C_cksn_nokey, TSPX_RANDDef, TimingAdv(30))			3.2.
35		+ChanRel(TCV_ch)			
		ltree_confreq4			
36		+NoReaction(C_T_T3212min)			
37		+MM_LupRej2(C_rc_networkfailure, MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_periodic_updating, TCV_cksn, TimingAdv(30))			
38		+NoReaction(C_T_T3211min)			
39		+MM_LupAndStop(MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_periodic_updating, TCV_cksn, TimingAdv(30), TCV_sacch_B, C_T_T3211min)			
40		+ltree_luprej3(MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_periodic_updating, TCV_cksn, TimingAdv(30))			
41		+NoReaction(C_T_T3211min)			
42		+ltree_confreq4_continue			
		ltree_confreq4_continue			
43		+MM_LupRej2(C_rc_networkfailure, MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_periodic_updating, TCV_cksn, TimingAdv(30))			4.1.
44		+BasicServiceMO(TSPX_MOBscSvcB, TSPX_MOChRateB)			
45		+InitCM(TCV_Service)			
46		+MM_LUP2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TimingAdv(30))			4.2.
47		START T_dly(10000)			
48		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_02)		4.3.
49		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
50		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
51	L_036	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_32(	(P)	

	4		MiTmsi_01, C_cksn_nokey))		
52		ACTIVATE(OtherEventsFail)			Restore Normal default
53		L!DL_DatRqCmsRej	CMSerRej(TCV_ch, CMServiceRej_04)		
54		+ChanRel(TCV_ch)			
55	L_0365	?TIMEOUT T_dly		(P)	4.3)
		ltree_confreq5			
56		+ImsiDetach(MiTmsi_01, TimingAdv(30), C_SIMneedRmv)			
57		+MM_PwrOrSimOn(C_SIMneedRmv)			
58		+MM_LupAndStop(MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_imsi_attach, C_cksn_nokey, TimingAdv(30), TCV_sacch_B, C_T_T3211min)			5.1.
59		+ltree_luprej3(MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_imsi_attach, C_cksn_nokey, TimingAdv(30))			5.1.
60		+NoReaction(C_T_T3211min)			
61		+MM_LupRej2(C_rc_networkfailure, MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_imsi_attach, C_cksn_nokey, TimingAdv(30))			5.1.
62		+NoReaction(C_T_T3211min)			
63		+MM_LupAndStop(MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_imsi_attach, C_cksn_nokey, TimingAdv(30), TCV_sacch_B, 0)			5.2.
64		+MM_LUPperauth(MiImsi_01, MiTmsi_01iei, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_norm_period_attach, C_cksn_nokey, TSPX_RANDDef, TimingAdv(30))			5.3.
65		+ChanRel(TCV_ch)			
		ltree_confreq6			
66		+ImsiDetach(MiTmsi_01, TimingAdv(30), C_SIMneedRmv)			
67		+MM_PwrOrSimOn(C_SIMneedRmv)			
68		+ltree_luprej3(MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_imsi_attach, TCV_cksn, TimingAdv(30))			
69		+NoReaction(C_T_T3211min)			
70		+MM_LupRej2(C_rc_networkfailure, MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_imsi_attach, TCV_cksn, TimingAdv(30))			
71		+NoReaction(C_T_T3211min)			
72		+MM_LupAndStop(MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_imsi_attach, TCV_cksn, TimingAdv(30), TCV_sacch_B, C_T_T3211min)			
73		+ltree_luprej3(MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_imsi_attach, TCV_cksn, TimingAdv(30))			6.1.
74		+BasicServiceMO(TSPX_MOBscSvcC, TSPX_MOChRateC)			
75		+InitCM(TCV_Service)			
76		+MM_LUPauth2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			6.2.
77		+ChanRel(TCV_ch)			
78		+ltree_cmservinit(MiTmsi_01, TCV_cksn)			

79		L!DL_DatRqCmsRej	CMSerRej(TCV_ch, CMServiceRej_04)		
80		+ChanRel(TCV_ch)			
		ltree_cmservinit(par1: MI; par2: BITSTRING)			
81		START T_dly(15000)			
82		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_02)		
83		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
84		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
85	L_0366	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_32(par1, par2))	(P)	
86		ACTIVATE(OtherEventsFail)			Restore Normal default
87	L_0367	?TIMEOUT T_dly		(F)	
		ltree_CMmsg			
88	L_0368	L?DL_DatInSetup (TCV_Mt := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)	(P)	1.2.
89	L_0369	L?DL_DatInRegister	Register_03(RegisterPdu_01)	(P)	1.2.
90		L?DL_EstIn	DLEstInd_2(TCV_ch)		
91	L_0370	L?DL_DatInCpData	DatInCpData(TCV_ch, CpDataPdu_any)	(P)	1.2.
		ltree_luprej3(par_mi:MI; mcc, mnc, lac:OCTETSTRING; locup:B_2; cksn:BITSTRING; ta:TA)			
92		+MM_LupInit(par_mi, mcc, mnc, lac, locup, cksn, ta)			
93		+ChanRel(TCV_ch)			
		ltree_MOCall			
94		+MM_PwrOrSimOn(C_SIMneedRmv)			
95		+MM_LupAndStop(MiTmsi_01, C_MCC_1, C_PLMN_1, TCV_lac, C_imsi_attach, TCV_cksn, TimingAdv(30), TCV_sacch_B, 0)			2.1)
96		+BasicServiceMO(TSPX_MOBscSvcD, TSPX_MOChRateD)			
97		+InitCM(TCV_Service)			
98		+ltree_cmservinit(MiTmsi_01, TCV_cksn)			2.2)
99		+Ciphering_on(TCV_ch)			
100		+ltree_CMmsg			
101		+ChanRel(TCV_ch)			
102		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			

Detailed Comments:

1. Test of Conformance Requirement 1

1.1. failure during a periodic location updating procedure

1.2. - then the MS shall be able to attempt an CM connection

1.3. - then the MS shall not attempt a location updating procedure

2. Test of Conformance Requirement 2

2.1. failure during imsi attach procedure

2.2. - then the MS shall be able to establish an MM connection

2.3. - then the MS shall not attempt a location updating procedure

3. Test of Conformance Requirement 3

3.1. When a failure during a periodic location updating procedure and the attempt counter is smaller than 4 then the MS shall execute a periodic location updating procedure after T3211 expiry.

	3.2. When the attempt counter reaches 4 after T3212 expiry the MS shall make a LUP any type. 4. Test of Conformance Requirement 4 4.1. The attempt counter reaches the value 4 4.2. then the MS shall use a request from CM layer for an emergency call as a trigger for a LUP. 4.3. this part is optional 5. Test of Conformance Requirement 5 5.1. When a failure during an imsi attach procedure and the attempt counter is smaller than 4 then the MS shall execute a location updating procedure(imsi attach) after T3211expiry. 5.2. Attempt Counter reaches the value 4 5.3. When the attempt counter reaches 4 after T3212 expiry the MS shall make a normal LUP. 6. Test of Conformance Requirement 6 6.1. Failure during an imsi attach procedure and the attempt counter reaches the value 4. 6.2. then the MS shall use a request from CM layer for an emergency call as a trigger for a LUP.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_4
Group: GSM_L3_MS_v4220/MM/
Purpose: To verify that the MS aborts the RR-connection at the expiry of timer T3240.
Default: OtherEventsFail
Comments: Initial Conditions of MS:
The MS has valid TMSI and CKSN.It is "idle updated" on cell A.

Foreseen final state of the MS:
The MS has valid TMSI and CKSN. It is "idle updated" on cell B.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+StartCellB(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			
4	body	+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_09)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
8		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_04(C_normal_updating, MiTmsi_01))		
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		+NoReaction(C_T_T3240min)			
11		START T_dly(C_T_T3240tol)			
12	L_0371	L?DL_RelIn CANCEL T_dly	DLRelInd_01	P	1)
13	L_0372	?TIMEOUT T_dly		F	

Detailed Comments: 1) MS shall release the L2-Connection after expiring of T3240.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_5_1

Group: GSM_L3_MS_v4220/MM/

Purpose: 1) To check that when the location updating timer is reduced, the timer running in the MS is started with a value depending on the current timer value and the new broadcasted T3212 value.

2) To verify that when the MS is reactivated in the same cell (as the one in which it was deactivated), IMSI attach being forbidden, the MS starts the timer T3212 with a value between zero and the broadcasted value

Default: OtherEventsFail

Comments: Initial Conditions of MS:

The MS is deactivated. The stored MCC, MNC and LAC correspond to the broadcasted values. The stored update status is "updated".

Foreseen final state of the MS:

The MS has valid TMSI and CKSN. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(1200)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_5, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_TmsiOff, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			ATT = 1 - IMSI attach allowed
3	body	+ImsiAttach(MiTmsi_01, TimingAdv(30), C_MCC_1, C_PLMN_1, C_SIMneedRmv)			
4		+step1			
5		+step2			
		step1			
6		START T_dly(180000)			
7		START T_dly2(C_T_T3212min)			5min.45s.
8		?TIMEOUT T_dly			
9		+StartCellA(C_E_default, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_02, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			1)
10		?TIMEOUT T_dly2			2)
11		+MM_LUPper2(30000, C_periodic_updating, C_MCC_1, C_PLMN_1, TCV_lac, TimingAdv(30))			6min.15.
12		+ChanRel(TCV_ch)			
		step2			
13		+StartCellA(C_E_default, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_1, C_LAC_1,CellChDes_02, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			ATT = 0
14		START T_dly(5000)			
15		?TIMEOUT T_dly			
16		+MM_PwrOrSimOff(C_SIMneedRmv)			
17		+MM_PwrOrSimOn(C_SIMneedRmv)			
18		+MM_LUPper2(420000, C_periodic_updating, C_MCC_1,			3)

		C_PLMN_1, TCV_lac, TimingAdv(30) +ChanRel_end(TCV_ch)			
19					

Detailed Comments:

1) Reducing of the timer T3212
2) MS shall use the new value of T3212 for periodic updating.
3) MS shall start the periodic location updating between 0 sec. and 6min after switching on of MS.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_5_2

Group: GSM_L3_MS_v4220/MM/

Purpose: To verify that the MS stops and resets the timer T3212 of the periodic location updating procedure when:

- the first MM-message is received in the case of MM-connection establishment, ciphering mode being not set,
- the MS has responded to paging and the first correct L3 message is received that is not an RR message

Default: OtherEventsFail

Comments: Initial Conditions of MS:

The MS has valid TMSI and CKSN.It is "idle updated" on cell A.

Foreseen final state of the MS:

The MS has valid TMSI and CKSN. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(1800)			
2		+BasicServiceMO(TSPX_MOBscSvcF, TSPX_MOChRateF)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_2, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4	body	+InitCall(TCV_Service)			
5		START T_dly(705000)			
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
9	L_0373	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)	(P)	
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		L!DL_DatRqCmsRej	CMSerRej(TCV_ch, CMServiceRej_30(C_rc_networkfailure))		
12		+ChanRel(TCV_ch)			
13		?TIMEOUT T_dly			
14		+local_continue			
		local_continue			
15		+MM_LUPper2(30000, C_periodic_updating, C_MCC_1, C_PLMN_1, TCV_lac, TimingAdv(30))			
16		+ChanRel(TCV_ch)			
17		START T_dly(60000)			
18		?TIMEOUT T_dly			
19		+ltree_continue1			
		ltree_continue1			
20		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg,		

			PgReqTp1_30(MiImsi_01))		
21		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
22		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
23		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
24		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
25		ACTIVATE(OtherEventsFail)			Restore Normal default
26		+Authentication(TCV_ch, TCV_cksn, TSPX_RANDDef)			1.
27		+ChanRel(TCV_ch)			2.
28		START T_dly(705000)			
29		?TIMEOUT T_dly			
30		+MM_LUPper2(30000, C_periodic_updating, C_MCC_1, C_PLMN_1, TCV_lac, TimingAdv(30))			
31		+ChanRel_end(TCV_ch)			

Detailed Comments:
1. MS shall stop the timer T3212 after receiving of the first L3-message
2. MS shall reset and restart the timer T3212

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_5_3

Group: GSM_L3_MS_v4220/MM/

Purpose: To verify that the MS stops and resets the timer T3212 of the periodic location updating procedure when a Location Updating Accept or a Location Updating Reject message is received.

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has valid TMSI and CKSN.It is "idle updated" on cell A.

Foreseen final state of the MS:
The MS has valid TMSI and CKSN. It is "idle updated" on cell B.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(1200)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_2, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3		+StartCellB(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_2, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			
4	body	+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
5		+MM_LUP3(C_MCC_1, C_PLMN_2, TCV_lac, C_normal_updating, TimingAdv(30))			1)
6		+NoReaction(345000)			
7		+MM_LUPper2(30000, C_periodic_updating, C_MCC_1, C_PLMN_2, TCV_lac, TimingAdv(30))			
8		+ChanRel(TCV_ch)			
9		+ImsiDetach(MiTmsi_01, TimingAdv(30), C_SIMneedRmv)			2)
10		+ImsiAttach(MiTmsi_01, TimingAdv(30), C_MCC_1, C_PLMN_2, C_SIMneedRmv)			
11		+NoReaction(345000)			
12		+MM_LUPper2(30000, C_periodic_updating, C_MCC_1, C_PLMN_2, TCV_lac, TimingAdv(30))			
13		+ChanRel_end(TCV_ch)			

Detailed Comments: 1) MS shall reset the timer T3212 after normal location updating
2) MS shall reset the timer T3212 after IMSI attach procedure.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_5_4_1

Group: GSM_L3_MS_v4220/MM/

Purpose: To verify that when a cell of the HPLMN becomes available, following the successful location request on the VPLMN of the home country and after the first search the mobile has failed to find its HPLMN, that the MS shall perform a location update request on the HPLMN after time T. Were T is the HPLMN Search Period stored in the SIM.

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS is switched off. The HPLMN Search Period on the SIM shall be set to 6 minutes.
The Location Area Information on the SIM is deleted.

Foreseen final state of the MS:
The MS has valid TMSI and CKSN. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(1020)			
2		(TCV_Null := OO_SIM2Ins())			
3		+IdleUpdated(C_E_neighbourdefault, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_2, C_LAC_2, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_01, BcchFreqLst_48, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_Start_Rej, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1)
4	body	+MM_PwrOrSimOn(C_SIMneedRmv)			
5		+MM_LupInit5(C_normal_updating, TimingAdv(30))			
6		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1, C_PLMN_2, TCV_lac))		PLMN2
7		+ChanRel(TCV_ch)			
8		+NoReaction(480000)			
9		+ltree_continue			
		ltree_continue			
10		+StartCellA(C_E_default, C_Immass,TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_02, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			
11		+Varinit_fix(C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
12		START T_dly(480000)			
13	L_0374	?TIMEOUT T_dly		(F)	
14		+MM_LupInit4(C_normal_updating, TimingAdv(30))			
15	L_0375	L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1, C_PLMN_1, TCV_lac))	(P)	PLMN1
16		+ChanRel_end(TCV_ch)			

Detailed Comments: 1) Initial condition: LAI deleted, HPLMNsearchperiod=6min

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_5_4_2
Group: GSM_L3_MS_v4220/MM/
Purpose: To verify that no HPLMN Search is performed when the MS is not in automatic mode
Default: OtherEventsFail
Comments: Initial Conditions of MS:
The MS is switched off. The HPLMN Search Period on the SIM shall be set to 6 minutes.
The Location Area Information on the SIM is deleted.

Foreseen final state of the MS:
The MS has valid TMSI and CKSN. It is "idle updated" on cell B.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(420)			
2		(TCV_Null := OO_SIM2Ins())			
3		+IdleUpdated(C_E_neighbourdefault, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_2, C_LAC_2, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_01, BcchFreqLst_48, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_Start_Rej, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1)
4	body	+MM_PwrOrSimOn(C_SIMneedRmv)			
5		+MM_LupInit5(C_normal_updating, TimingAdv(30))			
6		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1, C_PLMN_2, TCV_lac))		PLMN2
7		+ChanRel(TCV_ch)			
8		+ltree_continue			
		ltree_continue			
9		(TCV_Null := OO_PLMNselModeMan())			2)
10		+StartCellA(C_E_default, C_Immass,TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_02, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			3)
11		+Varinit_fix(C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
12		+NoReaction(420000)			

Detailed Comments:
1) Initial condition: LAI deleted, HPLMN searchperiod = 6min
2) MS in manual mode.
3) made cell A available.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_5_4_3

Group: GSM_L3_MS_v4220/MM/

Purpose: To verify that the MS waits at least 2 minutes and at most T minutes before attempting its first HPLMN Search.

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS is switched off. The HPLMN Search Period on the SIM shall be set to 6 minutes.
The Location Area Information on the SIM is deleted.

Foreseen final state of the MS:
The MS has valid TMSI and CKSN. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(480)			
2		(TCV_Null := OO_SIM2Ins())			
3		+IdleUpdated(C_E_neighbourdefault, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_2, C_LAC_2, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_01, BcchFreqLst_48, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_Start_Rej, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1)
4	body	+MM_PwrOrSimOn(C_SIMneedRmv)			
5		+MM_LupInit5(C_normal_updating, TimingAdv(30))			
6		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1, C_PLMN_2, TCV_lac))		PLMN2
7		+ChanRel(TCV_ch)			
8		+ltree_continue			
		ltree_continue			
9		+StartCellA(C_E_default, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_02, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			2)
10		+Varinit_fix(C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
11		+NoReaction(120000)			
12		START T_dly(360000)			3)
13	L_0376	?TIMEOUT T_dly		(F)	
14		+MM_LupInit6(C_normal_updating, TimingAdv(30))			
15		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1, C_PLMN_1, TCV_lac))		PLMN1
16		+ChanRel_end(TCV_ch)			

Detailed Comments: 1) Initial condition: LAI deleted, HPLMNsearchperiod=6min
2) made cell A available.

3) MS shall send the messages on cell A within T minutes. T means HPLMN search period.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_4_6

Group: GSM_L3_MS_v4220/MM/

Purpose: 1) To check that if the PLU timer expires while the MS is out of coverage, the MS informs the network of its return to coverage.

2) To check that the PLU timer is not disturbed by cells of forbidden PLMNs.

3) To check that if the PLU timer does not expire while out of coverage and if the mobile returns to the LA where it is updated, the mobile does not inform the network of its return to coverage.

Default: OtherEventsFail

Comments: Initial Conditions of MS:

The MS is deactivated. The PLMN of cell B is entered in the SIM's forbidden PLMN list.

Foreseen final state of the MS:

The MS is "idle updated". The PLMN of cell B is entered in the SIM's forbidden PLMN list.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(2100)			
2		+IdleUpdated(C_E_neighbourdefault, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_2, C_LAC_2, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_PLMNnot, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			PR 1)
3		+StartCellA(C_E_default, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_02, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			
4		+Varinit_fix(C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
5		+MM_LUP(MiTmsi_01iei, C_MCC_1, C_PLMN_1, C_LAC_1, C_normal_updating, TimingAdv(30))			PR 2)
6		+MM_PwrOrSimOff(C_SIMneedRmv)			
7		+ltree_StartCellAandB			
8	body	+ltree_body1			
9		+ltree_body2			
10		+ltree_body3			
11		+RmvForbiddenPlmn(C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			
		ltree_body1			
12		+MM_PwrOrSimOn(C_SIMneedRmv)			1)
13		START T_dly1(C_T_Wait1stChReq)			
14		+ltree_perluprest(C_imsi_attach)			2)
15		START T_dly1(705000)			3)
16		+NoReaction(60000)			
17		(TCV_Null := OM_StopCell(C_CellA))			
18		+NoReaction(420000)			
19		+ltree_StartCellAandB			
20	L_0377	?TIMEOUT T_dly1		(P)	3)

		ltree_body2			
21		START T_dly1(30000)			
22		+ltree_perluprest(C_periodic_updating)			4)
23		+NoReaction(180000)			
24		(TCV_Null := OM_StopCell(C_CellA))			
		ltree_body3			
25		+NoReaction(660000)			5)
26		+ltree_StartCellAandB			
27		(TCV_Null := OM_StopCell(C_CellB))			
28		START T_dly1(180000)			
29		+ltree_perluprest(C_periodic_updating)			6)
		ltree_perluprest(locup:B_2)			
30	L_0378	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly1	ChReq(ChRequest_02)	(P)	
31		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
32		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
33	L_0379	L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(locup))	(P)	
34		ACTIVATE(OtherEventsFail)			Restore Normal default
35		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, C_MCC_1, C_PLMN_1, TCV_lac))		
36		+ChanRel(TCV_ch)			
37	L_0380	?TIMEOUT T_dly1		(F)	
		ltree_StartCellAandB			
38		+StartCellA(C_E_default, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_2, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_01, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			
39		+StartCellB(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, C_LAC_2, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			

Detailed Comments:

PR1) Initial condition: The PLMN of cell B is entered in the SIM's forbidden PLMN list.
PR2) Initial condition: Idle updated in cell A.

1) MS is in automatic network selection mode.
2) IMSI attach procedure
3) Conformance Requirement 1
Delay for 11 min and 40 seconds. In this time MS may not location update in cell A.
4) The MS shall execute the periodic location updating after 12 min.
5) Conformance Requirement 5
No Reaction in 7 min. In this time MS may not location update in cell B.
6) The MS shall execute the periodic location updating before 17 min after last periodic LUP.

PS1) Final postamble: The PLMN of cell B is supressed on the SIM's forbidden PLMN list.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_5_2

Group: GSM_L3_MS_v4220/MM/

Purpose: To verify that the MS can correctly set up an MM connection in an originating CM connection establishment when ciphering mode setting is not required.

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has valid TMSI. It is "idle updated".
Expected values in the SIM card: TMSI: MiTmsi_01, CKSN: TSPX_CKSNDef

Foreseen final state of the MS:
The MS has valid TMSI and CKSN.It is "idle updated".

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcH, TSPX_MOChRateH)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+InitCM(TCV_Service)			
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
8	L_0381	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_32(MiTmsi_01, TSPX_CKSNDef))	(P)	
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		+Authentication(TCV_ch, TCV_cksn, TSPX_RANDDef)			
11		+Ciphering_on(TCV_ch)			
12		+Itree_CMmsgs			
13		+ChanRel_end(TCV_ch)			
		Itree_CMmsgs			
14		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
15	L_0382	L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_04(TCV_TI))	(P)	
16		L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI.ti_f := '1'B)	Register_01(TCV_ch, RegisterPdu_01)		
17		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_08(TCV_TI))		
18		L?DL_EstIn	DLEstInd_2(TCV_ch)		
19		L?DL_DatInCpData (TCV_Rpmr := DL_DatInCpData.msg.CPdata.rpdata.rpmr,	DatInCpData(TCV_ch		

		TCV_TI0 := DL_DatInCpData.msg.ti, TCV_TI.ti_v := TCV_TI0.ti_v, TCV_TI.ti_f := '1'B)	, CpDataPdu_04)		
20		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_03(TCV_TI), TCV_ch)		
21		L!DL_DatRqCpData	DatRqCpData(CpDataPdu_02(TI_08(TCV_TI.ti_v), CpData_04(TCV_Rpmr)), TCV_ch)		
22	L_0383	L?DL_DatInCpDataAck	DatInCpDataAck(CpDataAckPdu_02(TI_07(TCV_TI0.ti_v)))	(P)	
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_5_3
Group: GSM_L3_MS_v4220/MM/
Purpose: To verify that the MS can correctly set up an MM connection in an originating CM connection establishment when ciphering mode setting is not required.
Default: OtherEventsFail
Comments: Initial Conditions of MS:
The MS has valid TMSI. It is "idle updated".

Foreseen final state of the MS:
The MS has valid TMSI. It is "idle updated".

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcG, TSPX_MOChRateG)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+InitCM(TCV_Service)			
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
8	L_0384	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)	(P)	
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
11		+ltree_msgs			
12		+ChanRel_end(TCV_ch)			
		ltree_msgs			
13	L_1042	L?DL_DatInSetup (TCV_Mt := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)	(P)	
14	L_0385	L?DL_DatInRegister	Register_01(TCV_ch, RegisterPdu_01)	(P)	
15		L?DL_EstIn	DLEstInd_2(TCV_ch)		
16		L?DL_DatInCpData (TCV_Rpmr := DL_DatInCpData.msg.CPdata.rpdata.rpmr, TCV_TI0 := DL_DatInCpData.msg.ti, TCV_TI := TCV_TI0, TCV_TI.ti_f := '1'B)	DatInCpData(TCV_ch, CpDataPdu_04)		
17		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_03(TCV_TI), TCV_ch)		
18		L!DL_DatRqCpData	DatRqCpData(CpDataPdu_02(TI_08(TCV_TI.ti_v), CpData_04(TCV_Rpmr)), TCV_ch)		
19	L_038	L?DL_DatInCpDataAck	DatInCpDataAck(	(P)	

	6		CpDataAckPdu_02(TI_07(TCV_TI0.ti_v)))		

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_5_4
Group: GSM_L3_MS_v4220/MM/
Purpose: To verify that the MS does not send a layer 3 message when the service request is rejected by the SS.
Default: OtherEventsFail
Comments: Initial Conditions of MS:
The MS has valid TMSI. It is "idle updated".

Foreseen final state of the MS:
The MS has valid TMSI. It is "idle updated".

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(420)			
2		+BasicServiceMO(TSPX_MOBscSvcB, TSPX_MOChRateB)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+InitCall(TCV_Service)			
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
8	L_0387	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)	(P)	
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		L!DL_DatRqCmsRej	CMSerRej(TCV_ch, CMServiceRej_30(C_rc_reqservoptnotsub))		
11		+NoReaction(5000)			
12		+ChanRel_end(TCV_ch)			

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_5_5

Group: GSM_L3_MS_v4220/MM/

Purpose: To verify that the MS can correctly accept a CM SERVICE REJECT message with reject cause "IMSI unknown in VLR".

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has valid TMSI. It is "idle updated".
Expected values in the SIM card: TMSI: MiTmsi_01, CKSN: TSPX_CKSNDef

Foreseen final state of the MS:
The MS has valid TMSI and CKSN. It is "idle updated".

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcE, TSPX_MOChRateE)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+InitCall(TCV_Service)			
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
8	L_0388	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_32(MiTmsi_01, TSPX_CKSNDef))	(P)	1.
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		L!DL_DatRqCmsRej	CMSerRej(TCV_ch, CMServiceRej_30(C_rc_imsiunknownvlr))		
11		+ChanRel(TCV_ch)			2.
12		+MM_LUPauth2(MiTmsi_01iei,MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			3.
13		+ChanRel_end(TCV_ch)			

Detailed Comments:
1. Conformance Requirement
The MS shall be able to correctly set up an MM connection in a Mobile Originating CM connection attempt.
2. Conformance Requirement
The MS shall wait for the network to release the RR connection
3. Conformance Requirement
The MS shall be able to perform a location updating procedure.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_5_6
Group: GSM_L3_MS_v4220/MM/
Purpose: To verify that at T3230 expiry, the MS aborts the MM-connection establishment.
Default: OtherEventsFail
Comments: Initial Conditions of MS:
The MS has valid TMSI. It is "idle updated".

Foreseen final state of the MS:
The MS has valid TMSI. It is "idle updated".

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcJ, TSPX_MOChRateJ)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+InitCall(TCV_Service)			
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
8	L_0389	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)	(P)	
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		+NoReaction(C_T_T3230min)			
11	L_0390	L?DL_DatInMmst	MMSt(MMstatus_03)	(P)	1.
12		+ChanRel_end(TCV_ch)			

Detailed Comments: 1. The MS shall abort the MM connection after T3230 expiry.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_5_7_1

Group: GSM_L3_MS_v4220/MM/

Purpose: To check that upon reception of an ABORT message with cause #6 during call establishment:
- the MS does not send any layer 3 message.
- after reception of an ABORT message and after having been deactivated and reactivated, the MS performs location updating using its IMSI as mobile identity and indicates deleted LAI and CKSN.
- the MS does not perform location updating, does not answer to paging with TMSI, rejects any request for mobile originating call except emergency call, does not perform IMSI detach.
- the MS accepts a request for emergency call.

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has valid TMSI, CKSN and Kc.It is "idle updated on cell B".

Foreseen final state of the MS:
The MS has valid TMSI. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_2, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+StartCellA(C_E_neighbourdefault, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_02, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			
5		+InitCall(TCV_Service)			
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(30)))		
9	L_0391	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)	(P)	
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		+Authentication(TCV_ch, TCV_cksn, TSPX_RANDDef)			
12		L!DL_DatRqAbrt	Abort(TCV_ch, Abortmsg_01(C_rc_illegal_me))		
13		+ltree_continue			
		ltree_continue			
14		+NoReaction(5000)			1.
15		+ChanRel(TCV_ch)			

16		+Switchcell(C_CellB, C_CellA, C_LAC_1, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TCV_Ccd0A, TSPX_IMSI)			
17		+NoReaction(5000)			2.1.
18		+NoReaction(420000)			2.2.
19		+MM_no_paging(MiTmsi_01, 3000, TCV_Ccd0B, C_CellB)			2.3.
20		+ltree_continue1			
		ltree_continue1			
21		+SelectPagingCh(C_CellA)			re-select active cell
22		+MM_no_cmservices(3000)			2.4.
23		+MM_check_ecall1(TimingAdv(30), MiImei_01)			3.
24		+ImsiDetachNoReaction(3000, C_SIMneedRmv)			7.
25		+MM_PwrOrSimOn(C_SIMneedRmv)			
26		+MM_LUPauth2(MiTmsi_01iei, MiImsi_01, C_MCC_1, C_PLMN_1, C_lacdeleted, TCV_lac, C_normal_updating, C_cksn_nokey, TCV_cksn, TSPX_RANDDef, TimingAdv(30))			4.
27		+ChanRel(TCV_ch)			

Detailed Comments: After ABORT with the cause_value 'illegal_me' the MS shall

1. Conformance requirement 1:
   MS shall wait for network command.
2. Conformance requirement 2:
   2.1. not perform location updating
   2.2. not perform periodic location updating
   2.3. not respond to paging with TMSI
   2.4. reject any request for MOC establishment with except emergency call.
   2.5. not perform IMSI detach.
3. Conformance requirement 3:
   if it support speech, perform emergency call.
4. Conformance requirement 4:
   delete stored LAI, CKSN and TMSI

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_5_7_2

Group: GSM_L3_MS_v4220/MM/

Purpose: To check that when multiple MM connections are established, the MS releases all MM connections upon reception of an ABORT message, in the case when the two MM connections are established for a mobile terminating call and a non call related supplementary service operation.

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS is in state U10 of a mobile terminating call.

Foreseen final state of the MS:
The MS has valid TMSI. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcC, TSPX_MTChRateC)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU10_late(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), TSPX_CKSNDef, TSPX_RANDDef)			
6	body	+InitNonCallSupp			1.
7	L_0392	L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)	(P)	
8		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
9		L?DL_DatInRegister	Register_01(TCV_chTch, RegisterPdu_01)		
10		L!DL_DatRqAbrt	Abort(TCV_chTch, Abortmsg_01(C_rc_networkfailure))		2.
11		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TI_02, Cause_01, ProgInd_omit, UuInfo_omit))		
12	L_0393	L?DL_DatInRelCmp	RelComRcv(ReleaseCmpRcv(TI_01, Cause_07))	(P)	
13		+ChanRel_end(TCV_chTch)			

Detailed Comments:
1. Initiate a non call related supplementary service.
2. Upon reception of an ABORT-message the MS shall release any ongoing MM connection.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_5_8_1

Group: GSM_L3_MS_v4220/MM/

Purpose: To check that when the network does not include the follow on proceed IE in a LOCATION UPDATING ACCEPT message, a MS that has a CM application request pending does not attempt to establish a new MM connection on that RR connection.

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has valid TMSI and is deactivated.

Foreseen final state of the MS:
The MS has valid TMSI. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_TmsiOff, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			att=1
3	body	+ImsiAttachIni(MiTmsi_01, TimingAdv(30), C_MCC_1, C_PLMN_1, C_SIMneedRmv)			
4		+BasicServiceMO(TSPX_MOBscSvcF, TSPX_MOChRateF)			
5		+InitCall(TCV_Service)			
6		?TIMEOUT T_dly			
7		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp(C_MCC_1, C_PLMN_1, TCV_lac))		1.
8		+NoReaction(8000)			2.
9	post	+ChanRel_end(TCV_ch)			

Detailed Comments:
1. Follow on proceed IE not included.
2. MS shall not send any layer 3 message for 8 seconds after reception of Location Updating Acc.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_5_8_2

Group: GSM_L3_MS_v4220/MM/

Purpose: To check that when the network includes the follow on proceed IE in a LOCATION UPDATING ACCEPT message, a MS that supports the follow on request procedure and that has a CM application request pending establishes successfully a new MM connection on that RR connection.

Default: OtherEventsFail

Comments: Initial Conditions of MS:
The MS has valid TMSI and is deactivated.

Foreseen final state of the MS:
The MS has valid TMSI. It is "idle updated" on cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_TmsiOff, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			att=1
3	body	+ImsiAttachIni(MiTmsi_01, TimingAdv(30), C_MCC_1, C_PLMN_1, C_SIMneedRmv)			
4		+BasicServiceMO(TSPX_MOBscSvcI, TSPX_MOChRateI)			
5		+InitCM(TCV_Service)			
6		?TIMEOUT T_dly			
7		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_02(C_MCC_1, C_PLMN_1, TCV_lac))		1.
8		[NOT TSPC_followOnReq]			
9		+NoReaction(8000)			
10		+ChanRel_end(TCV_ch)			
11		[TSPC_followOnReq]			
12		+CMsrvcRq			
13	post	+ChanRel_end(TCV_ch)			

Detailed Comments: 1. Follow on proceed IE included.

Test Case Dynamic Behaviour

Test Case Name: TC_26_7_5_8_3

Group: GSM_L3_MS_v4220/MM/

Purpose:
1) To check that a MS that has no CM application request pending sets the Follow-On-Request bit to No follow-on request pending in a LOCATION UPDATING REQUEST message.
2) To check that when the network includes the follow on proceed IE in a LOCATION UPDATING ACCEPT message, a MS that has no CM application request pending does not attempt to establish a new MM connection on that RR connection.
3) To check that the MS accepts establishment by the network of a new MM connection on the existing RR connection.

Default: OtherEventsFail

Comments: Initial Conditions of MS: The MS has valid TMSI and is deactivated.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcC, TSPX_MTChRateC)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_1, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_TmsiOff, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			att=1
4	body	+ImsiAttachIni(MiTmsi_01, TimingAdv(30), C_MCC_1, C_PLMN_1, C_SIMneedRmv)			
5		L!DL_DatRqLupAcp CANCEL T_dly	LocAcp(TCV_ch, LocUpdtAcp_02(C_MCC_1, C_PLMN_1, TCV_lac))		1.
6		+NoReaction(5000)			2.
7		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		
8	L_0394	L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))	(P)	
9		+ChanRel_end(TCV_ch)			
10	L_0395	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TI_01))	(P)	
11		+ChanRel_end(TCV_ch)			

Detailed Comments:
1. Follow on proceed IE included.
2. MS shall not send any layer 3 message for 5 seconds after reception of Location Updating Acc.

Test Group CC

Test Case Dynamic Behaviour					
Test Case Name:		TC_26_8_1_2_1_1			
Group:		GSM_L3_MS_v4220/CC/			
Purpose:		To verify that upon initiation of an outgoing basic call by user the MS initiates establishment of an MM connection, using as first MM message a CM-SERVICE REQUEST message with CM service type "Mobile originating call establishment or packet mode connection establishment".			
Default:		OtherEventsFail			
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcC, TSPX_MOChRateC)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5	body	+InitCall(TCV_Service)			
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_19)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(C_AGCH_A_1, ImmAsgn_nfh(TCV_Rr, TCV_Fn, TCV_chtype, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TCV_tch_arfcn, C_normal_paging))		1.
9		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_04)		
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		+PostMainLinkRel(TCV_chTch)			
Detailed Comments:		1. To assign TCH channel.			

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_2_1

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of the MS in CC-state U0.1, "MM-connection pending", upon the MS receiving a CM SERVICE REJECT message, returns to CC state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcA, TSPX_MOChRateA)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+InitCall(TCV_Service)			
5		+PreEnterCCstateU01_21(TSPX_SDCCH4SubDef, TimingAdv(0))			
6	body	L!DL_DatRqCmsRej	CMSerRej(TCV_ch, CMServiceRej_01)		2.
7		+CheckTIsInStateU0(TRUE, TCV_ch)			
8		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service and channel rate, if the MS supports telephony the selected basic service is telephony.
2. To reject the CM-SERVICE REQUEST.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_2_2

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of the MS in CC-state U0.1, "MM-connection pending", upon the MS receiving a CM SERVICE ACCEPT message, sends a SETUP message specifying the Called party BCD number that was entered into the MS and then enters CC state U1, "Call initiated".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcB, TSPX_MOChRateB)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+InitCall(TCV_Service)			
5		+PreEnterCCstateU01_21(TSPX_SDCCH4SubDef, TimingAdv(0))			
6	body	L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
7		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			2.
8		(TCV_CalledNum := TCV_Setup_mo1.cdpn, TCV_Res := OO_CalledPtyNumCHK(TCV_CalledNum))			
9	L_0396	[NOT TCV_Res]		(F)	3.
10		+PostMainLinkRel(TCV_ch)			
11	L_0397	[TCV_Res]		(P)	
12		+CCstatuschk_05(C_U1, TCV_TI,TCV_ch)			4.
13		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service and channel rate, if the MS supports telephony the selected basic service is telephony.
2. To receive the SETUP message with called party BCD number.
3. The called party number contained in the SETUP message is not the one entered.
4. To check whether the MS is in state U1.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_2_3

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that after the MS with a CC entity in state U0.1, "MM-connection pending", has detected a lower layer failure and has returned to idle mode, the CC entities relating to the seven mobile originating transaction identifiers are in state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcC, TSPX_MOChRateC)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+InitCall(TCV_Service)			
5		+PreEnterCCstateU01_21(TSPX_SDCCH4SubDef, TimingAdv(0))			
6	body	(TCV_Null := OM_LowerLayerFail(TCV_ch))			
7		START T_dly(20000)			
8		?TIMEOUT T_dly			
9		+CCEstablishMT_SDCCH4(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
10		+CheckTIsInStateU0(TRUE, TCV_ch)			
11		+PostMainLinkRel(TCV_ch)			

Detailed Comments: 1. To select a basic service and channel rate, if the MS supports telephony the selected basic service is telephony.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_3_1

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of the MS in CC-state U1, "Call initiated", upon receipt of a CALL PROCEEDING message, enters CC state U3, "Mobile originating call proceeding".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcD, TSPX_MOChRateD)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU1_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqCallProc	CallProc(TCV_chTch, TCV_CallProc)		
8		+CCstatuschk_05(C_U3, TCV_TI, TCV_chTch)			2.
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a basic service and channel rate, if the MS supports telephony the selected basic service is telephony.
2. Check that CC state is U3.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_3_2

Group: GSM_L3_MS_v4220/CC/

Purpose:
1) To verify that a CC entity of the MS in CC-state U1, "Call initiated", upon receipt of a RELEASE COMPLETE message with valid cause value, enters CC state U0, "Null".
2) To verify that in returning to idle mode, the CC entities relating to the seven mobile originating transaction identifiers are in state U0, "Null".
3) To verify that in releasing the MM-connection, the MS shall wait for MM layer release initiated by SS.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcE, TSPX_MOChRateE)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU1_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_04(TCV_TI))		
8		+CheckTIsInStateU0(TRUE, TCV_chTch)			
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To select a basic service and channel rate, if the MS supports telephony the selected basic service is telephony.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_3_3

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of the MS in CC-state U1, "Call initiated", upon expiry of T303 (accuracy +/- 20% between reception of CM-SERVICE REQUEST and DISCONNECT by SS) sends a DISCONNECT message to its peer entity and enters state U11, "Disconnect request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcF, TSPX_MOChRateF)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU1_22Timer(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L?DL_DatInDisc (TCV_Fn1 := DL_DatInDisc.fn) READTIMER T_dly(TCV_Time), CANCEL T_dly	DiscRcv(TCV_chTch, DisconnR(TCV_TI0, Cause_Def))	(P)	
8	L_0398	[(TCV_Time < 24000) OR (TCV_Time >= 36000)]		(F)	
9		+PostMainLinkRel(TCV_chTch)			
10	L_0399	[(TCV_Time >= 24000) AND(TCV_Time < 36000)]		(P)	
11		+CCstatuschk_05(C_U11, TCV_TI, TCV_chTch)			
12		+PostMainLinkRel(TCV_chTch)			

Detailed Comments: 1. To select a basic service and channel rate, if the MS supports telephony the selected basic service is telephony.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_3_4

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that after the MS with a CC entity in state U1 "Call initiated", has detected a lower layer failure and has returned to idle mode, the CC entities relating to the seven mobile originating transaction identifiers are in state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcG, TSPX_MOChRateG)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU1_24(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			2.
7	body	(TCV_Null := OM_LowerLayerFail(TCV_chTch))			3.
8		START T_dly(20000)			
9		?TIMEOUT T_dly			4.
10		+CCEstablishMT_SDCCH4(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
11		+CheckTIsInStateU0(TRUE, TCV_ch)			5.
12		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service and channel rate, if the MS supports telephony the selected basic service is telephony.
2. To bring the MS into U1 state.
3. A layer failure generated in the test system.
4. Waiting for the MS return to idle state.
5. To check that CC entities related to all transaction identifiers are return to idle.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_3_5

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of the MS in CC-state U1, "Call initiated", upon receipt of an ALERTING message, enters CC state U4, "Call delivered".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcH, TSPX_MOChRateH)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU1_24(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
8		+CCstatuschk_05(C_U4, TCV_TI, TCV_chTch)			
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments: 1. To select a basic service and channel rate, if the MS supports telephony the selected basic service is telephony.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_3_6

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of the MS in CC-state U1, "Call initiated", upon receipt of a CONNECT message, sends a CONNECT ACKNOWLEDGE message to its peer entity and enters CC state U10, "Active".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcI, TSPX_MOChRateI)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU1_24(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
8		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
9		+CCstatuschk_05(C_U10, TCV_TI, TCV_chTch)			
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments: 1. To select a basic service and channel rate, if the MS supports telephony the selected basic service is telephony.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_3_7

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of the MS in CC-state U1, "Call initiated", upon receipt of a message with message type not defined for the protocol discriminator unknown message from its peer entity returns a STATUS message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcJ, TSPX_MOChRateJ)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+InitCall(TCV_Service)			
5		+PreEnterCCstateU1_21(TSPX_SDCCH4SubDef, TimingAdv(0))			
6	body	L!DL_DatRqUndefCC	UndefCC(TCV_ch, UndefCC_02(TCV_TI))		2.
7	L_0400	L?DL_DatInCcst	CCStRcv(TCV_ch, CCStatus_08(TCV_TI0, C_U1))	(P)	3.
8		+CCstatuschk_05(C_U1, TCV_TI, TCV_ch)			4.
9		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service and channel rate, if the MS supports telephony the selected basic service is telephony.
2. Message type not defined for CC.
3. The expected STATUS message received.
4. To check whether the MS is still in CC state U1.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_4_1

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U3, "Mobile Originating Call Proceeding", upon receipt of a ALERTING message enters CC-state U4, "Call Delivered".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcA, TSPX_MOChRateA)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU3_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
8		+CCstatuschk_05(C_U4, TCV_TI, TCV_chTch)			
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments: 1. To select a circuit switched basic service for testing.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_4_2

Group: GSM_L3_MS_v4220/CC/

Purpose: 1) To verify that a CC-entity of the MS in CC-state U3, "Mobile Originating Call Proceeding", upon receipt of a CONNECT message returns a "CONNECT ACKNOWLEDGE" message to its peer entity and enters the CC state U10, "Active".
2) To verify that the MS stops locally generated indication, if any.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcB, TSPX_MOChRateB)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU3_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
8		(TCV_Res := OO_ToneStopCHK())			
9		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
10	L_0401	[TCV_Res]		(P)	
11		+CCstatuschk_05(C_U10, TCV_TI, TCV_chTch)			
12		+PostMainLinkRel(TCV_chTch)			
13	L_0402	[NOT TCV_Res]		(F)	
14		+PostMainLinkRel(TCV_chTch)			

Detailed Comments: 1. To select a circuit switched basic service for testing.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_4_3
Group: GSM_L3_MS_v4220/CC/
Purpose: 1) To verify that a CC-entity of the MS in CC-state U3, "Mobile Originating Call Proceeding", upon receipt of a PROGRESS message with valid cause values stays in CC-state U3.
2) To verify that after receipt of the PROGRESS message timer T310 is stopped.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcC, TSPX_MOChRateC)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU3_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqProg	Progress(TCV_chTch, Progress_01(TCV_TI))		
8		+CCstatuschk_05(C_U3, TCV_TI, TCV_chTch)			2.
9		START T_dly(45000)			3.
10	L_0403	L?DL_DatInDisc CANCEL T_dly	DiscRcv(TCV_chTch, DisconnR(TCV_TI0, Cause_Def))	(F)	
11		+PostMainLinkRel(TCV_chTch)			
12		?TIMEOUT T_dly			
13		+CCstatuschk_05(C_U3, TCV_TI, TCV_chTch)			4.
14		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To select2 a circuit switched basic service for testing.
2. To check that CC state is U3
3. To check that the MS has stopped T310.
4. To check that CC state is still U3

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_4_4

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U3, "Mobile Originating Call Proceeding", upon receipt of a PROGRESS message indicating in-band announcement through-connects the traffic channel for speech, if TCH is in speech mode. If TCH is not in a speech mode, the MS does not through-connect the TCH.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcD, TSPX_MOChRateD)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU3_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqProg	Progress(TCV_chTch, Progress_02(TCV_TI))		
8		(TCV_Res := OO_TCHThroConnCHK())			
9		[TCV_ChMod.mode = C_ChMod_r]			2.
10	L_0404	[TCV_Res]		(P)	
11		+localtree			
12	L_0405	[NOT TCV_Res]		(F)	
13		+PostMainLinkRel(TCV_chTch)			
14		[TCV_ChMod.mode<> C_ChMod_r]			3.
15	L_0406	[NOT TCV_Res]		(P)	
16		+localtree			
17	L_0407	[TCV_Res]		(F)	
18		+PostMainLinkRel(TCV_chTch)			
		localtree			
19		+CCstatuschk_05(C_U3, TCV_TI, TCV_chTch)			4.
20		START T_dly(45000)			5.
21	L_0408	L?DL_DatInDisc CANCEL T_dly	DiscRcv(TCV_chTch, DisconnR(TCV_TI0, Cause_Def))	(F)	
22		+PostMainLinkRel(TCV_chTch)			
23		?TIMEOUT T_dly			
24		+CCstatuschk_05(C_U3, TCV_TI, TCV_chTch)			6.
25		(TCV_Res := OO_TCHThroConnCHK())			
26		[TCV_ChMod.mode = C_ChMod_r]			7.
27	L_0409	[TCV_Res]		(P)	
28		+PostMainLinkRel(TCV_chTch)			

29	L_0410	[NOT TCV_Res]		(F)	
30		+PostMainLinkRel(TCV_chTch)			
31	L_0411	[TCV_ChMod.mode<> C_ChMod_r]		(P)	
32		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a circuit switched basic service for testing.
2. To check if the TCH is connected through if the channel mode is speech, so that the inband information may be heard.
3. To check if the TCH is not connected through if the channel mode is not speech.
4. To check that CC is in state U3.
5. To check that the MS has stopped T310
6. To check that CC is still in state U3.
7. To check if the TCH is connected through if the channel mode is speech.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_4_5

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U3, "Mobile Originating Call Proceeding", upon receipt of a DISCONNECT with progress indicator #8 through-connects the speech channel to make in-band announcements available, if traffic channel is in speech mode. If TCH is not in speech mode, the MS sends a RELEASE message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcE, TSPX_MOChRateE)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU3_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_03iei, UuInfo_omit))		
8		[TCV_ChMod.mode = C_ChMod_r]			
9		(TCV_Res := OO_TCHThroConnCHK())			
10	L_0412	[TCV_Res]		(P)	2.
11		+CCstatuschk_05(C_U12, TCV_TI, TCV_chTch)			3.
12		+PostMainLinkRel(TCV_chTch)			
13	L_0413	[NOT TCV_Res]		(F)	
14		+PostMainLinkRel(TCV_chTch)			
15		[TCV_ChMod.mode <> C_ChMod_r]			
16	L_0414	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	
17		+CCstatuschk_05(C_U19, TCV_TI, TCV_chTch)			4.
18		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a circuit switched basic service for testing.
2. To check that the TCH is connected through if channel mode is speech.
3. To check that CC is in state U12.
4. To check that CC is in state U19.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_4_6

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U3, "Mobile Originating Call Proceeding", upon receipt of a DISCONNECT without progress indicator returns a RELEASE message and enters the CC-state U19, "Release Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcF, TSPX_MOChRateF)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU3_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
8		L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))		
9		+CCstatuschk_05(C_U19, TCV_TI, TCV_chTch)			2.
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To select a circuit switched basic service for testing.
2. To check that CC is in state U19.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_4_7

Group: GSM_L3_MS_v4220/CC/

Purpose:
1) To verify that a CC-entity of the MS in CC-state U3, "Mobile Originating Call Proceeding", upon receipt of a RELEASE will return a RELEASE COMPLETE and enter the CC-state U0, "Null".
2) To verify that the MS on returning to the idle mode releases the MM-connection and that the CC-entities relating to the seven mobile originating transaction identifiers are in CC-state U0, "Null".
3) To verify that in releasing the MM-connection, the MS shall wait for MM layer release initiated by SS.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcG, TSPX_MOChRateG)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU3_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_03(TCV_TI))		
8	L_0415	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TCV_TI0))	(P)	
9		+CheckTIsInStateU0(TRUE, TCV_chTch)			
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments: 1. To select a circuit switched basic service for testing.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_4_8

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U3, "Mobile Originating Call Proceeding", upon request by the user to terminate will send a DISCONNECT message and enter the CC-state U11, "Disconnect Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcH, TSPX_MOChRateH)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+InitCall(TCV_Service)			
5		+PreEnterCCstateU3_23(TSPX_SDCCH4SubDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			2.
6	body	(TCV_Null := OO_TermCall())			
7	L_0416	L?DL_DatInDisc	DiscRcv(TCV_ch, DisconnR(TCV_TI0, Cause_Def))	(P)	3.
8		+CCstatuschk_05(C_U11, TCV_TI, TCV_ch)			4.
9		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a circuit switched basic service for testing.
2. To initiate MO call and bring the MS into U3 state.
3. The expected DISCONNECT received.
4. To check whether the MS is in CC state U11.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_4_9

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U3, "Mobile Originating Call Proceeding", when allocated a traffic channel by the network performing the assignment procedure, performs a layer 2 establishment on the FACCH without changing the state of the call in progress.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcI, TSPX_MOChRateI)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU3_23(TSPX_SDCCH4SubDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
7	body	+CCAssignTCH(TSPX_TmSltDef, TSPX_TscDef)			2.
8		+CCstatuschk_05(C_U3, TCV_TI, TCV_chTch)			3.
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4.
2. To assign traffic channel.
3. To check that the MS is in state U3.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_4_10

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U3, "Mobile Originating Call Proceeding" will, upon expiry of timer T310, initiate call release by sending DISCONNECT and enter the CC-state U11, "Disconnect Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcJ, TSPX_MOChRateJ)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+InitCall(TCV_Service)			
5		+PreEnterCCstateU3_23(TSPX_SDCCH4SubDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			2.
6	body	START T_dly1(45000)			
7	L_0417	?TIMEOUT T_dly1		(F)	3.
8		+PostMainLinkRel(TCV_ch)			
9		L?DL_DatInDisc	DiscRcv(TCV_ch, DisconnR(TCV_TI0, Cause_Def))		
10		READTIMER T_dly1(TCV_Time), CANCEL T_dly1			
11	L_0418	[TCV_Time < 29400]		(F)	4.
12		+PostMainLinkRel(TCV_ch)			
13	L_0419	[TCV_Time >= 29400]		(P)	
14		+CCstatuschk_05(C_U11, TCV_TI, TCV_ch)			5.
15		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4.
2. To initiate MO call and bring the MS into U3 state.
3. T310 >= 45 seconds, fail.
4. T310 < 29.4 seconds, fail.
5. To check that the MS is in the state U11.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_4_11

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U3, "Mobile Originating Call Proceeding" having detected a lower layer failure and having returned to idle mode, the CC entities relating to the seven mobile originating transaction identifiers are in state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcA, TSPX_MOChRateA)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			2.
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU3_24(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	(TCV_Null := OM_LowerLayerFail(TCV_chTch))			
8		START T_dly(20000)			
9		?TIMEOUT T_dly			3.
10		+CCEstablishMT_SDCCH4(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
11		+CheckTIsInStateU0(TRUE, TCV_ch)			4.
12		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service and channel rate, if the MS supports telephony the selected basic service is telephony.
2. To setup the full rate or half rate traffic channel .
3. Waiting for the MS return to idle state.
4. To check that CC entities related to all transaction identifiers are return to idle.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_4_12

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U3, "Mobile Originating Call Proceeding" having received an unknown message from its peer entity returns a STATUS message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcB, TSPX_MOChRateB)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+InitCall(TCV_Service)			
5		+PreEnterCCstateU3_21(TSPX_SDCCH4SubDef, TimingAdv(0))			
6	body	L!DL_DatRqUndefCC	UndefCC(TCV_ch, UndefCC_02(TCV_TI))		2.
7	L_0420	L?DL_DatInCcst	CCStRcv(TCV_ch, CCStatus_08(TCV_TI0, C_U3))	(P)	3.
8		+CCstatuschk_05(C_U3, TCV_TI, TCV_ch)			4.
9		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4.
2. To send a message which type is not defined for the CC.
3. The expected STATUS message received.
4. To check wether the MS is still in the CC state U3.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_4_13

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that if the user connection is not attached to the radio path, the MS generates internally an alerting indication when the call control entity of the MS in the "mobile originating call proceeding" state receives an ALERTING message then it enters "call delivered" state and, for speech calls, if the user connection is not attached to the radio path, the MS shall internally generate an alerting indication.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+SpeechService			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+InitCall(TCV_Service)			
5		+PreEnterCCstateU3_21(TSPX_SDCCH4SubDef, TimingAdv(0))			
6	body	L!DL_DatRqAlert	AlertSnd(TCV_ch, Alerting_01(TCV_TI))		
7		(TCV_Res := OO_AltIndCHK())			
8	L_0421	[TCV_Res]		(P)	1.
9		+CCstatuschk_05(C_U4, TCV_TI, TCV_ch)			2.
10		+PostMainLinkRel(TCV_ch)			
11	L_0422	[NOT TCV_Res]		(F)	3.
12		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. The Ms generates a alerting indication to the user.
2. To check whether the MS is in CC state U4.
3. The MS does not generate a alerting indication to the user.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_5_1

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U4, "Call Delivered", upon receipt of the CONNECT message returns a CONNECT ACKNOWLEDGE to its peer entity and enters the CC-state U10, "Active".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcC, TSPX_MOChRateC)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU4_23(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
7	body	L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
8	L_0423	L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))	(P)	2.
9		(TCV_Res := OO_AltIndCHK())			
10	L_0424	[TCV_Res]		(F)	
11		+PostMainLinkRel(TCV_chTch)			3.
12	L_0425	[NOT TCV_Res]		(P)	4.
13		+CCstatuschk_05(C_U10, TCV_TI, TCV_chTch)			5.
14		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To set a physical channel as BCCH, CCCH and SDCCH4.
2. The expected CONNECT ACKNOWLEDGE received.
3. Alerting does not stop.
4. Alerting stopped.
5. To check whether the MS is in state U10.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_5_2

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U4, "Call Delivered", upon request by the user to terminate will send a DISCONNECT message and enter the CC-state U11, "Disconnect Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcD, TSPX_MOChRateD)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU4_23(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
7	body	(TCV_Null := OO_TermCall())			
8	L_0426	L?DL_DatInDisc	DiscRcv(TCV_chTch, DisconnR(TCV_TI0, Cause_Def))	(P)	2.
9		+CCstatuschk_05(C_U11, TCV_TI, TCV_chTch)			3.
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4.
2. The expected DISCONNECT message received.
3. To check whether the MS enters into the state U11.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_5_3

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U4, "Call Delivered", upon receipt of a DISCONNECT with a progress indicator indicating in-band information, through-connects the speech channel to make in-band announcements available, if traffic channel is in speech mode. If TCH is not in speech mode, the MS shall send a RELEASE message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcE, TSPX_MOChRateE)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU4_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_03iei, UuInfo_omit))		
8		[TCV_ChMod.mode = C_ChMod_r]			
9		+CCstatuschk_05(C_U12, TCV_TI, TCV_chTch)			
10		+PostMainLinkRel(TCV_chTch)			
11		[TCV_ChMod.mode <> C_ChMod_r]			
12	L_0427	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	
13		+CCstatuschk_05(C_U19, TCV_TI, TCV_chTch)			
14		+PostMainLinkRel(TCV_chTch)			

Detailed Comments: 1. To select a circuit switched basic service for testing.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_5_4

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U4, "Call Delivered", upon receipt of a DISCONNECT without progress indicator, returns a RELEASE message and enters the CC-state U19, "Release Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcF, TSPX_MOChRateF)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU4_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
8	L_0428	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	
9		+CCstatuschk_05(C_U19, TCV_TI, TCV_chTch)			
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments: 1. To select a circuit switched basic service for testing.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_5_5

Group: GSM_L3_MS_v4220/CC/

Purpose:
1) To verify that a CC-entity of the MS in CC-state U4, "Call Delivered", upon receipt of the RELEASE message will respond with the RELEASE COMPLETE message and enter the CC-state U0, "Null"
2) To verify that the MS on returning the idle mode releases the MM-connection and that the CC-entities relating to the seven mobile originating transaction identifiers are in CC-state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcG, TSPX_MOChRateG)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU4_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_03(TCV_TI))		
8	L_0429	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TCV_TI0))	(P)	
9		+CheckTIsInStateU0(TRUE, TCV_chTch)			
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments: 1. To select a circuit switched basic service for testing.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_5_6

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U4, "Call Delivered" having detected a lower layer failure and has returned to idle mode, the CC-entities relating to the seven mobile originating transaction identifiers are in CC-state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcH, TSPX_MOChRateH)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU4_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	(TCV_Null := OM_LowerLayerFail(TCV_chTch))			
8		START T_dly(20000)			
9		?TIMEOUT T_dly			
10		+CCEstablishMT_SDCCH4(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
11		+CheckTIsInStateU0(TRUE, TCV_ch)			
12		+PostMainLinkRel(TCV_ch)			

Detailed Comments: 1. To select a circuit switched basic service for testing.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_5_7

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U4, "Call Delivered", when allocated a traffic channel by the network performing the assignment procedure, shall perform a layer 2 establishment on the FACCH without changing the state of the call in progress.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcI, TSPX_MOChRateI)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU4_21(TSPX_SDCCH4SubDef, TimingAdv(0))			2.
7	body	+CCAssignTCH(TSPX_TmSltDef, TSPX_TscDef)			3.
8		+CCstatuschk_05(C_U4, TCV_TI, TCV_chTch)			4.
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4.
2. To initiate MO call and bring the MS into U4 state.
3. The assignment procedure succeeds.
4. To check that the MS is still in state U4.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_5_8

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U4, "Call Delivered", having received an unknown message from its peer entity returns a STATUS message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcJ, TSPX_MOChRateJ)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			2.
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU4_24(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqUndefCC	UndefCC(TCV_chTch, UndefCC_02(TCV_TI))		3.
8		L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_08(TCV_TI0, C_U4))		
9		+CCstatuschk_05(C_U4, TCV_TI, TCV_chTch)			
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a basic service and channel rate, if the MS supports telephony the selected basic service is telephony.
2. To setup the full rate or half rate traffic channel and BCCH, CCCH for the test.
3. To send a CC message which message type is undefined for the CC.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_6_1

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that the a CC-entity of the MS in CC-state U10, "Call Active", upon request by the user to terminate will send a DISCONNECT message and enter the CC-state U11, "Disconnect Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+InitCall(TCV_Service)			
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+PreEnterCCstateU10_21(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			2.
7	body	(TCV_Null := OO_TermCall())			
8	L_0430	L?DL_DatInDisc	DiscRcv(TCV_chTch, DisconnR(TCV_TI0, Cause_Def))	(P)	3.
9		+CCstatuschk_05(C_U11, TCV_TI, TCV_chTch)			4.
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4.
2. To bring the MS into U10 state.
3. The expected DISCONNECT message received.
4. To check whether the MS enters the state U11.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_6_2

Group: GSM_L3_MS_v4220/CC/

Purpose:
1) To verify that the a CC-entity of the MS in CC-state U10, "Call Active", upon receive of the RELEASE will respond with the RELEASE COMPLETE message and enter the CC-state U0, "Null"
2) To verify that the MS on returning to the idle mode releases the MM-connection and that the CC-entities relating to the seven mobile originating transaction identifiers are in CC-state U0, "Null"

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcB, TSPX_MOChRateB)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+InitCall(TCV_Service)			
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+PreEnterCCstateU10_21(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			2.
7	body	L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_03(TCV_TI))		
8	L_0431	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TCV_TI0))	(P)	3.
9		+CheckTIsInStateU0(TRUE, TCV_chTch)			
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To setup a physical channel as BCCH, CCCH and SDCCH4.
2. To bring the MS into U10 state.
3. The expected RELEASE COMPLETE message received.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_6_3

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U10, "Call Active", upon receipt of a DISCONNECT message with a Progress Indicator indicating in-band information, through-connects the speech channel to make in-band announcements available, if traffic channel is in speech mode. If TCH is not in speech mode, the MS sends a RELEASE message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcC, TSPX_MOChRateC)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			2.
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU10_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_03iei, UuInfo_omit))		3.
8		[TCV_ChMod.mode = C_ChMod_r]			
9		(TCV_Res := OO_TCHThroConnCHK())			4.
10	L_0432	[TCV_Res]		(P)	Inband info audible
11		+CCstatuschk_05(C_U12, TCV_TI, TCV_chTch)			
12		+PostMainLinkRel(TCV_chTch)			
13	L_0433	[NOT TCV_Res]		(F)	No inband info audible
14		+PostMainLinkRel(TCV_chTch)			
15		[TCV_ChMod.mode <> C_ChMod_r]			
16	L_0434	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	
17		+CCstatuschk_05(C_U19, TCV_TI, TCV_chTch)			
18		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a circuit switched basic service for testing.
2. To setup BCCH, CCCH and traffic channels.
3. With progress indicator #8.
4. To check whether the audio path is connected for inband tones.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_6_4

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that the a CC-entity of the MS in CC-state U10, "Call Active", upon receipt of a DISCONNECT message without progress indicator, returns a RELEASE message and enters the CC-state U19, "Release Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcD, TSPX_MOChRateD)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			2.
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU10_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		3.
8	L_0435	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	
9		+CCstatuschk_05(C_U19, TCV_TI, TCV_chTch)			
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a circuit switched basic service for testing.
2. To setup BCCH, CCCH and traffic channels.
3. Without progress indicator #8.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_6_5

Group: GSM_L3_MS_v4220/CC/

Purpose: 1) To verify that a CC entity of the MS in CC-state U10, "Call active" upon receipt of a RELEASE COMPLETE message with valid cause value, enters CC state U0, "Null".

2) To verify that in returning to idle mode, the CC entities relating to the seven mobile originating transaction identifiers are in state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcE, TSPX_MOChRateE)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			2.
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU10_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_04(TCV_TI))		
8		+CheckTIsInStateU0(TRUE, TCV_chTch)			
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments: 1. To select a circuit switched basic service for testing.

2. To setup BCCH, CCCH and traffic channels.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_6_6

Group: GSM_L3_MS_v4220/CC/

Purpose: 1) To verify that a Mobile Station that has a call established and receives a SETUP message answers either with a CALL CONFIRMED message with cause "user busy" if it supports call waiting, or with a RELEASE COMPLETE message with cause "user busy" otherwise.

2) To verify that after having sent this message, the MS is still in state U10 for the established call.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcA, TSPX_MTChRateA)			1.
3		+BasicServiceMOorTelephony(TSPX_MOBscSvcF, TSPX_MOChRateF)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+InitCall(TCV_Service)			
7		+PreEnterCCstateU10_21(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			3.
8	body	(TCV_Setup_mt.sig := Signal_02, TCV_Setup_mt.ti := TCV_TI0)			4.
9		L!DL_DatRqSetup	SetupSnd(TCV_chTch, TCV_Setup_mt)		5.
10	L_0436	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_10(TCV_TI))	(P)	6.
11		+CCstatuschk_05(C_U10, TCV_TI, TCV_chTch)			7.
12		+PostMainLinkRel(TCV_chTch)			
13	L_0437	L?DL_DatInCallCo	CallCfm(CallConfirm_02(TCV_TI))	(P)	8.
14		L?DL_DatInAlert	AlertRcv(AlertingInd_02(TCV_TI))		
15		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI0))		
16		+CCstatuschk_05(C_U10, TCV_TI, TCV_chTch)			7.
17		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. Select a MT service supported by the MS, The SETUP_PDU will be stored in TCV_Setup_mt.
2. To setup a physical channel as BCCH, CCCH and SDCCH4.
3. To bring the MS into the state U10 of a MO call.

4. a) add signal IE with "call waiting tone on" - b) Set TI value same as MO call and TI flag for MT.
5. To establish a second transaction for MT call with the same TI value as that in the MO call
6. Call waiting not supported.
7. To check whether the MS is still in the state U10.
8. Call waiting supported.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_7_1

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that the a CC-entity of the MS in CC-state U11, "Disconnect Request", upon receipt of a DISCONNECT message, returns to its peer entity the RELEASE message and enters the CC-state U19, "Release Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcG, TSPX_MOChRateG)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU11_23(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			2.
7	body	L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
8	L_0438	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	3.
9		+CCstatuschk_05(C_U19, TCV_TI, TCV_chTch)			4.
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4.
2. To bring the MS into the state U11.
3. The expected RELEASE message received.
4. To check whether the MS enters the state U19.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_7_2

Group: GSM_L3_MS_v4220/CC/

Purpose: 1) To verify that the a CC-entity of the MS in CC-state U11, "Disconnect Request", upon receipt of the RELEASE message shall return RELEASE COMPLETE and enter the CC-state U0, "Null".

2) To verify that the MS on returning to the idle mode releases the MM-connection and that the CC-entities relating to the seven mobile originating transaction identifiers are in CC-state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcH, TSPX_MOChRateH)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU11_23(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			2.
7	body	L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_03(TCV_TI))		
8	L_0439	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TCV_TI0))	(P)	3.
9		+CheckTIsInStateU0(TRUE, TCV_chTch)			4.
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4.
2. To bring the MS into the state U11.
3. The expected RELEASE COMPLETE message received.
4. To check that the CC entities with relevant transaction identifiers are in the state U0.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_7_3

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that the a CC-entity of the MS in CC-state U11, "Disconnect Request" shall on expiry of T305, proceeds ahead with the connection release procedure by sending the RELEASE message to its peer entity and enters the CC-state U19, "Release Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcI, TSPX_MOChRateI)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU11_23Timer(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			2.
7	body	(TCV_Cau0.iei := '00001000'B)			3.
8	L_0440	L?DL_DatInRel (TCV_Fn1 := DL_DatInRel.fn) READTIMER T_dly(TCV_Time), CANCEL T_dly	ReleaseRcv(Release_05(TCV_TI0, TCV_Cau0))	(P)	
9	L_0441	[(TCV_Time < 27000) OR (TCV_Time >= 33000)]		(F)	4.
10		+PostMainLinkRel(TCV_chTch)			
11	L_0442	[(TCV_Time >= 27000) AND(TCV_Time < 33000)]		(P)	5.
12		+CCstatuschk_05(C_U19, TCV_TI, TCV_chTch)			6.
13		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4.
2. To bring the MS into the state U11.
3. Cause IE in DISCONNECT without IEI, add IEI to the Cause IE received from DISCONNECT.
4. Fail, if the timeout value of the T305 timer is either greater than or equal to 33 seconds, or less than 27 seconds.
5. Pass, if the timeout value of T305 is OK.
6. To check whether the MS enters the state U19.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_7_4

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that the a CC-entity of the MS in CC-state U11, "Disconnect Request" having detected a lower layer failure returns to the idle mode. The CC entities relating to the seven mobile originating transaction identifiers are thus in state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcJ, TSPX_MOChRateJ)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			2.
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU11_24(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	(TCV_Null := OM_LowerLayerFail(TCV_chTch))			
8		START T_dly(20000)			
9		?TIMEOUT T_dly			
10		+CCEstablishMT_SDCCH4(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
11		+CheckTIsInStateU0(TRUE, TCV_ch)			
12		+PostMainLinkRel(TCV_ch)			

Detailed Comments:
1. To select a circuit switched basic service for testing.
2. To setup BCCH, CCCH and traffic channels.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_7_5

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that the a CC-entity of the MS in CC-state U11, "Call Delivered" having received an unknown message from its peer entity shall return a STATUS message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcA, TSPX_MOChRateA)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			2.
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU11_24(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqUndefCC	UndefCC(TCV_chTch, UndefCC_02(TCV_TI))		3.
8	L_0443	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_08(TCV_TI0, C_U11))	(P)	
9		+CCstatuschk_05(C_U11, TCV_TI, TCV_chTch)			
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a circuit switched basic service for testing.
2. To setup BCCH, CCCH and traffic channels.
3. To send a CC message which message type is undefined for the CC.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_8_1

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U12, "Disconnect Indication" being in network initiated call release phase, shall, upon receiving a call release request from the user sends a RELEASE to its peer entity and enters CC-state U19, "Release Request"

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+SpeechService			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_ChMod_r, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU12_21(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			2.
7	body	(TCV_Null := OO_TermCall())			3.
8		L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))		4.
9		+CCstatuschk_05(C_U19, TCV_TI, TCV_chTch)			5.
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCHH4, and wait for the MS in service.
2. To bring the MS into the state U12.
3. MMI action, "on hook".
4. The expected RELEASE message received.
5. To check whether the MS is in the state U19.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_8_2

Group: GSM_L3_MS_v4220/CC/

Purpose:
1) To verify that a CC-entity of the MS in CC-state U12, "Disconnect Indication", upon receipt of a RELEASE message returns to its peer entity the RELEASE COMPLETE message and enters the CC-state U0, "Null"
2) To verify that the MS on returning to the idle mode releases the MM-connection and that the CC-entities relating to the seven mobile originating transaction identifiers are in CC-state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+SpeechService			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_ChMod_r, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU12_21(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			2.
7	body	L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_03(TCV_TI))		
8	L_0444	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TCV_TI0))	(P)	3.
9		+CheckTIsInStateU0(TRUE, TCV_chTch)			4.
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To setup a physical channel as BCCH, CCCH and SDCHH4, and wait for the MS in service.
2. To bring the MS into the state U12. Within the step a physical channel as appropriate traffic channel is setup.
3. The expected RELEASE COMPLETE message received.
4. To check whether the CC entities releated to the seven mobile originating transaction is in the state U0.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_8_3

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U12, "Disconnect Indication" having detected a lower layer failure returns to idle mode. The CC-entities relating to the seven mobile originating transaction identifiers are thus in state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+SpeechService			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU12_22(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	(TCV_Null := OM_LowerLayerFail(TCV_chTch))			
8		START T_dly(20000)			
9		?TIMEOUT T_dly			
10		+CCEstablishMT_SDCCH4(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
11		+CheckTIsInStateU0(TRUE, TCV_ch)			
12		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_8_4
Group: GSM_L3_MS_v4220/CC/
Purpose: To verify that a CC-entity of the MS in CC-state U12, "Disconnect Indication" having received an unknown message from its peer entity returns a STATUS message.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+SpeechService			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_ChMod_r, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU12_23(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			2.
7	body	L!DL_DatRqUndefCC	UndefCC(TCV_chTch, UndefCC_02(TCV_TI))		3.
8		L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_08(TCV_TI0, C_U12))		4.
9		+CCstatuschk_05(C_U12, TCV_TI, TCV_chTch)			5.
10		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To setup a physical channel as BCCH, CCCH and SDCCH4, waiting for the MS in service.
2. To bring the MS into the state U12.
3. To send a CC message which message type is undefined for the CC.
4. The expected STATUS message with cause #97 received.
5. To check whether the MS is still in the state U12.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_9_1

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U19, "Release Request" will, upon the first expiry of timer T308 send the RELEASE message to its peer entity and remain in the CC-state U19.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcB, TSPX_MOChRateB)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU19_24Timer(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L?DL_DatInRel (TCV_Fn1 := DL_DatInRel.fn) READTIMER T_dly(TCV_Time), CANCEL T_dly	ReleaseRcv(Release_10(TCV_TI0))		Any cause value
8	L_0445	[(TCV_Time < 27000) OR (TCV_Time >= 33000)]		(F)	1.
9		+PostMainLinkRel(TCV_chTch)			
10	L_0446	[(TCV_Time >= 27000) AND(TCV_Time < 33000)]		(P)	2.
11		+CCstatuschk_05(C_U19, TCV_TI, TCV_chTch)			
12		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. Fail, if the T308 timer value is either greater than or equal to 33 seconds, or less than 27 seconds.
2. Pass, if the T308 timer value is OK.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_9_2

Group: GSM_L3_MS_v4220/CC/

Purpose:
1) To verify that a CC-entity of the MS in CC-state U19, "Release Request", upon the 2nd expiry of the timer T308, enters the CC-state U0, "Null".
2) To verify that subsequently the MS proceeds with releasing the MM-connection and enters the idle mode with the CC entities relating to the seven mobile originating transaction identifiers in state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcC, TSPX_MOChRateC)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			2.
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU19_24(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))		3.
8		+CCstatuschk_05(C_U19, TCV_TI, TCV_chTch)			
9		START T_dly(50000)			
10	L_0447	?TIMEOUT T_dly		(F)	4.
11	L_0448	L?DL_RelIn START T_dly(10000)	DLRelInd_01	(P)	
12		?TIMEOUT T_dly			
13		+CCEstablishMT_SDCCH4(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
14		+CheckTIsInStateU0(TRUE, TCV_ch)			
15		+PostMainLinkRel(TCV_ch)			

Detailed Comments:
1. To select a circuit switched basic service for testing.
2. To setup BCCH, CCCH and traffic channels.
3. First timeout of T308, release with any cause value accepted.
4. Second timeout of T308 (30 seconds), timeout of T3240 (10 seconds) and 10 seconds for the MS to return to listening to paging.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_9_3

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U19, "Release Request", upon receipt of a RELEASE, shall release the MM-connection and enters the CC-state U0, "Null" with the CC entities relating to the seven mobile originating transaction identifiers in state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcD, TSPX_MOChRateD)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU19_24(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			
7	body	L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_06(TCV_TI))		
8		+CheckTIsInStateU0(TRUE, TCV_chTch)			
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_9_4

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U19, "Release Request", upon receipt of a RELEASE COMPLETE, shall release the MM-connection and enters the CC-state U0, "Null" with the CC entities relating to the seven mobile originating transaction identifiers in state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcE, TSPX_MOChRateE)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU19_21(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			2.
7	body	L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		3.
8		+CheckTIsInStateU0(TRUE, TCV_chTch)			4.
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4, waiting the MS in service.
2. To bring the MS into the state U19.
3. To send a RELEASE COMPLETE message to the MS.
4. To check whether the CC entities related to the seven mobile originating transaction identifiers are in the state U0.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_2_9_5

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U19, "Release Request", having detected a lower layer failure, returns to the idle mode, the CC entities relating to the seven mobile originating transaction identifiers are in state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMOorTelephony(TSPX_MOBscSvcF, TSPX_MOChRateF)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+PreEnterCCstateU19_21(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0))			2.
7	body	(TCV_Null := OM_LowerLayerFail(TCV_chTch))			3.
8		START T_dly(20000)			
9		?TIMEOUT T_dly			4.
10		+CCEstablishMT_SDCCH4(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
11		+CheckTIsInStateU0(TRUE, TCV_ch)			5.
12		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4, waiting the MS in service.
2. To bring the MS into the state U19.
3. To generate lower layer failure in the lower emulator.
4. To wait for the MS back to listening to paging.
5. To check whether the CC entities related to the seven mobile originating transaction identifiers are in the state U0.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_1_1

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of the MS, upon receipt of SETUP containing one bearer capability and this bearer capability is not supported, returns a RELEASE COMPLETE with correct cause value to its peer entity and return to the idle mode. To verify that the CC-entities relating to the seven mobile terminating transaction identifiers are then in the state U0, "NULL".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_SpecialCase := C_Yes)			
3		+BasicServiceMT(TSPX_MT_NonSptSvc, C_Full)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
5	body	+CCEstablishMT_SDCCH4(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			2.
6		+Authentication(TCV_ch, TSPX_CKSNDef, TSPX_RANDDef)			
7		+Ciphering_on(TCV_ch)			
8		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		3.
9	L_0449	L?DL_DatInRelCmp	RelComRcv(ReleaseCmpRcv(TI_01, Cause_12))	(P)	4.
10		+CheckTIsInStateU0(FALSE, TCV_ch)			5.
11		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4.
2. To assign SDCCH4 channel.
3. To send a SETUP message containing a bearer capability not supported by the MS.
4. The expected RELEASE COMPLETE message with cause #88 received.
5. To check that the CC entity is in state U0 with all the relevant transaction identifiers.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_2_1

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of the MS in CC-state U6, "Call Present", shall upon receipt of a rejection indication of the incoming call from the user, shall send RELEASE COMPLETE with the appropriate cause value to its peer entity and enter the CC-state U0, "Null". The CC entities relating to the seven mobile terminating transaction identifiers are then in state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTBscSvcB, TSPX_MTChRateB)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		(TCV_Null := OO_SetRefuseCall())			
5		+PreEnterCCstateU6_32(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
6	body	L?DL_DatInRelCmp	RelComRcv(ReleaseCmpRcv(TI_01, Cause_13))	(P)	3.
7		+CheckTIsInStateU0(FALSE, TCV_ch)			
8		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony, otherwise the selected basic service is indicated by TSPX_MTBscSvcA.
2. One physical channel as BCCH, CCCH and SDCCH4.
3. The expected RELEASE COMPLETE message with cause #21 received.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_3_1

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity in CC-state U9, "MS Terminating Call Confirmed", (if signalled by the network in previous SETUP message that it may alert) will either send a ALERTING message to its peer entity and enter state U7, or send a CONNECT message to its peer entity and enter U8

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTBscSvcB, TSPX_MTChRateB)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+PreEnterCCstateU9_32(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
5	body	L?DL_DatInConn	ConnRcv(Connect_01(TI_01))	(P)	3.
6		+CCstatuschk_05(C_U8, TI_02, TCV_ch)			
7		+PostMainLinkRel(TCV_ch)			
8	L_0450	L?DL_DatInAlert	AlertRcv(AlertingInd_02(TI_01))	(P)	4.
9		+CCstatuschk_05(C_U7, TI_02, TCV_ch)			
10		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service.
2. One physical channel as BCCH, CCCH and SDCCH4.
3. The expected CONNECT message received in case of the MS supporting immediate connect.
4. The expected ALERTING message received in case of the MS not supporting immediate connect.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_3_2

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that A CC-entity of the MS in CC-state U9, "MS Terminating Call Confirmed", when allocated a traffic channel by the network performing the assignment procedure, performs a layer 2 establishment on the FACCH, sends a ALERTING message and enters state U7.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcA, TSPX_MTNIC_ChRateA)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU9_34(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			2.
6	body	+CCAssignTCH(TSPX_TmSltDef, TSPX_TscDef)			3.
7	L_0451	L?DL_DatInAlert	AlertRcv(AlertingInd_02(TI_01))	(P)	4.
8		+CCstatuschk_05(C_U7, TI_02, TCV_chTch)			
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony, otherwise the selected basic service is indicated by TSPX_MTNIC_BscSvcA and the channel rate is specified by TSPX_MTNIC_ChRateA.
2. Use SETUP message without signal IE.
3. Assign an appropriate traffic channel.
4. The expected ALERTING message received.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_3_3

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U9, "MS Terminating Call Confirmed", upon request by the user to terminate will send a DISCONNECT message and enter the CC-state U11, "Disconnect Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcB, TSPX_MTNIC_ChRateB)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+PreEnterCCstateU9_34(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			3.
5	body	(TCV_Null := OO_TermCall())			4.
6	L_0452	L?DL_DatInDisc	DiscRcv(TCV_ch, DisconnR(TI_01, Cause_Def))	(P)	5.
7		+CCstatuschk_05(C_U11, TI_02, TCV_ch)			
8		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. One physical channel as BCCH, CCCH and SDCCH4.
3. Bring MS to state U9.
4. To terminate the call.
5. The expected DISCONNECT received.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_3_4

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U9, "MS Terminating Call Confirmed", upon receipt of a DISCONNECT returns a RELEASE message and enters the CC-state U19, "Release Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcC, TSPX_MTNIC_ChRateC)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+PreEnterCCstateU9_34(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			3.
5	body	L!DL_DatRqDisc	DiscSnd(TCV_ch, DisconnS(TI_02, Cause_01, ProgInd_omit, UuInfo_omit))		4.
6	L_0453	L?DL_DatInRel	ReleaseRcv(Release_10(TI_01))	(P)	5.
7		+CCstatuschk_05(C_U19, TI_02, TCV_ch)			
8		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. One physical channel as BCCH, CCCH and SDCCH4.
3. Bring MS to state U9.
4. To send DISCONNECT message.
5. The expected RELEASE message received.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_3_5

Group: GSM_L3_MS_v4220/CC/

Purpose: 1) To verify that a CC-entity of the MS in CC-state U9, "MS Terminating Call Confirmed", upon receipt of a RELEASE will return a RELEASE COMPLETE and enter the CC-state U0, "Null".

2) To verify that the MS on returning to the idle mode releases the MM-connection and that the CC-entities relating to the seven mobile terminating transaction identifiers are in CC-state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcD, TSPX_MTNIC_ChRateD)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+PreEnterCCstateU9_34(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			3.
5	body	L!DL_DatRqRel	ReleaseSnd(TCV_ch, Release_03(TI_02))		4.
6	L_0454	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TI_01))	(P)	5.
7		+CheckTIsInStateU0(FALSE, TCV_ch)			
8		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. One physical channel as BCCH, CCCH and SDCCH4.
3. Bring MS to state U9.
4. To send RELEASE message with cause = "normal, unspecified".
5. The expected RELEASE COMPLETE received.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_3_6

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of the MS in CC-state U9, "MS Terminating Call Confirmed", having detected a lower layer failure returns to idle mode with the CC entities relating to the seven mobile terminating transaction identifiers in CC-state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcE, TSPX_MTNIC_ChRateE)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+PreEnterCCstateU9_34(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			3.
5	body	(TCV_Null := OM_LowerLayerFail(TCV_ch))			4.
6		START T_dly(20000)			5.
7		?TIMEOUT T_dly			
8		+CCEstablishMT_SDCCH4(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
9		+CheckTIsInStateU0(FALSE, TCV_ch)			
10		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. One physical channel as BCCH, CCCH and SDCCH4.
3. Bring MS to state U9.
4. To generate lower layer failure.
5. To wait 20 s for the MS to return to listening to paging.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_3_7

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC-entity of the MS in CC-state U9, "MS Terminating Call Confirmed" having received an unknown message from its peer entity returns a STATUS message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcF, TSPX_MTNIC_ChRateF)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+PreEnterCCstateU9_34(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			3.
5	body	L!DL_DatRqUndefCC	UndefCC(TCV_ch, UndefCC_02(TI_02))		4.
6	L_0455	L?DL_DatInCcst	CCStRcv(TCV_ch, CCStatus_08(TI_01, C_U9))	(P)	
7		+CCstatuschk_05(C_U9, TI_02, TCV_ch)			
8		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. One physical channel as BCCH, CCCH and SDCCH4.
3. Bring MS to state U9.
4. To send an unknown message.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_4_1

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U7, "Call Received", upon a user accepting the incoming call, shall send a CONNECT message to its peer entity and enter the CC-state U8, "Connect Request"

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcG, TSPX_MTNIC_ChRateG)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU7_33(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
6		(TCV_Null := OO_HookOff())			
7	body	L?DL_DatInConn	ConnRcv(Connect_01(TI_01))	(P)	2.
8		+CCstatuschk_05(C_U8, TI_02, TCV_chTch)			3.
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. The expected CONNECT message received.
3. To check whether the MS is in the state U8. If not so the test case fails in the test step.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_4_2

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U7, "Call Received", upon request by the user to terminate will send a DISCONNECT message and enter the CC-state U11, "Disconnect Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcH, TSPX_MTNIC_ChRateH)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU7_33(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
6	body	(TCV_Null := OO_TermCall())			
7	L_0456	L?DL_DatInDisc	DiscRcv(TCV_chTch, DisconnR(TI_01, Cause_Def))	(P)	2.
8		+CCstatuschk_05(C_U11, TI_02, TCV_chTch)			3.
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. The expected DISCONNECT message received.
3. To check whether the MS is in the state U11. If not so the test case fails in the test step.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_4_3

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U7, "Call Received", upon receipt of a DISCONNECT with a progress indicator indicating in-band information from network, if a TCH was not assigned, returns a RELEASE message and enters the CC-state U19, "Release Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcI,TSPX_MTNIC_ChRateI)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+PreEnterCCstateU7_31(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef)			
5	body	L!DL_DatRqDisc	DiscSnd(TCV_ch, DisconnS(TI_02, Cause_01, ProgInd_03iei, UuInfo_omit))		3.
6	L_0457	L?DL_DatInRel	ReleaseRcv(Release_10(TI_01))	(P)	4.
7		+CCstatuschk_05(C_U19, TI_02, TCV_ch)			5.
8		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. To setup a physical channels as BCCH, CCCH and SDCCH4.
3. To send DISCONNECT message with progress indicator indicating cause #8.
4. The expected RELEASE message received.
5. To check whether the MS is in the state U19.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_4_4

Group: GSM_L3_MS_v4220/CC/

Purpose:
1) To verify that a CC entity of a MS in CC-state U7, "Call Received", upon receipt of a RELEASE will return a RELEASE COMPLETE and enter the CC-state U0, "Null".
2) To verify that the MS on returning to the idle mode releases the MM-connection and that the CC-entities relating to the seven mobile terminating transaction identifiers are in CC-state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcJ, TSPX_MTNIC_ChRateJ)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+PreEnterCCstateU7_31(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef)			
5	body	L!DL_DatRqRel	ReleaseSnd(TCV_ch, Release_03(TI_02))		3.
6		L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TI_01))		4.
7		+CheckTIsInStateU0(FALSE, TCV_ch)			5.
8		+PostMainLinkRel(TCV_ch)			

Detailed Comments:
1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. To setup a physical channels as BCCH, CCCH and SDCCH4.
3. To send RELEASE message with cause "Normal, unspecified".
4. The expected RELEASE COMPLETE message received.
5. To check that the CC entity has returned to state U0 with all transaction identifiers.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_4_5

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U7, "Call Received", having detected a lower layer failure returns to idle mode with the CC entities relating to the seven mobile terminating transaction identifiers in CC-state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcA, TSPX_MTNIC_ChRateA)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+PreEnterCCstateU7_32(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
5	body	(TCV_Null := OM_LowerLayerFail(TCV_ch))			
6		START T_dly(20000)			
7		?TIMEOUT T_dly			
8		+CCEstablishMT_SDCCH4(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
9		+CheckTIsInStateU0(FALSE, TCV_ch)			
10		+PostMainLinkRel(TCV_ch)			

Detailed Comments:
1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. To setup a physical channel as BCCH, CCCH and SDCCH4.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_4_6

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U7, "Call Received", having received an unknown message from its peer entity returns a STATUS message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcB, TSPX_MTNIC_ChRateB)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU7_33(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
6	body	L!DL_DatRqUndefCC	UndefCC(TCV_chTch, UndefCC_02(TI_02))		2.
7		L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_08(TI_01, C_U7))		3.
8		+CCstatuschk_05(C_U7, TI_02, TCV_chTch)			4.
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. To send an unknown mesage to the MS.
3. The expeceted STATUS message with cause #97 and state U7 received.
4. To chweck that the MS remains in the state U7.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_4_7

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U7, "Call Received", when allocated a traffic channel by the network performing the assignment procedure, shall perform a layer 2 establishment on the FACCH without changing the state of the call in progress.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcC, TSPX_MTNIC_ChRateC)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU7_31(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef)			
6	body	+CCAssignTCH(TSPX_TmSltDef, TSPX_TscDef)			2.
7		+CCstatuschk_05(C_U7, TI_02, TCV_chTch)			3.
8		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. The assignment procedure succeeded.
3. To check whether the MS remains in the state U7.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_4_8

Group: GSM_L3_MS_v4220/CC/

Purpose: 1) To verify that a CC entity of the MS in CC-state U7, "Call received", upon receipt of a RELEASE COMPLETE message with valid cause value, enters CC state U0, "Null".

2) To verify that in returning to idle mode, the CC entities relating to the seven mobile originating transaction identifiers are in state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTNIC_BscSvcD, TSPX_MTNIC_ChRateD)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+PreEnterCCstateU7_31(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef)			
5	body	L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_04(TI_02))		
6		+CheckTIsInStateU0(FALSE, TCV_ch)			3.
7		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. To setup a physical channels as BCCH, CCCH and SDCCH4.
3. To check that the CC entities relating to the seven MT transaction identifiers are in state U0.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_5_1
Group: GSM_L3_MS_v4220/CC/
Purpose: To verify that a CC entity of a MS in CC-state U8, "Connect Request", upon receipt of CONNECT ACKNOWLEDGE shall enter the CC-state U10, "Call Active".
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTBscSvcC, TSPX_MTChRateC)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU8_32(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
6	body	L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
7		+CCstatuschk_05(C_U10, TI_02, TCV_chTch)			2.
8		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. To check whether the MS is in U10 state. if it is not in U10 fail in the test step.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_5_2

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U8, "Connect Request", having waited for a reasonable length of time (eg. expiry of timer T313) without receiving the appropriate protocol message to complete the incoming call, shall initiate the clearing of that incoming call by sending the CC message DISCONNECT and enter the CC-state U11, "Disconnect Request"

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTBscSvcD, TSPX_MTChRateD)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU8_32(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
6	body	+localtree1			
		localtree1			
7		START T_dly(15000)			
8		?TIMEOUT T_dly			
9		START T_dly(18000)			
10	L_0458	?TIMEOUT T_dly		(F)	
11		+PostMainLinkRel(TCV_chTch)			
12	L_0459	L?DL_DatInDisc CANCEL T_dly	DiscRcv(TCV_chTch, DisconnR(TI_01, Cause_Def))	(P)	2.
13		+CCstatuschk_05(C_U11, TI_02, TCV_chTch)			3.
14		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. The expected DISCONNECT message received within the time interval with any valid cause location and cause value.
3. To check whether the MS is in the expected U11 state.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_5_3

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U8, "Connect Request", upon request by the user to terminate will send a DISCONNECT message and enter the CC-state U11, "Disconnect Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTBscSvcE, TSPX_MTChRateE)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU8_32(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
6	body	(TCV_Null := OO_TermCall())			2.
7	L_0460	L?DL_DatInDisc	DiscRcv(TCV_chTch, DisconnR(TI_01, Cause_Def))	(P)	3.
8		+CCstatuschk_05(C_U11, TI_02, TCV_chTch)			4.
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. The user terminates the call.
3. The expected DISCONNECT message received.
4. To check whether the MS is in the expected U11 state.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_5_4

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U8, "Connect Request", upon receipt of a DISCONNECT with progress indicator #8 enters CC-state U12, if the traffic channel is in speech mode, and that the MS sends a RELEASE message and enters CC-state U19 if the TCH is not in speech mode.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTBscSvcF, TSPX_MTChRateF)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU8_33(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
6	body	L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TI_02, Cause_01, ProgInd_03iei, UuInfo_omit))		
7		[TCV_ChMod.mode = C_ChMod_r]			2.
8		+CCstatuschk_05(C_U12, TI_02, TCV_chTch)			
9		+PostMainLinkRel(TCV_chTch)			
10		[TCV_ChMod.mode <> C_ChMod_r]			3.
11	L_0461	L?DL_DatInRel	ReleaseRcv(Release_10(TI_01))	(P)	
12		+CCstatuschk_05(C_U19, TI_02, TCV_chTch)			
13		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. TCH is in speech mode.
3. TCH is not in speech mode.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_5_5

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U8, "Connect Request", upon receipt of a DISCONNECT without progress indicator, returns a RELEASE message and enters the CC-state U19, "Release Request".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTBscSvcG, TSPX_MTChRateG)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU8_33(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
6	body	L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TI_02, Cause_01, ProgInd_omit, UuInfo_omit))		
7	L_0462	L?DL_DatInRel	ReleaseRcv(Release_10(TI_01))	(P)	2.
8		+CCstatuschk_05(C_U19, TI_02, TCV_chTch)			
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. The expected RELEASE message received.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_5_6

Group: GSM_L3_MS_v4220/CC/

Purpose: 1) To verify that a CC entity of a MS in CC-state U8, "Connect Request", upon receipt of a RELEASE will return a RELEASE COMPLETE and enter the CC-state U0, "Null".

2) To verify that the MS on returning to the idle mode releases the MM-connection and that the CC-entities relating to the seven mobile terminating transaction identifiers are in CC-state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTBscSvcH, TSPX_MTChRateH)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Immass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU8_33(TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
6	body	L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_03(TI_02))		
7	L_0463	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TI_01))	(P)	2.
8		+CheckTIsInStateU0(FALSE, TCV_chTch)			
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. The expected RELEASE COMPLETE message received.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_5_7

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U8, "Connect Request", having detected a lower layer failure returns to idle mode with the CC entities relating to the seven mobile terminating transaction identifiers in CC-state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTBscSvcI, TSPX_MTChRateI)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+PreEnterCCstateU8_31(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef)			
5	body	(TCV_Null := OM_LowerLayerFail(TCV_ch))			3.
6		START T_dly(20000)			
7		?TIMEOUT T_dly			
8		+CCEstablishMT_SDCCH4(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
9		+CheckTIsInStateU0(FALSE, TCV_ch)			
10		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected teleservice is telephony.
2. To setup a physical channel as BCCH, CCCH and SDCCH4.
3. To generate lower layer failure.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_5_8

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U8, "Connect Request", when allocated a traffic channel by the network performing the assignment procedure, shall perform a layer 2 establishment on the FACCH without changing the state of the call in progress.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTBscSvcJ, TSPX_MTChRateJ)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU8_31(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef)			
6	body	+CCAssignTCH(TSPX_TmSltDef, TSPX_TscDef)			2.
7		+CCstatuschk_05(C_U8, TI_02, TCV_chTch)			3.
8		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. To assign a suitable traffic channel.
3.. To check whether the MS is still in the state U8.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_3_5_9

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U8, "Connect Request", having received an unknown message from its peer entity returns a STATUS message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMTorTelephony(TSPX_MTBscSvcG, TSPX_MTChRateG)			1.
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			2.
4		+PreEnterCCstateU8_31(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef)			
5	body	L!DL_DatRqUndefCC	UndefCC(TCV_ch, UndefCC_02(TI_02))		3.
6	L_0464	L?DL_DatInCcst	CCStRcv(TCV_ch, CCStatus_08(TI_01, C_U8))	(P)	4.
7		+CCstatuschk_05(C_U8, TI_02, TCV_ch)			5.
8		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To select a basic service. if the MS supports telephony the selected basic service is telephony.
2. To setup a physical channel as BCCH, CCCH and SDCCH4.
3. To send an unknown mesage to the MS.
4. The expected STATUS message with cause #97 and state U8 received.
5. To check whether the MS is still in the state U8.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_1_1

Group: GSM_L3_MS_v4220/CC/

Purpose:
1) To verify that an MS supporting the Mobile originating DTMF protocol control procedure, having a CC entity for speech in state U10, "Active": when made to send a DTMF tone, sends a START DTMF message on the correct DCCH.
2) To verify that an MS supporting the Mobile originating DTMF protocol control procedure, having a CC entity for speech in state U10, "Active": when made to send a DTMF tone (the corresponding IA5 character being selected from among the ones supported), sends a START DTMF message specifying the correct IA5 character in the "keypad information" field of the keypad facility information element.

Default: OtherEventsFail

Comments: The test is carried on full rate speech.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcE, TSPX_MTChRateE)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			2.
6	body	[TSPC_BasCharSet]			3.
7		+localtree1("0")			
8		+localtree1("1")			
9		+localtree1("2")			
10		+localtree1("3")			
11		+localtree1("4")			
12		+localtree1("5")			
13		+localtree1("6")			
14		+localtree1("7")			
15		+localtree1("9")			
16		+continue			
17	L_0465	[NOT TSPC_BasCharSet]		I	
		continue			
18		+localtree1("#")			
19		+localtree1("*")			
20		[TSPC_AddCharSet]			4.
21		+localtree1("A")			
22		+localtree1("B")			
23		+localtree1("C")			
24		+localtree1("D")			
25		+localtree			
26		[NOT TSPC_AddCharSet]			
27		+localtree			5.

		localtree			
28		(TCV_Null := OO_ShortKeyDepr("0"))			
29		L?DL_DatInStartDtmf	StartDTMFRcv(StartDtmf_01(TI_01, "0"))		
30		L!DL_DatRqStartDtmfRej	StartDTMFRejSnd(TCV_chTch, StartDtmfRej_01(TI_02))		
31		+CCstatuschk_05(C_U10, TI_02, TCV_chTch)			6.
32		+PostMainLinkRel(TCV_chTch)			
		localtree1(character:IA5String)			
33		(TCV_Null := OO_ShortKeyDepr(character), TCV_Char := character)			
34		L?DL_DatInStartDtmf	StartDTMFRcv(StartDtmf_01(TI_01, TCV_Char))		
35		L!DL_DatRqStartDtmfAck	StartDTMFAckSnd(TCV_chTch, StartDtmfAck_01(TI_02, TCV_Char))		
36		+CCstatuschk_05(C_U10, TI_02, TCV_chTch)			6.
37		[TSPX_DTMFInd]			
38		(TCV_Res := OO_DTMFIndCHK(character))			
39	L_0466	[NOT TCV_Res]		(F)	7.
40		+PostMainLinkRel(TCV_chTch)			
41	L_0467	[TCV_Res]		(P)	8.
42		+localtree2			
43		[NOT TSPX_DTMFInd]			
44		+localtree2			
		localtree2			
45		L?DL_DatInStopDtmf	StopDTMFRcv(StopDtmf_01(TI_01))		
46		L!DL_DatRqStopDtmfAck	StopDTMFAckSnd(TCV_chTch, StopDtmfAck_01(TI_02))		
47		+CCstatuschk_05(C_U10, TI_02, TCV_chTch)			6.

Detailed Comments:

1. To setup 2 phsical channels one as BCCH, CCCH and SDCCH4, another as full rate traffic channel.
2. To bring the MS into the state U10 for speech.
3. To check the character set of "0-9, #, *".
4. To check the character set of "A, B, C, D" if it is supported by the MS.
5. To test the DTMF tone being rejected.
6. To verify that the MS is still in the state U10.
7. The DTMF indication is not correct.
8. The DTMF indication is correct.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_2_1

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U10, "active", upon receiving of a NOTIFY message remains in the active state.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcA, TSPX_MTChRateA)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			2.
6	body	L!DL_DatRqNotify	NotifySnd(TCV_chTch, NotifiReq_01(TI_02))		3.
7		+CCstatuschk_05(C_U10, TI_02, TCV_chTch)			4.
8		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup 2 phsical channels one as BCCH, CCCH and SDCCH4, another as full rate traffic channel.
2. To bring the MS into the state U10 for speech by generic call setup procedure.
3. To send the NOTIFY message to the MS.
4. To verify whether the MS is still in the state U10, the verdict is assigned in the test step.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_3_1
Group: GSM_L3_MS_v4220/CC/
Purpose: To verify that the MS being in the call active state after having successful completed a channel assignment or having completed a handover command remains in the call active state.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcC, TSPX_MTChRateC)			
3		(TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
5		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
6		[TCV_ChRate = C_Full]			
7		+FullRateCh_A_2(C_Synho, TSPX_TmSltC, TSPX_TscDef, TCV_ChMod, FreqTCH(C_arfcn_tchA), FreqTCH(C_arfcn_tchAd), TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			2.
8		+localtree			
9		[TCV_ChRate = C_Half]			
10		+HalfRateCh_A_2(TSPX_TCHHSubDef, C_Synho, TSPX_TmSltC, TSPX_TscDef, TCV_ChMod, FreqTCH(C_arfcn_tchA), FreqTCH(C_arfcn_tchAd), TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			3.
11		+localtree			
		localtree			
12		(TCV_Tchtype := TCV_chtype)			
13		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			4.
14		(TCV_ch := TCV_chTch)			
15		+AssCmdGen(C_CellA, TCV_ChRate, TSPX_TmSltC, TSPX_TscDef, TSPX_TCHHSubA)			
16	body	+Adjust_gsmanddcs_powerlvl(0, 3,TCV_AssCmd)			
17		+AssCh_complete(TCV_ch, TCV_chTch1, TCV_AssCmd)			5.
18		+CCstatuschk_05(C_U10, TI_02, TCV_chTch1)			7.
19		(TCV_Null:=OM_FreeResource(TCV_chTch))			
20		+CCConfigTCH(TSPX_TCHHSubA, C_Synho, TSPX_TmSltA, TSPX_TscDef, TimingAdv(0),			

			C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
21			L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch1, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltA, TSPX_TscDef, C_NCC, C_BCC, TCV_chdescr_arfcn, TCV_tch_arfcn, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
22			L?DL_RacInHoacc	HndOvAccRcv(TCV_chTch, HandOverAcc_02(TSPX_HoRefA))		
23			L?DL_RacInHoacc	HndOvAccRcv(TCV_chTch, HandOverAcc_02(TSPX_HoRefA))		
24			L?DL_RacInHoacc	HndOvAccRcv(TCV_chTch, HandOverAcc_02(TSPX_HoRefA))		
25			L?DL_RacInHoacc	HndOvAccRcv(TCV_chTch, HandOverAcc_02(TSPX_HoRefA))		
26			L?DL_EstIn	DLEstInd(TCV_chTch)		
27	L_0468		L?DL_DatInHoCom	HndOvCmpRcv(TCV_chTch, HandOverCmp_01)	(P)	6.
28			+CCstatuschk_05(C_U10, TI_02, TCV_chTch)			7.
29			+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup a physical channels as full rate traffic channel or half traffic channel.
2. To setup a physical channel as second full rate traffic channel.
3. To setup a physical channel as the second half rate traffic channel.
4. To bring the MS into U10 state by MT call generic setup procedure.
5. The assignment procedure succeeds.
6. The handover procedure succeeds.
7. To check whether the MS is still in the state U10.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_3_2

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that the MS, when returning to the old channel after handover failure and establishing correctly the link, will remain in the call active state.

Default: OtherEventsFail_01, RcvHdOvAcc

Comments: bearer capability TSPX_BCd is used in the test case. The generic MT call setup procedure is used.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcD, TSPX_MTChRateD)			
3		(TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
5		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+StartCellB_1(C_E_neighbourdefault, C_Rcv,TCV_slot, TCV_tsc, TimingAdv(0), 0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, C_noRestablishment, C_BCC, C_NCC)			1.
7		(TCV_ch1 := OC_SubchOfSdcch4(TSPX_SDCCH4SubDef, C_CellB), TCV_Null := OM_CphMd(TCV_ch1, CphMod_01, TCV_CphKey))			
8		+localtree			
		localtree			
9		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
10	body	+handover			
11		ACTIVATE(RcvHdOvAcc, OtherEventsFail_01)			
12		REPEAT local_hobursts UNTIL [TCV_Res]			
13		L?DL_EstIn	DLEstInd(TCV_chTch)		
14	L_0469	L?DL_DatInHofl	HndOvFlRcv(TCV_chTch, HandOvFail_01)	(P)	2.
15		+CCstatuschk_05(C_U10, TI_02, TCV_chTch)			3.
16		+AssCmdGen_fh(C_Full, TCV_slot, TCV_tsc, TSPX_TCHHSubDef, 7, 7, '000000'B, '000000'B, Frql_02, Frql_02, CellChDes_12, CellChDes_12d, ChMod_sign_iei, MoblAllc_02, MoblAllc_02, CphMod_omit)			
17		+AssCh_failure(TCV_chTch, TCV_AssCmd, FALSE)			4.
18		+CCstatuschk_05(C_U10, TI_02, TCV_chTch)			3.

19		+PostMainLinkRel(TCV_chTch)			
		handover			
20		(TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubDef)), 5), TCV_Res := FALSE)			
21		[TSPC_PGSM OR TSPC_EGSM]			
22		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC, C_arfcnB, C_arfcnB, TSPX_HoRefB, TCV_Pwrlvl_ho, Synchi(C_report_otd, C_non_synchronised) , ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_02iei))		Non-sync HO
23		START T_dly(C_T_T3124)			
24		[TSPC_DCS]			
25		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TCV_slot, TCV_tsc, C_NCC, C_BCC, C_arfcnBd, C_arfcnBd, TSPX_HoRefB, TCV_Pwrlvl_ho, Synchi(C_report_otd, C_non_synchronised) , ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_02iei))		Non-sync HO
26		START T_dly(C_T_T3124)			
		local_hobursts			
27		L?DL_RacInHoacc	HndOvAccRcv(TCV_ch1, HandOverAcc_02(TSPX_HoRefB))		
28		?TIMEOUT			
29		(TCV_Res := TRUE)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4 used as cell B.
2. The expected HANDOVER FAILURE message received on the old channel.
3. To check whether the MS is still in the state U10, if no test case fails in the test step.
4. The expected ASSIGNMENT FAILURE message received on the old channel.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_4_1

Group: GSM_L3_MS_v4220/CC/

Purpose: 1) To verify that an MS supporting the network originated in-call modification procedure, after having received a MODIFY message with a new mode which is not the actual one and cannot be supported by the MS, rejects it by sending a MODIFY REJECT.
2) To verify that an MS not supporting the network originated in-call modification procedure, after having received a MODIFY message, responds with a STATUS message.

Default: OtherEventsFail

Comments: The generic MT call setup procedure is used to bring the MS into U10 state.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2	L_0470	[(TSPX_MT_DualModSvc <> C_AltSpchData) OR(TSPX_MT_DualModSvc <> C_SpchData) OR(TSPX_MT_DualModSvc <> C_AltSpchFax)]		I	not dual mode service
3		[(TSPX_MT_DualModSvc = C_AltSpchData) OR (TSPX_MT_DualModSvc = C_SpchData) OR (TSPX_MT_DualModSvc = C_AltSpchFax)]			
4		+BasicServiceMT(TSPX_MT_DualModSvc, C_Full)			
5		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
6		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
7		+PreEnterCCstateU10(TSPX_SDCCH4SubDef, TCV_Setup_mt, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			
8		(TCV_B := TCV_Setup_mt.bcap2, TCV_Bcap2 := Bcap_NoIEI(TCV_B.iel,TCV_B.extb3,TCV_B.rchr,TCV_B.cs,TCV_B.tm,TCV_B.itc,TCV_B.extb4,TCV_B.spb,TCV_B.strc,TCV_B.dplxm,TCV_B.config,TCV_B.nirr,TCV_B.est,TCV_B.extb5,TCV_B.accid,TCV_B.ra,TCV_B.sacp,TCV_B.extb6,TCV_B.l1id,TCV_B.uil1,TCV_B.sb,TCV_B.extb6a,TCV_B.nsb,TCV_B.nb,TCV_B.ndb,TCV_B.ur,TCV_B.extb6b,TCV_B.ir,TCV_B.nictx,TCV_B.nicrx,TCV_B.pi,TCV_B.extb6c,TCV_B.ce,TCV_B.modemt))			
9		(TCV_B := TCV_Setup_mt.bcap1, TCV_Bcap1 := Bcap_NoIEI(TCV_B.iel,TCV_B.extb3,TCV_B.rchr,TCV_B.cs,TCV_B.tm,TCV_B.itc,TCV_B.extb4,TCV_B.spb,TCV_B.strc,TCV_B.dplxm,TCV_B.config,TCV_B.nirr,TCV_B.est,TCV_B.extb5,TCV_B.accid,TCV_B.ra,TCV_B.sacp,TCV_B.extb6,TCV_B.l1id,TCV_B.uil1,TCV_B.sb,TCV_B.extb6a,TCV_B.nsb,TCV_B.nb,TCV_B.ndb,TCV_B.ur,TCV_B.extb6b,TCV_B.ir,TCV_B.nictx,TCV_B.nicrx,TCV_B.pi,TCV_B.extb6c,TCV_B.ce,TCV_B.mo			

		demt))			
10		L!DL_DatRqModify	ModifySnd(TCV_chTch, ModifyReq_01(TI_02, Bcap_MT(C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, C_I440_450, C_Synchronous, TSPX_TE_stopbit, TSPX_TE_databit, C_4800bs, C_ir_8kbs, TSPX_TE_parity, C_transparent, C_modemt_none)))		
11		[TSPC_InCallMod]			1.
12	L_0471	L?DL_DatInModifyRej	ModifyRejRcv(TCV_chTch, ModifyRj_01(TI_01, TCV_Setup_mt.bcap1))	(P)	
13		+CCstatuschk_05(C_U10, TI_02, TCV_chTch)			
14		+PostMainLinkRel(TCV_chTch)			
15		[NOT TSPC_InCallMod]			2.
16	L_0472	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_08(TI_01, C_U10))	(P)	
17		+CCstatuschk_05(C_U10, TI_02, TCV_chTch)			
18		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. The In-Call modification procedure is supported.
2. The In-Call modification procedure is not supported.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_5_1

Group: GSM_L3_MS_v4220/CC/

Purpose: 1) To verify that the procedure is initiated by the MS in the "active" state of the call. It sends a MODIFY message including the new mode to be changed to; and enters the "mobile originating modify" state. The new mode given in the MODIFY message is one of those already negotiated and agreed during the establishment phase of the call. The MODIFY originating side stops sending Bm-channel information.

2) To verify that upon receipt of the MODIFY COMPLETE message the MS starts sending channel information according to the new call mode and enters the "active" state.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MO_DualModSvc, TSPX_MO_DualModRate)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreModifySetup(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			1.
6	body	+CCstatuschk_05(C_U26, TCV_TI, TCV_chTch)			
7		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_07(TCV_chtype, TCV_ChModb.mode, TSPX_TmSltDef, TSPX_TscDef, TCV_tch_arfcn))		
8	L_0473	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_08(TCV_chtype, TCV_ChModb.mode, TSPX_TmSltDef, TSPX_TscDef, TCV_tch_arfcn))	(P)	2.
9		(TCV_Res := OC_RcsdPresent(TCV_Modify))			
10		[TCV_Res]			
11		L!DL_DatRqModifyCom	ModifyCmpSnd(TCV_chTch, ModifyComp_02(TCV_TI, TCV_Bcap2))		
12		+local_continue			
13		[NOT TCV_Res]			
14		L!DL_DatRqModifyCom	ModifyCmpSnd(TCV_chTch, ModifyComp_03(TCV_TI, TCV_Bcap2))		

15		+local_continue			
		local_continue			
16		(TCV_Null := OM_ChMdModi(TCV_chTch, TCV_ChModb))			
17		START T_dly(2000)			
18		?TIMEOUT T_dly			
19		+CCstatuschk_05(C_U10, TCV_TI, TCV_chTch)			
20		(TCV_Res := OM_BmInfo(TCV_chTch, TCV_ChModb.mode))			
21	L_0474	[TCV_Res]		(P)	3.
22		+PostMainLinkRel(TCV_chTch)			
23	L_0475	[NOT TCV_Res]		(F)	
24		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To attempt a dual mode call and initiate incall modification.
2. The expected CMM ACKNOWLEDGE message received.
3. The MS does start sending Bm channel information according to the new mode.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_5_2

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that upon receipt of the MODIFY REJECT message with the old bearer capability the MS resumes sending Bm-channel information according to the present call mode; resumes interpreting received Bm-channel information according to the present call mode; and enters the "active" state.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MO_DualModSvc, TSPX_MO_DualModRate)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreModifySetup(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			1.
6	body	L!DL_DatRqModifyRej	ModifyRejRqSnd(TCV_chTch, ModifyRjRq_01(TCV_TI, TCV_Bcap1))		
7		+CCstatuschk_05(C_U10, TCV_TI, TCV_chTch)			2.
8		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To attempt a dual mode call and initiate incall modification.
2. To check whether the MS is still in the state U10 and the verdict is assigned in the test step.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_5_3

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that upon receipt of the MODIFY COMPLETE message indicating a call mode which does not correspond to the requested one the MS discards it and takes no action.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MO_DualModSvc, TSPX_MO_DualModRate)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreModifySetup(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			1.
6	body	L!DL_DatRqModifyCom	ModifyCmpSnd(TCV_chTch, ModifyComp_03(TCV_TI, TCV_Bcap2))		2.
7		+CCstatuschk_05(C_U26, TCV_TI, TCV_chTch)			3.
8		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To attempt a dual mode call.
2. The mode (TCV_Bcap1) does not correspond to the requested one (TCV_Bcap2).
3. To check whether the MS does not take any action and the verdict is assigned in the test step.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_5_4

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that upon receipt of the MODIFY REJECT message indicating a call mode which does not correspond to the actual one the MS discards it and takes no action.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MO_DualModSvc, TSPX_MO_DualModRate)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreModifySetup(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			1.
6	body	L!DL_DatRqModifyRej	ModifyRejRqSnd(TCV_chTch, ModifyRjRq_01(TCV_TI, TCV_Bcap1))		2.
7		+CCstatuschk_05(C_U26, TCV_TI, TCV_chTch)			3.
8		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To attempt a dual mode call.
2. The mode (TCV_Bcap1) does not correspond to the actual one (TCV_Bcap2).
3. To check whether the MS does not take any action and the verdict is assigned in the test step.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_5_5

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that upon expiration of T323 the MS shall initiate the procedures for call clearing with cause #102 "recovery on timer expiry".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MO_DualModSvc, TSPX_MO_DualModRate)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreModifySetup(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			1.
6		START T_dly(45000)			
7	body	L?DL_DatInDisc (TCV_Fn1 := DL_DatInDisc.fn) READTIMER T_dly(TCV_Time), CANCEL T_dly	DiscRcv(TCV_chTch, DisconnR(TCV_TI0, Cause_14))		2.
8	L_0476	[(TCV_Time < 27000) OR (TCV_Time >= 33000)]		(F)	3.
9		+PostMainLinkRel(TCV_chTch)			
10	L_0477	[(TCV_Time >= 27000) AND(TCV_Time < 33000)]		(P)	4.
11		+CCstatuschk_05(C_U11, TCV_TI, TCV_chTch)			
12		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To attempt a dual mode call.
2. The expected DISCONNECT message received.
3. Fail, if the T323 timer value is either greater than or equal to 33 seconds, or less than 27 seconds.
4. Pass, if the timeout value of the T323 timer is OK.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_5_6

Group: GSM_L3_MS_v4220/CC/

Purpose: 1. To verify that a CC-entity of the MS in CC-state U26, "Mobile Originating Modify", after successful completion of a channel assignment procedure remains in the call state U26.

2. To verify that upon receipt of the MODIFY COMPLETE message the MS starte sending channel information according to the new call mode and enters the "active" state.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MO_DualModSvc, TSPX_MO_DualModRate)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+local_2ndTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TCV_ChModb, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
6		+PreModifySetup(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			1.
7		+local_continue			
		local_continue			
8		+assign			
9		+CCstatuschk_05(C_U26, TCV_TI, TCV_chTch)			2.
10		(TCV_Res := OC_RcsdPresent(TCV_Modify))			
11		[TCV_Res]			
12		L!DL_DatRqModifyCom (DL_DatRqModifyCom.msg.rcsd := TCV_Modify.rcsd)	ModifyCmpSnd(TCV_chTch, ModifyComp_02(TCV_TI, TCV_Bcap2))		
13		+localtree1			
14		[NOT TCV_Res]			
15		L!DL_DatRqModifyCom	ModifyCmpSnd(TCV_chTch, ModifyComp_03(TCV_TI, TCV_Bcap2))		
16		+localtree1			
		localtree1			
17		+CCstatuschk_05(C_U10, TCV_TI, TCV_chTch1)			3.
18		+PostMainLinkRel(TCV_chTch1)			
		assign			
19		[TSPC_PGSM OR TSPC_EGSM]			
20		(TCV_AssCmd := AsgnCmd_nfh(TCV_chtype,			

		TCV_slt2, TSPX_TscDef, 7, C_arfcn_tchA, CellChDes_Omit, TCV_ChModb, StartingTm_omit, CphMod_omit))			
21		+AssCh_complete(TCV_chTch, TCV_chTch1, TCV_AssCmd)			
22		[TSPC_DCS]			
23		(TCV_AssCmd := AsgnCmd_nfh(TCV_chtype, TCV_slt2, TSPX_TscDef, 3, C_arfcn_tchAd, CellChDes_Omit, TCV_ChModb, StartingTm_omit, CphMod_omit))			
24		+AssCh_complete(TCV_chTch, TCV_chTch1, TCV_AssCmd)			
		local_2ndTCH(sub:BITSTRING; acttype:BITSTRING; slot:SN; tsc:TSC; chmod:CHMOD; arfcng, arfcnd: INTEGER; ta:TA; babr, cch_con, bpm:B_3)			
25		(TCV_slt2 := INT_TO_BIT(((1+ BIT_TO_INT(slot))MOD 8),3))			
26		[TCV_ChRate = C_Full]			
27		+FullRateCh_A_2(acttype, TCV_slt2, tsc, chmod, FreqTCH(arfcng), FreqTCH(arfcnd), ta, babr, cch_con, bpm)			
28		[TCV_ChRate = C_Half]			
29		+HalfRateCh_A_2(sub, acttype, TCV_slt2, tsc, chmod, FreqTCH(arfcng), FreqTCH(arfcnd), ta, babr, cch_con, bpm)			

Detailed Comments:

1. To attempt a dual mode call.
2. To check whether the MS is still in the state U26.
3. T0 check whether the MS is now in the state U10 and the verdict is assigned in the test step.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_5_7
Group: GSM_L3_MS_v4220/CC/
Purpose: To verify that a CC-entity of the MS in CC-state U26, "Mobile Originating Modify", when returning to the old channel after handover failure and having established the link, remains in the call state U26.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MO_DualModSvc, TSPX_MO_DualModRate)			
3		(TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
5		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+PreModifySetup(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			1.
7	body	+FullRateCh_A_2(C_Rcv, TSPX_TmSltC, TSPX_TscC, ChMod_speech, FreqTCH(C_arfcn_tchA), FreqTCH(C_arfcn_tchAd), TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			2.
8		+local_sendHoCMD			
9		(TCV_Cnt := 0)			
10		REPEAT local_Hoacc UNTIL [TCV_Cnt = 4]			
11	L_0478	L?DL_DatInHofl	HndOvFlRcv(TCV_chTch, HandOvFail_01)	(P)	5.
12		+CCstatuschk_05(C_U26, TCV_TI, TCV_chTch)			6.
13		+PostMainLinkRel(TCV_chTch)			
		local_Hoacc			
14		L?DL_RacInHoacc	HndOvAccRcv(TCV_chTch1, HandOverAcc_02(TSPX_HoRefA))		4.
15		(TCV_Cnt := TCV_Cnt +1)			
		local_sendHoCMD			
16		(TCV_Tchtype := '00001'B)			
17		[TSPC_PGSM OR TSPC_EGSM]			
18		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC,		3.

			TSPX_TscC, C_NCC, C_BCC, C_arfcnA, C_arfcn_tchA, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
19		[TSPC_DCS]			
20		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch, HandOverCmd_nfh(TCV_Tchtype, TSPX_TmSltC, TSPX_TscC, C_NCC, C_BCC, C_arfcnAd, C_arfcn_tchAd, TSPX_HoRefA, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		3.

Detailed Comments:

1. To attempt a dual mode call.
2. To setup a receiving only TCH/F for handover.
3. To handover to the channel which does not respond.
4. To check that the MS sends HANDOVER ACCESS messages on the new channel.
5. The expected HANDOVER FAILURE message received.
6. To check whether the MS is still in the state U26.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_5_8

Group: GSM_L3_MS_v4220/CC/

Purpose: To verify that a CC entity of a MS in CC-state U26, "Mobile Originating Modify", having received an unknown message from its peer entity returns a STATUS message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MO_DualModSvc, TSPX_MO_DualModRate)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreModifySetup(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			1.
6	body	L!DL_DatRqUndefCC	UndefCC(TCV_chTch, UndefCC_02(TCV_TI))		2.
7	L_0479	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_08(TCV_TI 0, C_U26))	(P)	3.
8		+CCstatuschk_05(C_U26, TCV_TI, TCV_chTch)			4.
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To attempt a dual mode call.
2. To send a message which message type is not defined for CC.
3. Received expected CC STATUS message with state U26.
4. To check whether the MS is still in the state U26 and assign the verdict.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_1_4_5_9

Group: GSM_L3_MS_v4220/CC/

Purpose: 1) To verify that a CC entity of a MS in CC-state U26, "Mobile Originating Modify", upon receipt of a RELEASE COMPLETE message with valid cause value, shall enter CC state U0, "Null".

2)To verify that on returning to idle mode, the CC entities relating to the seven mobile originating transcation identifiers shall be in state U0, "Null".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MO_DualModSvc, TSPX_MO_DualModRate)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreModifySetup(TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TSPX_CKSNDef, TSPX_RANDDef)			1.
6	body	+CCstatuschk_05(C_U26, TCV_TI, TCV_chTch)			
7		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		2.
8		+CheckTIsInStateU0(TRUE, TCV_chTch)			
9		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To attempt a dual mode call and initiate incall modification.
2. To send RELEASE COMPLETE message.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_2_1
Group: GSM_L3_MS_v4220/CC/
Purpose: The purpose of this test is to verify that the MS can correctly perform a call re-establishment procedure.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		(TCV_ch1 := OC_SubchOfSdcch4(TSPX_SDCCH4SubDef, C_CellB), TCV_chTch1 := C_FACCHF_B_1)			
5		+StartCellB_1(C_E_neighbourdefault, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, C_Restablishment, C_BCC, C_NCC)			
6		+CCConfigTCH_B(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltC, TSPX_TscC, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchB, C_arfcn_tchBd)			
7		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
8		+ltree			
		ltree			
9		+InitCall(TCV_Service)			
10		+CCEstablishMO_SDCCH4(TSPX_SDCCH4SubDef, TimingAdv(0))			
11		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
12		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_04)		
13		ACTIVATE(OtherEventsFail)			Restore Normal default
14		+Authentication(TCV_ch, TSPX_CKSNDef, TSPX_RANDDef)			
15		+Ciphering_on(TCV_ch)			
16		+localtree			
		localtree			
17		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
18		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
19		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			

20		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
21		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
22		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
23		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
24	body	+Switchcell(C_CellA, C_CellB, C_LAC_2, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TCV_Ccd0B, TSPX_IMSI)			Cell A still exists.
25		START T_dly(C_T_Wait)			
26		?TIMEOUT T_dly			
27		+localtree1			
		localtree1			
28		(TCV_Null := OM_Deactivate(TCV_chTch, TCV_sacchTch), TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
29	L_0480	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_10)	(P)	1.
30		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
31		L!DL_UdatRqImmass	ImmAss(C_AGCH_B_1, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
32		L?DL_EstInCmreRq	CmreReq_02(MiTmsi_01, LocAreaIdiei(C_MCC_1, C_PLMN_1, C_LAC_1), TSPX_CKSNDef)		
33		ACTIVATE(OtherEventsFail)			Restore Normal default
34		+Ciphering_on(TCV_ch1)			
35		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
36		+AssCh_complete(TCV_ch1, TCV_chTch1, TCV_AssCmd)			
37		(TCV_Res := OO_TCHThroConnCHK())			
38	L_0481	[NOT TCV_Res]		(F)	2.
39		+PostMainLinkRel(TCV_chTch1)			
40	L_0482	[TCV_Res]		(P)	
41		L!DL_DatRqDisc	DiscSnd(TCV_chTch1, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
42		L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))		
43		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch1, ReleaseCmp_08(TCV_TI))		
44		+PostMainLinkRel(			

		TCV_chTch1)			
Detailed Comments:	1. The expected call re-establishment is started. 2. The bearer channel is not through connected, fail.				

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_2_2
Group: GSM_L3_MS_v4220/CC/
Purpose: The purpose of this test is to verify that the MS does not attempt call re-establishment when it is not allowed to take place because of the unavailability of a cell allowing call re-establishment.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcC, TSPX_MOChRateC)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+CCEstablishMO_SDCCH4(TSPX_SDCCH4SubDef, TimingAdv(0))			
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_04)		
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		+Authentication(TCV_ch, TSPX_CKSNDef, TSPX_RANDDef)			
11		+Ciphering_on(TCV_ch)			
12		+localtree			
		localtree			
13		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
14		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
15		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
16		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
17		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
18		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
19		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
20		(TCV_Null := OM_Deactivate(TCV_chTch, TCV_sacchTch), TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
21		START T_dly(30000)			
22	L_0483	L?DL_RacInChRq CANCEL T_dly	ChReq(ChRequest_02)	F	2.
23	L_0484	?TIMEOUT T_dly		P	3.

Detailed Comments:	1. Default parameters, call reestablishment not allowed. 2. Re-establishment is attempted, fail. 3. No re-establishment attempting, pass.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_2_3
Group: GSM_L3_MS_v4220/CC/
Purpose: The purpose of this test is to verify that the MS does not attempt call re-establishment when it is not allowed to take place because of the call control state.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcB, TSPX_MOChRateB)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		+CCEstablishMO_SDCCH4(TSPX_SDCCH4SubDef, TimingAdv(0))			
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_04)		
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		+Authentication(TCV_ch, TSPX_CKSNDef, TSPX_RANDDef)			
11		+Ciphering_on(TCV_ch)			
12		+localtree			
		localtree			
13		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
14		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
15		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
16		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
17		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
18	body	(TCV_Null := OM_Deactivate(TCV_chTch, TCV_sacchTch), TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
19		START T_dly(30000)			
20	L_0485	L?DL_RacInChRq CANCEL T_dly	ChReq(ChRequest_02)	F	2.
21	L_0486	?TIMEOUT T_dly		P	3.

Detailed Comments:
1. Default parameters, reestablishment allowed.
2. Re-establishment is attempted, fail.
3. No re-establishment attempting, pass.

Test Case Dynamic Behaviour

Test Case Name: TC_26_8_3
Group: GSM_L3_MS_v4220/CC/
Purpose: To verify that inclusion of the 'user-user' information element in a either of the down link messages, SETUP or DISCONNECT causes no adverse effects on the operation of the MS.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcF, TSPX_MTChRateF)			
3		(TCV_Setup_mt.uu := UuInfo, TCV_Setup_mt.sig := Signal_01)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
5		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+CCEstablishMT_SDCCH4(TimingAdv(0), TSPX_SDCCH4SubDef, TSPX_CKSNDef)			
7		+Authentication(TCV_ch, TSPX_CKSNDef, TSPX_RANDDef)			
8		+Ciphering_on(TCV_ch)			
9	body	L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		
10	L_0487	L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))	(P)	1.
11		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
12		+localtree			
13		L?DL_DatInAlert	AlertRcv(AlertingInd_02(TI_01))		
14		(TCV_Null := OO_HookOff())			
15		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
16		+localtree			
		localtree			
17		+CCAssignTCH(TSPX_TmSltDef, TSPX_TscDef)			
18		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
19		START T_dly(30000)			
20		?TIMEOUT T_dly			
21		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TI_02, Cause_01, ProgInd_omit, UuInfo))		
22	L_0488	L?DL_DatInRel	ReleaseRcv(Release_10(TI_01))	(P)	2.

23		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TI_02))		
24		+PostMainLinkRel(TCV_chTch)			
Detailed Comments:		1. The MS responds to the SETUP message with user-user information IE correctly. 2. The MS responds to the DISCONNECT message with user-user information IE correctly.			

Test Group StructureProc

Test Case Dynamic Behaviour

Test Case Name: TC_26_9_2

Group: GSM_L3_MS_v4220/StructureProc/

Purpose: 1) To verify that the MS in MM state "idle, updated" with a TMSI assigned, when made to initiate a call for a selected teleservice that is supported by the MS as declared in a PICS/PIXIT statement, displays the dialled number in the way described in a PICS/PIXIT statement.

2) To verify that the MS in MM state "idle, updated" and in RR idle mode, with a TMSI assigned, when made to initiate a call for a selected teleservice that is supported by the MS as declared in a PICS/PIXIT statement, starts to initiate an immediate assignment procedure by sending the CHANNEL REQUEST message with correct establishment cause.

3) To verify that subsequently after receipt of an IMMEDIATE ASSIGNMENT message allocating an SDCCH, after completion of establishment of the main signalling link, after having sent a CM SERVICE REQUEST message, after having successfully performed the authentication and cipher mode setting procedures, the MS sends a SETUP message with correct parameters.

4) To verify that subsequently, after receipt of a CALL PROCEEDING message and of an ASSIGNMENT COMMAND message allocating an appropriate TCH, after having completed the traffic channel early assignment procedure by replying with the ASSIGNMENT COMPLETE message, after receipt of an ALERTING message and a CONNECT message, the MS returns a CONNECT ACKNOWLEDGE message.

5) To verify that subsequently the MS has attached the user connection to the radio path. (This is checked by verifying that there is a point in time after transmission of the first L2 frame containing the (complete) CONNECT message, where the MS is sending appropriate speech or data frames whenever it doesn't have to transmit or acknowledge an I frame on layer 2 of the FACCH.)

6) To verify that subsequently upon the network initiating call clearing by sending a DISCONNECT message, the MS proceed to release the call with RELEASE.

7) To verify that subsequently, on receipt of a RELEASE COMPLETE message followed by a CHANNEL RELEASE message, the MS disconnects the main signalling link.

These test purposes are tested for all rates supported by the MS (full rate/half rate).

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2	body	[TSPC_Serv_TS11]			
3		+BasicServiceMO(C_Telephony, C_Full)			
4		+local_testfullrate			1.
5		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
6		[TSPC_DualRate]			
7		+BasicServiceMO(C_Telephony, C_Half)			
8		+local_testhalfrate			2.
9		[NOT TSPC_DualRate]			
10		[TSPC_Serv_TS62]			TS62
11		+BasicServiceMO(C_AutoFax, C_Full)			
12		+local_testfullrate			3.
13	L_0489	[C_Yes]		I	
		local_testfullrate			
14		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02,			

		BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNA, TSPX_RANDA, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
15		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
16		+local_initcall			
		local_testhalfrate			
17		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			7.
18		(TCV_Null := OM_ChMdModi(TCV_chTch, TCV_ChMod))			
19		+local_initcall			
		local_initcall			
20		(TCV_Null := OO_DialCalledNum())			4.
21		[TSPC_CalledNumDisp]			
22		(TCV_Res := OO_CalledNumCHK())			5.
23	L_0490	[NOT TCV_Res]		(F)	
24	L_0491	[TCV_Res]		(P)	
25		+InitCall(TCV_Service)			6.
26		+localtree			
27		[NOT TSPC_CalledNumDisp]			
28		+InitCall(TCV_Service)			
29		+localtree			
		localtree			
30	L_0492	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	
31		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
32		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
33		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_04)		
34		ACTIVATE(OtherEventsFail)			Restore Normal default
35		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNA, TSPX_RANDA))		
36		L?DL_DatInAuthRes(TCV_Sres:=DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
37		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, TSPX_RANDA))			
38	L_0493	[NOT TCV_Res]		(F)	
39		+PostMainLinkRel(TCV_ch)			
40	L_0494	[TCV_Res]		(P)	
41		+localtree1			
		localtree1			
42		+Ciphering_on(TCV_ch)			
43		+RcvSetupOrEsetup(TCV_Setup_mo,			

		TCV_Esetup, TCV_Ecall)			
44	L_0495	L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)	(P)	
45		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
46		+AssCh_complete(TCV_ch,TCV_chTch,TCV_AssCmd)			8.
47		+localtree2			
		localtree2			
48		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
49		[TSPC_AlertInd]			
50		(TCV_Res := OO_AltIndCHK())			
51	L_0496	[NOT TCV_Res]		(F)	
52		+PostMainLinkRel(TCV_chTch)			
53		[TCV_Res]			
54		+localtree3			
55		[NOT TSPC_AlertInd]			
56		+localtree3			
		localtree3			
57		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
58	L_0497	L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))	(P)	
59		(TCV_Res := OO_TCHThroConnCHK())			
60	L_0498	[NOT TCV_Res]		(F)	
61		+PostMainLinkRel(TCV_chTch)			
62	L_0499	[TCV_Res]		(P)	
63		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
64	L_0500	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	
65		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
66		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To perform the test on full rate telephony service.
2. To perform the test on half rate telephony.
3. To perform the test on TS62 service.
4. To enter the called party number.
5. To check whether the MS displays the called party number correctly.
6. To initiate the call.
7. To setup the previous full rate traffic channel into half rate traffic channel.
8. Early assignment.

Test Case Dynamic Behaviour

Test Case Name: TC_26_9_3

Group: GSM_L3_MS_v4220/StructureProc/

Purpose: 1) To verify that the MS in MM state "idle, updated" and in RR idle mode with a TMSI assigned, when made to initiate a call for a selected teleservice that is supported by the MS as declared in a PICS/PIXIT statement, starts to initiate an immediate assignment procedure by sending the CHANNEL REQUEST message.

2) To verify that subsequently after receipt of an IMMEDIATE ASSIGNMENT message allocating an SDCCH, after completion of establishment of the main signalling link, after having sent a CM SERVICE REQUEST message, after having successfully performed authentication and cipher mode setting procedures, after having sent a SETUP message, after having received a CALL PROCEEDING message followed by an ALERTING message and an ASSIGNMENT COMMAND message allocating an appropriate TCH, the MS sends an ASSIGNMENT COMPLETE message.

3) To verify that subsequently, after the suite of actions specified in test purposes 1 and 2, the MS after receiving a CONNECT message returns a CONNECT ACKNOWLEDGE message.

4) To verify that after the suite of actions specified in test purposes 1 and 2, the MS after receiving a CONNECT message attaches the user connection to the radio path. (This is checked by verifying that there is a point in time after transmission of the first L2 frame containing the (complete) CONNECT message, where the MS is sending appropriate speech or data frames whenever it doesn't have to transmit or acknowledge an I frame on layer 2 of the FACCH.)

These test purposes are tested for all rates supported by the MS (full rate/half rate).

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2	body	[TSPC_Serv_TS11]			
3		+BasicServiceMO(C_Telephony, C_Full)			
4		+local_testfullrate			1.
5		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
6		[TSPC_DualRate]			
7		+BasicServiceMO(C_Telephony, C_Half)			
8		+local_testhalfrate			2.
9		[NOT TSPC_DualRate]			
10		[TSPC_Serv_TS62]			TS62
11		+BasicServiceMO(C_AutoFax, C_Full)			
12		+local_testfullrate			3.
13	L_0501	[C_Yes]		I	
		local_testfullrate			
14		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNA, TSPX_RANDA, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
15		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
16		+local_initcall			

		local_testhalfrate			
17		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			7.
18		+local_initcall			
		local_initcall			
19		(TCV_Null := OO_DialCalledNum())			4.
20		[TSPC_CalledNumDisp]			
21		(TCV_Res := OO_CalledNumCHK())			5.
22	L_0502	[NOT TCV_Res]		(F)	
23	L_0503	[TCV_Res]		(P)	
24		+InitCall(TCV_Service)			6.
25		+localtree			
26		[NOT TSPC_CalledNumDisp]			
27		+InitCall(TCV_Service)			
28		+localtree			
		localtree			
29	L_0504	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	
30		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
31		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
32		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_04)		
33		ACTIVATE(OtherEventsFail)			Restore Normal default
34		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNA, TSPX_RANDA))		
35		L?DL_DatInAuthRes (TCV_Sres := DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
36		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, TSPX_RANDA))			
37	L_0505	[NOT TCV_Res]		(F)	
38		+PostMainLinkRel(TCV_ch)			
39		[TCV_Res]			
40		+localtree1			
		localtree1			
41		+Ciphering_on(TCV_ch)			
42		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
43	L_0506	L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)	(P)	
44		L!DL_DatRqAlert	AlertSnd(TCV_ch, Alerting_01(TCV_TI))		
45		[TSPC_AlertInd]			
46		(TCV_Res := OO_AltIndCHK())			
47	L_0507	[NOT TCV_Res]		(F)	
48		+localtree2			
49		[TCV_Res]			
50		+localtree2			
51		[NOT TSPC_AlertInd]			
52		+localtree2			

		localtree2			
53		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
54		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
55		+localtree3			
		localtree3			
56		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
57	L_0508	L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))	(P)	
58		(TCV_Res := OO_TCHThroConnCHK())			
59	L_0509	[NOT TCV_Res]		(F)	
60		+PostMainLinkRel(TCV_chTch)			
61	L_0510	[TCV_Res]		(P)	
62		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
63	L_0511	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	
64		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
65		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To perform the test on full rate telephony service.
2. To perform the test on half rate telephony.
3. To perform the test on TS62 service.
4. To enter the called party number.
5. To check whether the MS displays the called party number correctly.
6. To initiate the call.
7. To setup the previous full rate traffic channel into half rate traffic channel.
8. Later assignment

Test Case Dynamic Behaviour

Test Case Name: TC_26_9_4

Group: GSM_L3_MS_v4220/StructureProc/

Purpose: 1) To verify that the MS in MM state "idle, updated" and in RR idle mode with a TMSI assigned, after being paged by the network on the correct paging subchannel, after initiating the immediate assignment procedure by sending the CHANNEL REQUEST message, after receipt of an IMMEDIATE ASSIGNMENT message allocating an SDCCH, after having sent a PAGING RESPONSE message on the allocated SDCCH, after having performed successful authentication and cipher mode setting procedures, after receipt of a SETUP message not containing a signal information element, returns a CALL CONFIRMED message.

2) To verify that subsequently, the SS sending an ASSIGNMENT COMMAND message, the MS successfully continues a mobile terminating call establishment with early assignment of traffic channel

a) by replying to the ASSIGNMENT COMMAND with an ASSIGNMENT COMPLETE message, and

b) by continuing the call establishment by either sending a CONNECT message and through connecting the TCH in both directions, or, sending an ALERTING message.

3) To verify that if after sending a CALL PROCEEDING message, the MS sends an ALERTING message during MTC establishment with early assignment, it generates an alerting indication.

4) To verify that if an ALERTING had been sent, subsequently, when the user accepts the call (possibly internal action as declared in PICS/PIXIT statement), the MS returns a CONNECT message.

5) To verify that the MS:

- if the call is a speech call: after sending the CONNECT message has through connected the TCH in both directions (this is checked by verifying that after transmission of the first L2 frame containing the (complete) CONNECT message, the MS is sending appropriate speech or data frames whenever it doesn't have to transmit or acknowledge an I frame on layer 2 of the FACCH.)

- if the call is a data call: after receipt of a subsequent CONNECT ACKNOWLEDGE message through connects the TCH in both directions (this is checked by verifying that there is a point in time after transmission of the first L2 frame containing the (complete) CONNECT ACKNOWLEDGE message, where the MS is sending appropriate speech or data frames whenever it doesn't have to transmit or acknowledge an I frame on layer 2 of the FACCH.)

6) To verify that subsequently, the MS can initiate call clearing by sending a DISCONNECT message.

7) To verify that the MS in this phase of call release, upon receipt of a RELEASE message, returns a RELEASE COMPLETE message.

8) To verify that subsequently the MS, upon receipt of a CHANNEL RELEASE message, disconnects the main signalling link.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2	body	[TSPC_Serv_TS11]			
3		+BasicServiceMT(C_Telephony, C_Full)			
4		+local_testfullrate			1.
5		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
6		[TSPC_DualRate]			
7		+BasicServiceMT(C_Telephony, C_Half)			
8		+local_testhalfrate			2.
9		[NOT TSPC_DualRate]			
10		[TSPC_Serv_TS62]			TS62
11		+BasicServiceMT(C_AutoFax, C_Full)			
12		+local_testfullrate			3.
13	L_0512	[C_Yes]		I	

		local_testfullrate			
14		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNA, TSPX_RANDA, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
15		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
16		(TCV_Null := OM_ChMdModi(TCV_chTch, TCV_ChMod))			
17		+localtree(C_Full)			
		local_testhalfrate			
18		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			4.
19		+localtree(C_Half)			
		localtree(rate:IA5String)			
20		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
21		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
22		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
23		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
24		L?DL_EstInPgRes	PagingRes(PagingRes_03(TSPX_CKSNDef))		
25		ACTIVATE(OtherEventsFail)			Restore Normal default
26		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNA, TSPX_RANDA))		
27		L?DL_DatInAuthRes(TCV_Sres := DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
28		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, TSPX_RANDA))			
29	L_0513	[NOT TCV_Res]		(F)	
30		+PostMainLinkRel(TCV_ch)			
31	L_0514	[TCV_Res]		(P)	5.
32		+localtree1(rate)			
		localtree1(rate:IA5String)			
33		+Ciphering_on(TCV_ch)			
34		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		6.
35		L?DL_DatInCallCo(TCV_CallCfm :=	CallCfm(		

		DL_DatInCallCo.msg)	CallConfirm_01(TI_01))		
36		[TSPX_Immconn]			
37		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
38		+asstrafficch(rate)			7.
39		+localtree3			
40		[NOT TSPX_Immconn]			
41		+asstrafficch(rate)			7.
42		L?DL_DatInAlert	AlertRcv(AlertingInd_01(TI_01))		
43		[TSPC_AlertInd]			
44		(TCV_Res := OO_AltIndCHK())			
45	L_0515	[NOT TCV_Res]		(F)	8.
46		+PostMainLinkRel(TCV_chTch)			
47	L_0516	[TCV_Res]		(P)	
48		(TCV_Null := OO_HookOff())			9.
49		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
50		+localtree3			
51		[NOT TSPC_AlertInd]			
52		(TCV_Null := OO_HookOff())			9.
53		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
54		+localtree3			
		localtree3			
55		[TCV_Setup_mt.bcap1.itc = '000'B]			Speech Call
56		(TCV_Res := OO_TCHThroConnCHK())			
57	L_0517	[NOT TCV_Res]		(F)	
58		+PostMainLinkRel(TCV_chTch)			
59	L_0518	[TCV_Res]		(P)	
60		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
61		+localtree5			
62		[TCV_Setup_mt.bcap1.itc <> '000'B]			Data Call
63		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
64		(TCV_Res := OO_TCHThroConnCHK())			
65	L_0519	[NOT TCV_Res]		(F)	
66		+PostMainLinkRel(TCV_chTch)			
67	L_0520	[TCV_Res]		(P)	
68		+localtree5			
		localtree5			
69		(TCV_Null := OO_TermCall())			
70		L?DL_DatInDisc	DiscRcv(TCV_chTch, DisconnR(TI_01, Cause_Def))		
71		L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_08(TI_02))		
72		L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TI_01))		
73		+PostMainLinkRel(TCV_chTch)			
		asstrafficch(rate:IA5String)			

74		(TCV_Res := FALSE)			
75		+AssCmdGen(TCV_cellid, rate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
76		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			

Detailed Comments:

1. To perform the test on full rate telephony service.
2. To perform the test on half rate telephony.
3. To perform the test on TS62 service.
4. To setup the previous full rate channel into half rate channel.
5. Authentication is OK.
6. SETUP message withourt signal IE.
7. Early assignment.
8. Alerting indication not correct.
9. To accept the call at the MS.

Test Case Dynamic Behaviour

Test Case Name: TC_26_9_5

Group: GSM_L3_MS_v4220/StructureProc/

Purpose: 1) To verify that the MS in "Idle, Updated" state with a TMSI assigned, after being paged by the network on the correct paging subchannel, after initiating the immediate assignment procedure by sending the CHANNEL

REQUEST message, after receipt of an IMMEDIATE ASSIGNMENT message allocating an SDCCH, after having established the main signalling link, after having sent a PAGING RESPONSE message, after having performed successful authentication and cipher mode setting procedures, after receipt of a SETUP message containing a signal information element, returns a CALL CONFIRMED message followed by

- an ALERTING message

- or a CONNECT message.

2) To verify that in the situation of test purpose 1, if the MS sends an ALERTING message, the MS generates an alerting indication in the way described in a PICS/PIXIT statement.

3) To verify that subsequently the MS, if it had not yet sent a CONNECT message, upon acceptance of the call, sends a CONNECT message.

4) To verify that subsequently after receipt of an ASSIGNMENT COMMAND, the MS sends an ASSIGNMENT COMPLETE message.

5) To verify that subsequently the MS

- if the call is a speech call: after sending the ASSIGNMENT COMPLETE message has through connected the TCH in both directions (this is checked by verifying that after transmission of the first L2 frame containing the (complete) ASSIGNMENT COMPLETE message, the MS is sending appropriate speech or data frames whenever it doesn't have to transmit or acknowledge an I frame on layer 2 of the FACCH.)

- if the call is a data call: after receipt of a subsequent CONNECT ACKNOWLEDGE message through connects the TCH in both directions (this is checked by verifying that there is a point in time after transmission of the first L2 frame containing the (complete) CONNECT ACKNOWLEDGE message, where the MS is sending appropriate speech or data frames whenever it doesn't have to transmit or acknowledge an I frame on layer 2 of the FACCH.)

These test purposes are tested for all rates supported by the MS (full rate/half rate).

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2	body	[TSPC_Serv_TS11]			
3		+BasicServiceMT(C_Telephony, C_Full)			
4		+local_testfullrate			1.
5		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey), TCV_Null := OM_CphMd(TCV_chTch, CphMod_02, TCV_CphKey))			
6		[TSPC_DualRate]			
7		+BasicServiceMT(C_Telephony, C_Half)			
8		+local_testhalfrate			2.
9		[NOT TSPC_DualRate]			
10		[TSPC_Serv_TS62]			TS62
11		+BasicServiceMT(C_AutoFax, C_Full)			
12		+local_testfullrate			3.
13	L_0521	[C_Yes]		I	
		local_testfullrate			
14		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG,			

		C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNA, TSPX_RANDA, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
15		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_sign, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
16		(TCV_Null := OM_ChMdModi(TCV_chTch, TCV_ChMod))			
17		+localtree(C_Full)			
		local_testhalfrate			
18		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			4.
19		+localtree(C_Half)			
		localtree(rate:IA5String)			
20		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
21		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
22		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
23		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
24		L?DL_EstInPgRes	PagingRes(PagingRes_03(TSPX_CKSNDef))		
25		ACTIVATE(OtherEventsFail)			Restore Normal default
26		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNA, TSPX_RANDA))		
27		L?DL_DatInAuthRes(TCV_Sres:=DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
28		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, TSPX_RANDA))			
29	L_0522	[NOT TCV_Res]		(F)	
30		+PostMainLinkRel(TCV_ch)			
31	L_0523	[TCV_Res]		(P)	5.
32		+localtree1(rate)			
		localtree1(rate:IA5String)			
33		+Ciphering_on(TCV_ch)			
34		(TCV_Setup_mt.sig := Signal_01)			
35		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		6.
36		L?DL_DatInCallCo (TCV_CallCfm := DL_DatInCallCo.msg)	CallCfm(CallConfirm_01(TI_01))		
37		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
38		+localtree2(rate)			
39		L?DL_DatInAlert	AlertRcv(AlertingInd_01(TI_01))		

40		[TSPC_AlertInd]			
41		(TCV_Res := OO_AltIndCHK())			
42	L_0524	[NOT TCV_Res]		(F)	8.
43		+localtree2(rate)			
44	L_0525	[TCV_Res]		(P)	
45		(TCV_Null := OO_HookOff())			9.
46		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
47		+localtree2(rate)			
48		[NOT TSPC_AlertInd]			
49		(TCV_Null := OO_HookOff())			9.
50		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
51		+localtree2(rate)			
		localtree2(rate:IA5String)			
52		+AssCmdGen(TCV_cellid, rate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
53		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
54		+localtree3			
		localtree3			
55		[TCV_Setup_mt.bcap1.itc = '000'B]			
56		(TCV_Res := OO_TCHThroConnCHK())			
57	L_0526	[NOT TCV_Res]		(F)	
58		+PostMainLinkRel(TCV_chTch)			
59	L_0527	[TCV_Res]		(P)	
60		+localtree4			
61		[TCV_Setup_mt.bcap1.itc <> '000'B]			
62		+localtree4			
		localtree4			
63		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
64		[TCV_Setup_mt.bcap1.itc <> '000'B]			
65		(TCV_Res := OO_TCHThroConnCHK())			
66	L_0528	[NOT TCV_Res]		(F)	
67		+PostMainLinkRel(TCV_chTch)			
68	L_0529	[TCV_Res]		(P)	
69		+PostMainLinkRel(TCV_chTch)			
70		[TCV_Setup_mt.bcap1.itc = '000'B]			
71		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To perform the test on full rate telephony service.
2. To perform the test on half rate telephony.
3. To perform the test on TS62 service.
4. To setup the previous full rate channel into half rate channel.
5. Authentication is OK.
6. SETUP message without signal IE.
7. Late assignment.
8. Alerting indication not correct.
9. To accept the call at the MS.

Test Case Dynamic Behaviour

Test Case Name: TC_26_9_6_1_1

Group: GSM_L3_MS_v4220/StructureProc/

Purpose: 1) To verify that an MS supporting speech in the MM state "idle, updated", when made to call the number 112, sends a CHANNEL REQUEST message with establishment cause "emergency call".

2) To verify that after assignment of a dedicated channel the first layer message sent by the MS on the assigned dedicated channel is a CM SERVICE REQUEST message specifying the correct CKSN and TMSI, with CM Service Type "emergency call establishment" .

3) To verify that authentication and cipher mode setting are performed successfully.

4) To verify that after cipher mode setting acceptance by the SS, the MS sends an EMERGENCY SETUP message.

5) To verify that subsequently, the SS having sent a CALL PROCEEDING message and then an ALERT message and having initiated the assignment procedure of an appropriate speech traffic channel, which, if the MS supports both TCH/FS and TCH/HS, is at the preferred rate, the MS performs correctly that assignment procedure.

6) To verify subsequent correct performance of a connect procedure.

7) To verify that subsequently the MS has through connected the TCH in both directions.

8) To verify that the call is cleared correctly.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(C_EmgCall, TSPX_EmgCallRate)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNA, TSPX_RANDA, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4	body	+InitCall(TCV_Service)			
5	L_0530	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_16)	(P)	2.
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
8	L_0531	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_06(TSPX_CKSNA))	(P)	3.
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNA, TSPX_RANDA))		

11		L?DL_DatInAuthRes(TCV_Sres := DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
12		+local_continue			
		local_continue			
13		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, TSPX_RANDA))			
14	L_0532	[NOT TCV_Res]		(F)	4.
15		+PostMainLinkRel(TCV_ch)			
16	L_0533	[TCV_Res]		(P)	5.
17		+Ciphering_on(TCV_ch)			
18		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			6.
19		+local_preferred_rate			
20		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
21		L!DL_DatRqAlert	AlertSnd(TCV_ch, Alerting_01(TCV_TI))		
22		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
23		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
24		+local_continue1			
		local_continue1			
25		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
26	L_0534	L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))	(P)	
27		(TCV_Res := OO_TCHThroConnCHK())			
28	L_0535	[NOT TCV_Res]		(F)	7.
29		+PostMainLinkRel(TCV_chTch)			
30	L_0536	[TCV_Res]		(P)	8.
31		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
32	L_0537	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	
33		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
34		+PostMainLinkRel(TCV_chTch)			
		local_preferred_rate			
35		+set_radio_chan_req			
36		[TSPC_DualRate]			
37		[TCV_Rchr = '11'B]			9.
38		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd,TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
39		(TCV_ChRate := C_Full)			
40		[TCV_Rchr = '10'B]			10.
41		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd,			

		TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
42		(TCV_ChRate := C_Half)			
43		[NOT TSPC_DualRate]			
44		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd,TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
45		(TCV_ChRate := C_Full)			
		set_radio_chan_req			
46		[NOT TCV_Ecall]			
47		(TCV_Rchr := TCV_Setup_mo1.bcap1.rchr)			
48		[TCV_Ecall]			
49		(TCV_Rchr := TCV_Esetup1.bcap.rchr)			

Detailed Comments:

1. To setup a physical channels as BCCH CCCH and SDCCH4 combined channel.
2. CHANNEL REQUEST with "emergency call establishment" cause received.
3. CM SERVICE REQUEST with "emergency call establishment" service type correct TMSI and CKSN received.
4. Authentication failed.
5. Authentication passed.
6. EMERGENCY SETUP with BC or without BC received.
7. The TCH channel is not through connected, fail.
8. The TCH channel is through connected.
9. dual rate / full rate preferred" case. setup a physical channel as full rate traffic channel and the preferred full rate channel is to be assigned.
10. dual rate / half rate preferred" case. setup a physical channel as half rate traffic channel and the preferred half rate channel is to be assigned.

Test Case Dynamic Behaviour

Test Case Name: TC_26_9_6_1_2

Group: GSM_L3_MS_v4220/StructureProc/

Purpose: 1) To verify that the MS in the "idle, no IMSI" state (no SIM inserted) when made to call the number 112, sends a CHANNEL REQUEST message with establishment cause "emergency call".

2) To verify that after assignment of a dedicated channel the first layer message sent by the MS on the assigned dedicated channel is a CM SERVICE REQUEST message specifying the correct CKSN and TMSI, with CM Service Type "emergency call establishment".

3) To verify that after receipt of a CM SERVICE ACCEPT message from the SS, the MS sends an EMERGENCY SETUP message.

4) To verify that subsequently, the SS having sent a CALL PROCEEDING message and then an ALERT message and having initiated the assignment procedure of an appropriate speech traffic channel, which, if the MS supports both TCH/FS and TCH/HS, is at the non-preferred rate, the MS performs correctly that assignment procedure.

5) To verify subsequent correct performance of a connect procedure.

6) To verify that subsequently the MS has through connected the TCH in both directions.

7) To verify that the call is cleared correctly.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(C_EmgCall, TSPX_EmgCallRate)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNA, TSPX_RANDA, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4	body	+local_continue			
		local_continue			
5		+InitCall(TCV_Service)			
6	L_0538	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_16)	(P)	2.
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(TSPX_TimadvA)))		
9	L_0539	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_06(TSPX_CKSNA))	(P)	3.
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		+Authentication(TCV_ch, TSPX_CKSNA, TSPX_RANDA)			
12	L_0540	[NOT TCV_Res]		(F)	4.

13		+PostMainLinkRel(TCV_ch)			
14	L_0541	[TCV_Res]		(P)	5.
15		+Ciphering_on(TCV_ch)			
16		+local_continue1			
		local_continue1			
17		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			6.
18		+local_nonpreferred_rate			
19		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
20		L!DL_DatRqAlert	AlertSnd(TCV_ch, Alerting_01(TCV_TI))		
21		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
22		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
23		+local_continue2			
		local_continue2			
24		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
25	L_0542	L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))	(P)	
26		(TCV_Res := OO_TCHThroConnCHK())			
27	L_0543	[NOT TCV_Res]		(F)	9.
28		+PostMainLinkRel(TCV_chTch)			
29	L_0544	[TCV_Res]		(P)	10.
30		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
31	L_0545	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	
32		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
33		+PostMainLinkRel(TCV_chTch)			
		local_nonpreferred_rate			
34		(TCV_Rchr := TCV_Setup_mo1.bcap1.rchr)			
35		[TCV_Rchr = '10'B]			7.
36		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd,TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
37		(TCV_ChRate := C_Full)			
38		[TCV_Rchr = '11'B]			8.
39		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
40		(TCV_ChRate := C_Half)			

Detailed Comments:

1. To setup a physical channels as BCCH CCCH and SDCCH4 combined channel.
2. CHANNEL REQUEST with "emergency call establishment" cause received.
3. CM SERVICE REQUEST with "emergency call establishment" service type received.
4. Authentication failed.

	5. Authentication passed. 6. EMERGENCY SETUP with a BC indicating "dual rate/half rate preferred" or "dual rate/full rate preferred" received. 7. dual rate / half rate preferred" case. setup a physical channel as full rate traffic channel and the non-preferred full rate channel is to be assigned. 8. dual rate / full rate preferred" case. setup a physical channel as half rate traffic channel and the non-preferred half rate channel is to be assigned. 9. The TCH channel is not through connected, fail. 10. The TCH channel is through connected.

Test Case Dynamic Behaviour

Test Case Name: TC_26_9_6_2_1

Group: GSM_L3_MS_v4220/StructureProc/

Purpose: 1) To verify that the MS in the "idle, no IMSI" state (no SIM inserted) when made to call the number 112, sends a CHANNEL REQUEST message with establishment cause "emergency call".

2) To verify that after assignment of a dedicated channel the first layer message sent by the MS on the assigned dedicated channel is a CM SERVICE REQUEST message in which the cipher key sequence number IE indicates "no key is available", the CM service type IE indicates "emergency call establishment", and the mobile identity IE specifies the IMEI of the MS.

3) To verify that after receipt of a CM SERVICE ACCEPT message from the SS, the MS sends an EMERGENCY SETUP message.

4) To verify that subsequently, the SS having sent a CALL PROCEEDING message and then an ALERT message and having initiated the assignment procedure of an appropriate speech traffic channel, which, if the MS supports both TCH/FS and TCH/HS, is at the preferred rate, the MS performs correctly that assignment procedure.

5) To verify subsequent correct performance of a connect procedure.

6) To verify that subsequently the MS has through connected the TCH in both directions.

7) To verify that the call is cleared correctly.

Default: OtherEventsFail

Comments: For this test case the SIM card shall be removed form the MS.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(C_EmgCall, TSPX_EmgCallRate)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Rej, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5	body	+InitCall(TCV_Service)			
6	L_0546	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_16)	(P)	2.
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
9	L_0547	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_07)	(P)	3.
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
12		+local_continue			

		local_continue			
13		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			4.
14		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
15		L!DL_DatRqAlert	AlertSnd(TCV_ch, Alerting_01(TCV_TI))		
16		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			5.
17		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
18		+local_continue1			
		local_continue1			
19		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
20	L_0548	L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))	(P)	
21		(TCV_Res := OO_TCHThroConnCHK())			
22	L_0549	[NOT TCV_Res]		(F)	6.
23		+PostMainLinkRel(TCV_chTch)			
24	L_0550	[TCV_Res]		(P)	7.
25		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
26	L_0551	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	
27		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
28		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
29	L_0552	L?DL_RelIn	DLRelInd_01	(P)	8.

Detailed Comments:

1. To setup a physical channel as BCCH CCCH and SDCCH4 combined channel.
2. CHANNEL REQUEST with "emergency call establishment" cause received.
3. CM SERVICE REQUEST with "emergency call establishment" service type received and the mobile identity IE specifies the IMEI of the MS, the classmark IE has the value specified in PIXIT.
4. EMERGENCY SETUP with BC indicating or without BC received.
5. To assign the traffic channel.
6. The TCH channel is not through connected, fail.
7. The TCH channel is through connected.
8. Main signalling link is released.

Test Case Dynamic Behaviour

Test Case Name: TC_26_9_6_2_2

Group: GSM_L3_MS_v4220/StructureProc/

Purpose: 1) To verify that the MS in the "idle, no IMSI" state (no SIM inserted) when made to call the number 112, sends a CHANNEL REQUEST message with establishment cause "emergency call".

2) To verify that after assignment of a dedicated channel the first layer message sent by the MS on the assigned dedicated channel is a CM SERVICE REQUEST message in which the cipher key sequence number IE indicates "no key is available", the CM service type IE indicates "emergency call establishment", and the mobile identity IE specifies the IMEI of the MS.

3) To verify that after receipt of a CM SERVICE REJECT message from the SS, the MS abandons the emergency call establishment.

Default: OtherEventsFail

Comments: For this test case the SIM card shall be removed form the MS.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(C_EmgCall, TSPX_EmgCallRate)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Rej, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4	body	+local_continue			
		local_continue			
5		+InitCall(TCV_Service)			
6	L_0553	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_16)	(P)	2.
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
9	L_0554	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_07)	(P)	3.
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		L!DL_DatRqCmsRej	CMSerRej(TCV_ch, CMServiceRej_02)		
12		START T_dly(5000)			4.
13		?TIMEOUT T_dly			
14		+PostMainLinkRel(TCV_ch)			
15		START T_dly(20000)			5.
16		?TIMEOUT T_dly			

Detailed Comments:

1. To setup a physical channel as BCCH CCCH and SDCCH4.
2. CHANNEL REQUEST with "emergency call" received.
3. CM SERVICE REQUEST with "emergency call establishment", IMEI, "no key available" and classmark received.
4. To check whether the MS sends any L 3 messages, the test case fails in the default tree if the MS does.
5. To check whether the MS initiates an RR connection establishment, the test case fails in the default tree if the MS does.

Test Group EGSMSignalling

Test Case Dynamic Behaviour					
Test Case Name:		TC_26_10_2_1			
Group:		GSM_L3_MS_v4220/EGSMSignalling/			
Purpose:		To verify that the MS reports appropriate results when the test system gives information about neighbouring cells			
Default:		OtherEventsFail			
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		START T_guard(480)			
2		+BasicServiceMT(TSPX_MTBscSvcA, TSPX_MTChRateA)			
3		+IdleUpdated(C_E_neighbourdefault, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_21, BcchFreqLst_Omit, BcchFreqLst_27, BcchFreqLst_Omit, C_noRestablishment, C_BCC_3, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcn_2, C_arfcnAd_2, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		+StartMultiCells_02(BcchFreqLst_21, BcchFreqLst_27, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1, C_LAC_1)			
5		(TCV_Cnt:=0)			
6		REPEAT ltree_loopForC UNTIL [TCV_Cnt =6]			
		ltree_loopForC			
7		+ltree_StartMultiCells			
8		+CCConfigTCH(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
9	body	+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubA, 0, 3, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			
10		(TCV_Null := OM_StartMsrReport(TCV_sacchTch))			
11		+ltree_receiveMsrRept			
12		+local_MsrReptCHK(1, 2)			
13		START T_dly1(960)			960 ms
14		+ltree_receiveMsrRept2			
15		+local_MsrReptCHK(1, 2)			
16		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
17		+PostMainLinkRel(TCV_chTch)			
18		+execution2			
		execution2			
19		+ltree_sysinfo5and5bis			1.
20		+PreEnterCCstateU10_r01(TimingAdv(30), TSPX_SDCCH4SubA, 0, 3, TSPX_CKSNDef, TSPX_RANDDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef, TCV_ChRate)			
21		(TCV_Null :=			

		OM_StartMsrReport(TCV_sacchTch))			
22		+ltree_receiveMsrRept			
23		+local_MsrReptCHK(1, 3)			
24		START T_dly1(960)			960 ms
25		+ltree_receiveMsrRept2			
26		+local_MsrReptCHK(1, 3)			
27		(TCV_Null := OM_StopMsrReport(TCV_sacchTch))			
28		+PostMainLinkRel(TCV_chTch)			
29		(TCV_Cnt:=(TCV_Cnt +1))			
		ltree_sysinfo5and5bis			
30		+Wait(C_T_Wait1)			
31		[TCV_Cnt = 1]			
32		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_e201))		
33		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch,SysInf5bis(BcchFreqLst_e207))		
34		[TCV_Cnt = 2]			
35		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_e202))		
36		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_e208))		
37		[TCV_Cnt = 3]			
38		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_e203))		
39		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_e209))		
40		[TCV_Cnt = 4]			
41		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_e204))		
42		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_e210))		
43		[TCV_Cnt = 5]			
44		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_e205))		
45		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_e211))		
46		[TCV_Cnt = 6]			
47		L!DL_UdatRqSysinfo5	SysInfo5(TCV_sacch, SysInf5(BcchFreqLst_e206))		
48		L!DL_UdatRqSysinfo5bis	SysInfo5bis(TCV_sacch, SysInf5bis(BcchFreqLst_e212))		
		local_MsrReptCHK(in1, in2:INTEGER)			
49		(TCV_Res := OC_MsrReptChk(TCV_MsrRes, in1))			
50	L_0555	[NOT TCV_Res]		(F)	
51	L_0556	[TCV_Res]		(P)	
		ltree_receiveMsrRept			

52		[TCV_Cnt = 1]			
53		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_03e(MsrResult_03e1))		
54		[TCV_Cnt = 2]			
55		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_03e(MsrResult_03e2))		
56		[TCV_Cnt = 3]			
57		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_03e(MsrResult_03e3))		
58		[TCV_Cnt = 4]			
59		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_03e(MsrResult_03e4))		
60		[TCV_Cnt = 5]			
61		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_03e(MsrResult_03e5))		
62		[TCV_Cnt = 6]			
63		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr)	MsrRept(MsrReport_04e)		
		ltree_receiveMsrRept2			
64		[TCV_Cnt = 1]			
65	L_0557	?TIMEOUT T_dly1		(F)	1.
66		+PostMainLinkRel(TCV_chTch)			
67		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr) CANCEL T_dly1	MsrRept(MsrReport_03e(MsrResult_03e1))		
68		[TCV_Cnt = 2]			
69	L_0558	?TIMEOUT T_dly1		(F)	1.
70		+PostMainLinkRel(TCV_chTch)			
71		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr) CANCEL T_dly1	MsrRept(MsrReport_03e(MsrResult_03e2))		
72		[TCV_Cnt = 3]			
73	L_0559	?TIMEOUT T_dly1		(F)	1.
74		+PostMainLinkRel(TCV_chTch)			
75		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr) CANCEL T_dly1	MsrRept(MsrReport_03e(MsrResult_03e3))		
76		[TCV_Cnt = 4]			
77	L_0560	?TIMEOUT T_dly1		(F)	1.
78		+PostMainLinkRel(TCV_chTch)			
79		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr) CANCEL T_dly1	MsrRept(MsrReport_03e(MsrResult_03e4))		
80		[TCV_Cnt = 5]			
81	L_0561	?TIMEOUT T_dly1		(F)	1.
82		+PostMainLinkRel(TCV_chTch)			
83		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr) CANCEL T_dly1	MsrRept(MsrReport_03e(MsrResult_03e5))		
84		[TCV_Cnt = 6]			
85	L_0562	?TIMEOUT T_dly1		(F)	1.
86		+PostMainLinkRel(TCV_chTch)			
87		L?DL_UdatInMsrRpt (TCV_MsrRes := DL_UdatInMsrRpt.msg.msrr) CANCEL T_dly1	MsrRept(MsrReport_04e)		
		ltree_StartMultiCells			
88		[TCV_Cnt = 0]			

89		[TCV_Cnt = 1]			
90		+StartMultiCells_02e(BcchFreqLst_e201, BcchFreqLst_27, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), 0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, C_LAC_2, C_LAC_3, C_LAC_4, C_LAC_5, C_LAC_6, C_LAC_7, C_LAC_8)			
91		[TCV_Cnt = 2]			
92		+StartMultiCells_02e(BcchFreqLst_e202, BcchFreqLst_27, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), 0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, C_LAC_2, C_LAC_3, C_LAC_4, C_LAC_5, C_LAC_6, C_LAC_7, C_LAC_8)			
93		[TCV_Cnt = 3]			
94		+StartMultiCells_02e(BcchFreqLst_e203, BcchFreqLst_27, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), 0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, C_LAC_2, C_LAC_3, C_LAC_4, C_LAC_5, C_LAC_6, C_LAC_7, C_LAC_8)			
95		[TCV_Cnt = 4]			
96		+StartMultiCells_02e(BcchFreqLst_e204, BcchFreqLst_27, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), 0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, C_LAC_2, C_LAC_3, C_LAC_4, C_LAC_5, C_LAC_6, C_LAC_7, C_LAC_8)			
97		[TCV_Cnt = 5]			
98		+StartMultiCells_02e(BcchFreqLst_e205, BcchFreqLst_27, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), 0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, C_LAC_2, C_LAC_3, C_LAC_4, C_LAC_5, C_LAC_6, C_LAC_7, C_LAC_8)			
99		[TCV_Cnt = 6]			
100		+StartMultiCells_02e(BcchFreqLst_e206, BcchFreqLst_27, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), 0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, C_LAC_2, C_LAC_3, C_LAC_4, C_LAC_5, C_LAC_6, C_LAC_7, C_LAC_8)			

Detailed Comments: 1. The interval between 2 successive layer 2 frames containing MEASUREMENT REPORT exceeds
one layer 2 frame, fail.

Test Case Dynamic Behaviour

Test Case Name: TC_26_10_2_2
Group: GSM_L3_MS_v4220/EGSMSignalling/
Purpose: To verify that the MS can correctly set up a dedicated control channel when E-GSM frequencies are used.
Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1	pre	START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_19, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_46, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcn_1015, C_arfcn_1015, TSPX_IMSI, C_Test_egsm, C_NCCP_2)			
3		(TCV_ia_ts:= TSPX_TmSltNotZero, TCV_Cnt:= 1)			
4	body	REPEAT localtree_body UNTIL [TCV_Cnt =3]			
		localtree_body			
5		+localtree_config			
6		+Wait(C_T_Wait1)			
7		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
8		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
9		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
10		+ltree_send_immass			
11	L_0563	L?DL_EstInPgRes	PgRes(TCV_ch, PagingRes_01)	(P)	
12		ACTIVATE(OtherEventsFail_01)			Restore Normal default
13		+ChanRel(TCV_ch)			
14		(TCV_Null := OM_FreeResource(TCV_ch), TCV_Cnt := TCV_Cnt + 1)			
		localtree_config			
15		[TCV_Cnt = 1]			
16		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubA, C_Ass, TSPX_TmSltNotZero, TSPX_TscDef, ChMod_sign, FreqSDCCH8_e1, FreqSDCCH8_e1, TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3)			for 1)
17		+SysInfo_5bisSending(TCV_sacch8, TCV_sysinfo5bis)			
18		[TCV_Cnt =2]			
19		+SDCCH8_A_2_nociph(TSPX_SDCCH8SubB, C_Ass, TSPX_TmSltNotZero, TSPX_TscDef, ChMod_r01, FreqSDCCH8_e2, FreqSDCCH8_e2, TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
20		+SysInfo_5bisSending(TCV_sacchTch, TCV_sysinfo5bis)			
21		(TCV_ch := TCV_chTch)			
		ltree_send_immass			
22		[TCV_Cnt = 1]			
23		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_E_01(TCV_Rr, TCV_Fn, TCV_ia_ts,		1)

			TSPX_TscDef, TCV_chdescr_arfcn, TimingAdv(0)))		
24		[TCV_Cnt = 2]			
25		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_E_02(TCV_Rr, TCV_Fn, TCV_ia_ts, TSPX_TscDef, TimingAdv(0)))		2)

Detailed Comments: 1) Immediate Assignment with single RF on ARFCN=1015
2) Immediate Assignment with frequency hopping

Test Case Dynamic Behaviour

Test Case Name: TC_26_10_2_3
Group: GSM_L3_MS_v4220/EGSMSignalling/
Purpose: 1. To verify that upon receipt of an ASSIGNMENT COMMAND, the MS switches to the channel defined in the ASSIGNMENT COMMAND, establishes the link and sends an ASSIGNMENT COMPLETE message.

2. To verify that an MS, having received an ASSIGNMENT COMMAND, is able in case of frequency hopping to decode the mobile allocation and frequency list correctly and applies the specified frequencies.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1	pre	START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_22, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_46, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_egsm, C_NCCP_2)			
3		(TCV_ia_ts:= TSPX_TmSltA, TCV_Cnt := 1, TCV_chdescr_arfcn := C_arfcnC)			
4	body	REPEAT localtree_body UNTIL [TCV_Cnt =3]			
		localtree_body			
5		+SDCCH8_A_2_nociph(TSPX_SDCCH8SubDef, C_Ass, TCV_ia_ts, TSPX_TscDef, ChMod_sign, FreqBCCH(TCV_chdescr_arfcn), FreqBCCH(TCV_chdescr_arfcn), TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
6		+SysInfo_5bisSending(TCV_sacchTch, TCV_sysinfo5bis)			
7		(TCV_ch := TCV_chTch)			
8		+Wait(C_T_Wait1)			
9		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
10		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
11		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
12		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_27(TCV_Rr, TCV_Fn, TCV_ia_ts, TCV_chdescr_arfcn, TimingAdv(0)))		
13		L?DL_EstInPgRes	PgRes(TCV_ch, PagingRes_01)		
14		ACTIVATE(OtherEventsFail)			Restore Normal default
15		+ltree_chtype			
16		+ltree_send_ch_assign			
17		+ChanRel(TCV_chTch)			
18		(TCV_Null := OM_FreeResource(TCV_chTch), TCV_Cnt := (TCV_Cnt + 1))			Loop for K
19		+Wait(C_T_Wait1)			
		ltree_chtype			
20		[TSPC_SvcOnTCH]			
21		(TCV_chtype := '00001'B)			for TCH

22		[NOT TSPC_SvcOnTCH]			
23		(TCV_chtype := INT_TO_BIT((8 + BIT_TO_INT(TSPX_SDCCH8SubA)), 5))			for SDCCH8
		ltree_send_ch_assign			
24		[TCV_Cnt = 1]			
25		(TCV_asscmd_ts := INT_TO_BIT(((BIT_TO_INT(TSPX_TmSltA) + 1) MOD 8), 3), TCV_Cnt1 := 1)			
26		REPEAT ltree_assign1 UNTIL [TCV_Cnt1 = 7]			
27		[TCV_Cnt = 2]			
28		(TCV_Cnt1 := 1)			
29		REPEAT ltree_assign2 UNTIL [TCV_Cnt1 = 7]			
		ltree_assign1			
30		+ltree_assign1_setup			
31		+ltree_chnassign1			
32		(TCV_AssCmd := AsgnCmd_22_Ae1(TCV_asscmd_ts, TCV_chtype, TCV_flist, TCV_flistl, TCV_n))			
33		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			1)
34		START T_dly(500)			6)
35		?TIMEOUT T_dly			
36		(TCV_Cnt1 := (TCV_Cnt1 + 1), TCV_Null := OM_FreeResource(TCV_ch), TCV_ch := TCV_chTch)			Loop for C
		ltree_assign2			
37		+ltree_assign2_setup			
38		+ltree_asscmdsending			
39		START T_dly(500)			6)
40		?TIMEOUT T_dly			
41		(TCV_Cnt1 := (TCV_Cnt1 + 1), TCV_Null := OM_FreeResource(TCV_ch), TCV_ch := TCV_chTch)			
		ltree_asscmdsending			
42		[TCV_Cnt1 = 4]			
43		+ltree_chnassign1			3)
44		(TCV_AssCmd := AsgnCmd_22_Ae2(TCV_asscmd_ts, TCV_chtype, TCV_cchdescr, TCV_mae1, TCV_n))			
45		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			2)
46		[TCV_Cnt1 <> 4]			
47		+ltree_chnassign2			4)
48		(TCV_AssCmd := AsgnCmd_22_Ae3(TCV_asscmd_ts, TCV_chtype, TCV_cchdescr, TCV_mae1, TCV_mae2, TCV_n))			
49		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			2)
		ltree_chnassign1			
50		[TSPC_SvcOnTCH]			1)
51		[(TCV_Cnt1 MOD 2) > 0]			
52		+FullRateCh_A_1_nociph(C_Ass, TCV_asscmd_ts, TCV_tsc, ChMod_speech, FreqTCH_ef1(TCV_mae1, TCV_flist, TCV_flistl, TCV_n), FreqTCH_ef1(TCV_mae1, TCV_flist, TCV_flistl, TCV_n), TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
53		+SysInfo_5bisSending(TCV_sacchTch, TCV_sysinfo5bis)			
54		[(TCV_Cnt1 MOD 2) = 0]			
55		+FullRateCh_A_2_nociph(C_Ass, TCV_asscmd_ts, TCV_tsc, ChMod_speech, FreqTCH_ef1(TCV_mae1, TCV_flist, TCV_flistl, TCV_n), FreqTCH_ef1(TCV_mae1, TCV_flist, TCV_flistl, TCV_n), TimingAdv(0), C_BABR_0,			

		C_cch_1Comb, C_BPM_3)				
56		+SysInfo_5bisSending(TCV_sacchTch, TCV_sysinfo5bis)				
57		(TCV_chTch := TCV_chTch1)				
58		[(NOT TSPC_SvcOnTCH)]				2)
59		[(TCV_Cnt1 MOD 2) > 0]				
60		(TCV_chTch := TCV_ch)				
61		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubA, C_Ass, TCV_asscmd_ts, TCV_tsc, ChMod_sign, FreqTCH_ef1(TCV_mae1,TCV_flist, TCV_flistl, TCV_n), FreqTCH_ef1(TCV_mae1,TCV_flist, TCV_flistl, TCV_n), TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3)				
62		+SysInfo_5bisSending(TCV_sacchTch, TCV_sysinfo5bis)				
63		(TCV_chTch1 := TCV_chTch, TCV_chTch := TCV_ch, TCV_ch := TCV_chTch1)				5)
64		[(TCV_Cnt1 MOD 2) = 0]				
65		+SDCCH8_A_2_nociph(TSPX_SDCCH8SubA, C_Ass, TCV_asscmd_ts, TCV_tsc, ChMod_sign, FreqTCH_ef1(TCV_mae1,TCV_flist, TCV_flistl, TCV_n), FreqTCH_ef1(TCV_mae1,TCV_flist, TCV_flistl, TCV_n), TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3)				
66		+SysInfo_5bisSending(TCV_sacchTch, TCV_sysinfo5bis)				
		ltree_chnassign2				
67		[TSPC_SvcOnTCH]				1)
68		[(TCV_Cnt1 MOD 2) > 0]				
69		+FullRateCh_A_1_nociph(C_Ass, TCV_asscmd_ts, TCV_tsc, ChMod_speech, FreqTCH_ef2(TCV_mae2, TCV_mae1, TCV_flist, TCV_flistl, TCV_n), FreqTCH_ef2(TCV_mae2, TCV_mae1, TCV_flist, TCV_flistl, TCV_n), TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3)				
70		+SysInfo_5bisSending(TCV_sacchTch, TCV_sysinfo5bis)				
71		[(TCV_Cnt1 MOD 2) = 0]				
72		+FullRateCh_A_2_nociph(C_Ass, TCV_asscmd_ts, TCV_tsc, ChMod_speech, FreqTCH_ef2(TCV_mae2, TCV_mae1, TCV_flist, TCV_flistl, TCV_n), FreqTCH_ef2(TCV_mae2, TCV_mae1, TCV_flist, TCV_flistl, TCV_n), TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3)				
73		+SysInfo_5bisSending(TCV_sacchTch, TCV_sysinfo5bis)				
74		(TCV_chTch := TCV_chTch1)				
75		[(NOT TSPC_SvcOnTCH)]				2)
76		[(TCV_Cnt1 MOD 2) > 0]				
77		(TCV_chTch := TCV_ch)				
78		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubA, C_Ass, TCV_asscmd_ts, TCV_tsc, ChMod_sign, FreqTCH_ef2(TCV_mae2, TCV_mae1, TCV_flist, TCV_flistl, TCV_n), FreqTCH_ef2(TCV_mae2, TCV_mae1, TCV_flist, TCV_flistl, TCV_n), TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3)				
79		+SysInfo_5bisSending(TCV_sacchTch, TCV_sysinfo5bis)				
80		(TCV_chTch1 := TCV_chTch, TCV_chTch := TCV_ch, TCV_ch := TCV_chTch1)				
81		[(TCV_Cnt1 MOD 2) = 0]				
82		+SDCCH8_A_2_nociph(TSPX_SDCCH8SubA, C_Ass, TCV_asscmd_ts, TCV_tsc, ChMod_sign, FreqTCH_ef2(TCV_mae2, TCV_mae1, TCV_flist, TCV_flistl, TCV_n), FreqTCH_ef2(TCV_mae2, TCV_mae1, TCV_flist, TCV_flistl, TCV_n), TimingAdv(0), C_BABR_0,				

		C_cch_1nonComb, C_BPM_3)			
83		+SysInfo_5bisSending(TCV_sacchTch, TCV_sysinfo5bis)			
		ltree_assign1_setup			
84		[TCV_Cnt1 = 1]			
85		(TCV_flist := C_flist_e_401, TCV_flistl:='05'O, TCV_mae1:='00000111'B, TCV_n := 3)			
86		[TCV_Cnt1 = 2]			
87		(TCV_flist := C_flist_e_402, TCV_flistl:='06'O, TCV_mae1:='00011111'B, TCV_n := 5)			
88		[TCV_Cnt1 = 3]			
89		(TCV_flist := C_flist_e_403, TCV_flistl:='06'O, TCV_mae1:= '00001111'B, TCV_n := 4)			
90		[TCV_Cnt1 = 4]			
91		(TCV_flist := C_flist_e_404, TCV_flistl:='06'O, TCV_mae1:='00001111'B, TCV_n := 4)			
92		[TCV_Cnt1 = 5]			
93		(TCV_flist := C_flist_e_405, TCV_flistl:='07'O, TCV_mae1:='00111111'B, TCV_n := 6)			
94		[TCV_Cnt1 = 6]			
95		(TCV_flist := C_flist_e_406, TCV_flistl:='10'O, TCV_mae1:='00000111'B, TCV_n := 3)			
		ltree_assign2_setup			
96		[TCV_Cnt1 = 1]			
97		(TCV_cchdescr := C_cchd_e_407, TCV_mae1:='00011100'B, TCV_mae2:='00000000'B, TCV_flist:=C_flist_e_407, TCV_flistl:='10'O, TCV_n := 3)			
98		[TCV_Cnt1 = 2]			
99		(TCV_cchdescr := C_cchd_e_408, TCV_mae1:='00000000'B, TCV_mae2:='11111000'B, TCV_flist:= C_flist_e_408, TCV_flistl:='0D'O, TCV_n := 5)			
100		[TCV_Cnt1 = 3]			
101		(TCV_cchdescr := C_cchd_e_409, TCV_mae1:='00000000'B, TCV_mae2:='00001111'B, TCV_flist:= C_flist_e_409, TCV_flistl:='10'O, TCV_n := 4)			
102		[TCV_Cnt1 = 4]			
103		(TCV_cchdescr := C_cchd_e_415, TCV_mae1:='11100011'B, TCV_flist:= C_flist_e_415, TCV_flistl:='09'O, TCV_n := 5)			
104		[TCV_Cnt1 = 5]			
105		(TCV_cchdescr := C_cchd_e_411, TCV_mae1:='00000001'B, TCV_mae2:='11110000'B, TCV_flist:= C_flist_e_411, TCV_flistl:='07'O, TCV_n := 5)			
106		[TCV_Cnt1 = 6]			
107		(TCV_cchdescr := C_cchd_e_412, TCV_mae1:='00000000'B, TCV_mae2:='00000111'B, TCV_flist:= C_flist_e_412, TCV_flistl:='10'O, TCV_n := 3)			

Detailed Comments:

1) Assignment Command without Cell Channel Description IE
2) Assignment Command with Cell Channel Description IE
3) Length of mobile alocation contents is 1 octet
4) PDU contains Mobile Allocation with 2 octets
5) To ensure the correct channel id is in the correct test case variable
6) Ensure all communication is complete before freeing the resource

Test Case Dynamic Behaviour

Test Case Name: TC_26_10_2_4_1

Group: GSM_L3_MS_v4220/EGSMSignalling/

Purpose: To check that the MS correctly performs a non-synchronized handover, from a non hopping primary band SDCCH to a hopping TCH or SDCCH using E-GSM frequencies, whatever the coding used for the hopping sequence description and that it activates the new channel correctly.

This tested in the following case:

E-GSM signalling/ Handover / successful / call under establishment / non-synchronized /

- from SDCCH/8 to TCH/F if the MS supports a TCH
- from SDCCH/8 to SDCCH/8 if not

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcD, TSPX_MOChRateD)			
3		(TCV_ia_ts:= TSPX_TmSltNotZero, TCV_ts:= TSPX_TmSltNotZero, TCV_Horf:= TSPX_HoRefA, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5), TCV_flist := C_flist_e_416, TCV_flistl:= '10'O, TCV_mae1 := '00000101'B, TCV_mae2 := '00000001'B)			1.
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, TSPX_TscDef, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_23A, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_Omit, BcchFreqLst_46, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_egsm, C_NCCP_2)			
5		+local_continue			
		local_continue			
6		(TCV_sysinfo5bis_A := TCV_sysinfo5bis)			
7		+StartCellB_egsm(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TSPX_TmSltNotZero, TCV_tsc, TimingAdv(20), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_1, CellChDes_23B, BcchFreqLst_52, BcchFreqLst_47, C_noRestablishment, C_BCC, C_NCC)			
8		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TSPX_TmSltNotZero, TCV_tsc, ChMod_sign, FreqTCH_ef2(TCV_mae1, TCV_mae2, TCV_flist,TCV_flistl, TCV_n), FreqTCH_ef2(TCV_mae1, TCV_mae2, TCV_flist,TCV_flistl, TCV_n), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			TCV_ch
9		+SysInfo_5bisSending(TCV_sacch8, TCV_sysinfo5bis_A)			
10	body	(TCV_Cnt2 :=1, TCV_ch1 := TCV_ch)			
11		+ltree_chtype			
12		REPEAT local_body UNTIL [TCV_Cnt2 =4]			Loop of k = 1 to 3
		local_body			
13		(TCV_Cnt1 :=1)			
14		REPEAT ltree_execloopC UNTIL [TCV_Cnt1=7]			Loop of c = 1 to 6
15		(TCV_Cnt2 := (TCV_Cnt2 +1))			

		ltree_execloopC			
16		[(TCV_Cnt2 <>1) OR(TCV_Cnt1<>6)]			
17		+ltree_chconfig			
18		+Est_MO_Call_init(C_CHSDCCH8_FH, MoblAllc_281e2, MoblAllc_281e2, TimingAdv(30), TSPX_TscDef, 1, 1)			
19		+ltree_hocmd			
20		(TCV_chTch1 := TCV_ch)			
21		(TCV_ch := TCV_chTch)			
22		+RR_hocomp1(TCV_n, TCV_chTch1)			
23		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
24		+SendSeqNo_chk			
25		+ChanRel(TCV_ch)			
26		(TCV_Cnt1 := (TCV_Cnt1+1), TCV_ch := TCV_ch1, TCV_Null :=OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			SDCCH8_A_1 back to no ciphering mode
27		[(TCV_Cnt2 = 1) AND (TCV_Cnt1 = 6)]			no test for k=1 and c=6
28		(TCV_Cnt1 := (TCV_Cnt1+1), TCV_ch := TCV_ch1)			
		ltree_chtype			
29		[TSPC_Serv_TS11 OR TSPC_Serv_TS12]			
30		(TCV_chtype := '00001'B, TCV_n := 500, TCV_Cntref:= TSPX_NoOfHoAccA)			TCH
31		[(NOT TSPC_Serv_TS11) AND(NOT TSPC_Serv_TS12)]			
32		(TCV_chtype := INT_TO_BIT((8 + BIT_TO_INT(TSPX_SDCCH8SubDef)),5), TCV_n :=750, TCV_Cntref:= TSPX_NoOfHoAccl)			SDCCH8
		ltree_hocmd			
33		[TCV_Cnt2 = 1]			for k = 1
34		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_chtype, TCV_ts, TSPX_TscDef, C_NCC_1, C_BCC_3, C_BCCHcarrierB_ho, TCV_n, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_short(TCV_flist), Frql_omit, CellChDes_Omit, TCV_ChMod, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
35		[TCV_Cnt2 = 2]			for k = 2
36		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_chtype, TCV_ts, TSPX_TscDef, C_NCC_1, C_BCC_3, C_BCCHcarrierB_ho, TCV_n, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql(TCV_flist, TCV_flistl), CellChDes_Omit, TCV_ChMod, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		

37		[TCV_Cnt2 = 3]			for k = 3
38		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_chtype, TCV_ts, TSPX_TscDef, C_NCC_1, C_BCC_3, C_BCCHcarrierB_ho, TCV_n, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_20_Be(TCV_cchdescr), TCV_ChMod, Freqchseq_omit, MoblAllc_20_Be2iei(TCV_mae1, TCV_mae2), RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
		ltree_chconfig			
39		+local_setHoppingPara			
40		+local_configChOfCellB			
		local_setHoppingPara			
41		[TCV_Cnt2 = 1]			
42		[TCV_Cnt1 = 1]			
43		(TCV_flist := C_flist_e_401, TCV_flistl:='05'O, TCV_mae1:='00000111'B, TCV_n := 3)			k =1 and c =1
44		[TCV_Cnt1 = 2]			
45		(TCV_flist := C_flist_e_402, TCV_flistl:='06'O, TCV_mae1:='00011111'B, TCV_n := 5)			k =1 and c =2
46		[TCV_Cnt1 = 3]			
47		(TCV_flist := C_flist_e_403, TCV_flistl:='06'O, TCV_mae1:= '00001111'B, TCV_n := 4)			k =1 and c =3
48		[TCV_Cnt1 = 4]			
49		(TCV_flist := C_flist_e_404, TCV_flistl:='06'O, TCV_mae1:='00001111'B, TCV_n := 4)			k =1 and c =4
50		[TCV_Cnt1 = 5]			
51		(TCV_flist := C_flist_e_405, TCV_flistl:='07'O, TCV_mae1:='00111111'B, TCV_n := 6)			k =1 and c =5
52		[TCV_Cnt2 = 2]			
53		[TCV_Cnt1 = 1]			
54		(TCV_flist := C_flist_e_401, TCV_flistl:='05'O, TCV_mae1:='00000111'B, TCV_n := 3)			k =2 and c = 1
55		[TCV_Cnt1 = 2]			
56		(TCV_flist := C_flist_e_402, TCV_flistl:='06'O, TCV_mae1:='00011111'B, TCV_n := 5)			k =2 and c = 2
57		[TCV_Cnt1 = 3]			
58		(TCV_flist := C_flist_e_403, TCV_flistl:='06'O, TCV_mae1:= '00001111'B, TCV_n := 4)			k =2 and c = 3
59		[TCV_Cnt1 = 4]			
60		(TCV_flist := C_flist_e_404, TCV_flistl:='06'O, TCV_mae1:='00001111'B, TCV_n := 4)			k =2 and c = 4
61		[TCV_Cnt1 = 5]			
62		(TCV_flist := C_flist_e_405, TCV_flistl:='07'O, TCV_mae1:='00111111'B, TCV_n := 6)			k =2 and c = 5
63		[TCV_Cnt1 = 6]			
64		(TCV_flist := C_flist_e_406, TCV_flistl:='10'O, TCV_mae1:='00000111'B, TCV_n := 3)			k =2 and c = 6
65		[TCV_Cnt2 = 3]			
66		[TCV_Cnt1 = 1]			
67		(TCV_cchdescr := C_cchd_e_407, TCV_mae2:='00011100'B, TCV_mae1:='00000000'B, TCV_flist:=C_flist_e_407, TCV_flistl:='05'O, TCV_n := 3)			k =3 and c = 1
68		[TCV_Cnt1 = 2]			

69		(TCV_cchdescr := C_cchd_e_408, TCV_mae2:='00000000'B, TCV_mae1:='11111000'B, TCV_flist:= C_flist_e_408, TCV_flistl:='0D'O, TCV_n := 5)			k =3 and c = 2
70		[TCV_Cnt1 = 3]			
71		(TCV_cchdescr := C_cchd_e_409, TCV_mae2:='00000000'B, TCV_mae1:='00001111'B, TCV_flist:= C_flist_e_409, TCV_flistl:='10'O, TCV_n := 3)			k =3 and c = 3
72		[TCV_Cnt1 = 4]			
73		(TCV_cchdescr := C_cchd_e_410, TCV_mae2:='00000011'B, TCV_mae1:='00000011'B, TCV_flist:= C_flist_e_410, TCV_flistl:='0A'O, TCV_n := 4)			k =3 and c = 4
74		[TCV_Cnt1 = 5]			
75		(TCV_cchdescr := C_cchd_e_411, TCV_mae1:='00000001'B, TCV_mae2:='11110000'B, TCV_flist:= C_flist_e_411, TCV_flistl:='07'O, TCV_n := 4)			k =3 and c = 5
76		[TCV_Cnt1 = 6]			
77		(TCV_cchdescr := C_cchd_e_412, TCV_mae1:='00000000'B, TCV_mae2:='00000111'B, TCV_flist:= C_flist_e_412, TCV_flistl:='10'O, TCV_n := 3)			k =3 and c = 6
		local_configChOfCellB			
78		(TCV_cellid:=C_CellB)			
79		[TSPC_Serv_TS11 OR TSPC_Serv_TS12]			2.
80		(TCV_ChMod := ChMod_speech)			
81		[TCV_Cnt2 <> 3]			
82		+FullRateCh_B_1(C_Asynho, TCV_ts, TSPX_TscDef, TCV_ChMod, FreqTCH_ef1(TCV_mae1, TCV_flist, TCV_flistl, TCV_n), FreqTCH_ef1(TCV_mae1, TCV_flist, TCV_flistl, TCV_n), TimingAdv(20), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
83		[TCV_Cnt2 = 3]			
84		+FullRateCh_B_1(C_Asynho, TCV_ts, TSPX_TscDef, TCV_ChMod, FreqTCH_ef2(TCV_mae1, TCV_mae2, TCV_flist, TCV_flistl, TCV_n), FreqTCH_ef2(TCV_mae1, TCV_mae2, TCV_flist, TCV_flistl, TCV_n), TimingAdv(20), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
85		[(NOT TSPC_Serv_TS11) AND(NOT TSPC_Serv_TS12)]			3.
86		(TCV_ChMod := ChMod_sign)			
87		[TCV_Cnt2 <> 3]			
88		+SDCCH8_B_1(TSPX_SDCCH8SubDef, C_Asynho, TCV_ts, TSPX_TscDef, TCV_ChMod, FreqTCH_ef1(TCV_mae1, TCV_flist, TCV_flistl, TCV_n), FreqTCH_ef1(TCV_mae1, TCV_flist, TCV_flistl, TCV_n), TimingAdv(20), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
89		[TCV_Cnt2 = 3]			
90		+SDCCH8_B_1(TSPX_SDCCH8SubDef, C_Asynho, TCV_ts, TSPX_TscDef, TCV_ChMod, FreqTCH_ef2(TCV_mae1, TCV_mae2, TCV_flist, TCV_flistl, TCV_n), FreqTCH_ef2(TCV_mae1, TCV_mae2, TCV_flist, TCV_flistl, TCV_n), TimingAdv(20), C_BABR_0, C_cch_1nonComb, C_BPM_3)			

Detailed Comments:

1. Values TCV_flist , TCV_mae1 and TCV_mae2 are initialized for hopping frequencies : 20, 66, 78 in cell A.
2. for HO from SDCCH8 no FH in cell A to TCH/F FH in cell B
3. for HO from SDCCH8 no FH in cell A to SDCCH8 FH in cell B

Test Case Dynamic Behaviour

Test Case Name: TC_26_10_2_4_2

Group: GSM_L3_MS_v4220/EGSMSignalling/

Purpose: To check that the MS correctly returns to the old channel in the case of an handover failure caused by a layer 1 failure on the target cell, even if the origin is in the P-GSM band and the targetin the E-GSM band.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts:= TSPX_TmSltNotZero, TCV_ts:= TSPX_TmSltNotZero1, TCV_Cntref:= TSPX_NoOfHoAccA, TCV_Horf:= TSPX_HoRefA, TCV_Pwrlvl_ho:= '01100'B)			
3		+BasicServiceMT(TSPX_MTBscSvcD, TSPX_MTChRateD)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubA, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB(C_E_neighbourdefault, C_arfcnB, C_arfcnBd, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			
6		+FullRateCh_A_1(C_Ass, TCV_asscmd_ts, TCV_tsc, ChMod_speech, FreqTCH_omit, FreqTCH_omit, TSPX_TCHcarrierA_ho, TSPX_TCHcarrierA_hod, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
7		+ltree_u10			
8		+ltree_body			
		ltree_body			
9		(TCV_flist := C_flist_e_401, TCV_flistl := '05'O, TCV_mae1:= '00000111'B, TCV_n := 3, TCV_ch := TCV_chTch, TCV_sacch := TCV_sacchTch)			
10		+FullRateCh_B_1(C_Rcv, TCV_ts, TCV_tsc, ChMod_speech, FreqTCH_ef1(TCV_mae1, TCV_flist, TCV_flistl, TCV_n), FreqTCH_ef1(TCV_mae1, TCV_flist, TCV_flistl, TCV_n), TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
11		(TCV_L1Head0 := OM_GetL1Hd(TCV_sacch))			
12		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, HandOverCmd_fh(TCV_chtype, TCV_ts, TSPX_TscDef, C_NCC_1, C_BCC_3, C_BCCHcarrierB_ho, TCV_n, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_20_egsm(TCV_flist, TCV_flistl), Frql_omit, CellChDes_Omit,		

			TCV_ChMod, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))		
13		START T_dly(3000)			
14		(TCV_Cnt:=0)			
15		REPEAT ltree_hoacc UNTIL [TCV_Cnt = TCV_Cntref]			
16		L?DL_DatInHofl	HndOvFlRcv(TCV_ch, HandOvFail_02)		
17		(TCV_L1Head := OM_GetL1Hd(TCV_sacch))			
18	L_0564	[TCV_L1Head.mspwrlvl = TCV_L1Head0.mspwrlvl]		(P)	
19		+ChanRel_end(TCV_ch)			
20		[TCV_L1Head.mspwrlvl <> TCV_L1Head0.mspwrlvl]			
21		+ChanRel_end(TCV_ch)			
22	L_0565	?TIMEOUT T_dly		(F)	
23		+ChanRel(TCV_chTch)			
		ltree_hoacc			
24		L?DL_RacInHoacc	HndOvAccRcv(TCV_chTch, HandOverAcc_02(TCV_Horf))		
25		(TCV_Cnt := TCV_Cnt + 1)			
		ltree_u10			
26		+RRmtcallprepare(TimingAdv(0), TSPX_RANDDef)			
27		(TCV_Setup_mt.sig := Signal_01)			
28		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		
29		L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))		1.
30		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
31		+localtree			
32		L?DL_DatInAlert	AlertRcv(AlertingInd_01(TI_01))		
33		(TCV_Null := OO_HookOff())			
34		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
35		+localtree			
		localtree			
36		(TCV_AssCmd := AsgnCmd_nfh('00001'B, TCV_asscmd_ts, TSPX_TscDef, 10, C_arfcnEgsm_asscmd, CellChDes_Omit, ChMod_omit, StartingTm_omit, CphMod_omit))			
37		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			2.
38		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		

Detailed Comments:

1. If the MS supports the bearer capabilities, which are give in Setup message, it has to accept them. However, it may still send a bearer capability IE in the Call Confirm message.
2. TCH/F or SDCCH/4 with non hopping in selected cell.
   Power level = 10

Test Case Dynamic Behaviour

Test Case Name: TC_26_10_2_5
Group: GSM_L3_MS_v4220/EGSMSignalling/
Purpose: 1) To verify that the MS, after receiving a FREQUENCY REDEFINITION message, starts using the new frequencies and hopping sequence when some E-GSM freuencies are used.

2) To check that the last received Cell Channel Description information element is used to decode the Mobile Allocation IE received in the FREQUENCY REDEFINITION message.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1	pre	START T_guard(300)			
2		(TCV_ia_ts:= TSPX_TmSltNotZero)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_22, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_46, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_egsm, C_NCCP_2)			
4		+local_ChConfig			
5		(TCV_Cnt1 := 1)			
6	body	REPEAT localtree_body UNTIL [TCV_Cnt1 = 7]			
		localtree_body			
7		+Wait(C_T_Wait1)			
8		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
9		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
10		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
11		+ltree_send_immass			
12		L?DL_EstInPgRes	PgRes(TCV_ch, PagingRes_01)		
13		ACTIVATE(OtherEventsFail)			Restore Normal default
14		+ltree_setFreqRedefPara			
15		+ltree_LowerTestorManage			
16		+ltree_sendFreqRedef			
17		+ltree_checkchnafterfreqred			
18		+ChanRel(TCV_ch)			
19		(TCV_Cnt1 := (TCV_Cnt1 +1))			
		ltree_checkchnafterfreqred			
20		[(TCV_Cnt1 = 4) OR (TCV_Cnt1 = 6)]			Ma1)
21		(TCV_Res := OM_FHCHK(TCV_ch, CellChDes_20_Be(TCV_cchdescr), MoblAllc_20_Be1(TCV_mae1), TCV_chd1, 100, TCV_Fn))			
22	L_0566	[NOT TCV_Res]		(F)	
23	L_0567	[TCV_Res]		(P)	
24		[(TCV_Cnt1 <> 4) AND(TCV_Cnt1 <> 6)]			Ma2)
25		(TCV_Res := OM_FHCHK(TCV_ch, CellChDes_20_Be(TCV_cchdescr), MoblAllc_20_Be2(TCV_mae1, TCV_mae2), TCV_chd1, 100, TCV_Fn))			
26	L_056	[NOT TCV_Res]		(F)	

27	8 L_0569	[TCV_Res]		(P)	
		ltree_LowerTestorManage			
28		(TCV_Fn := OM_ComingFn(TCV_ch), TCV_chd1 := ChDescrp_fh(TCV_chtype, TSPX_TmSltNotZero1, TSPX_TscDef, INT_TO_BIT((TSPX_MAIO MOD TCV_n), 6), INT_TO_BIT(TSPX_HSN, 6)), TCV_Strt := OC_StartTime(TCV_Fn, 100, 0), TCV_Null := OM_SendNextOn(TCV_ch, TCV_Fn))			
29		[(TCV_Cnt1 = 4) OR (TCV_Cnt1=6)]			Ma1)
30		(TCV_Null := OM_FreqDef(TCV_Strt, MoblAllc_20_Be1(TCV_mae1), TCV_ch, TCV_chd1, CellChDes_20_Be(TCV_cchdescr), TCV_Fn))			
31		[(TCV_Cnt1 <>4) AND(TCV_Cnt1<>6)]			Ma2)
32		(TCV_Null := OM_FreqDef(TCV_Strt, MoblAllc_20_Be2(TCV_mae1, TCV_mae2), TCV_ch, TCV_chd1, CellChDes_20_Be(TCV_cchdescr), TCV_Fn))			
		ltree_sendFreqRedef			
33		[TCV_Cnt1 <> 6]			
34		[TCV_Cnt1 = 4]			Ma1)
35		L!DL_DatRqFrqre	FrqRedfSnd(TCV_ch, FreqRedef_20(TSPX_TmSltNotZero 1, TCV_chtype, TCV_cchdescr, TCV_mae1, TCV_Strt, TCV_n))		
36		[TCV_Cnt1 <> 4]			Ma2)
37		L!DL_DatRqFrqre	FrqRedfSnd(TCV_ch, FreqRedef_22(TSPX_TmSltNotZero 1, TCV_chtype, TCV_cchdescr, TCV_mae1, TCV_mae2, TCV_Strt, TCV_n))		
38		[TCV_Cnt1 = 6]			Ma1)
39		L!DL_DatRqFrqre	FrqRedfSnd(TCV_ch, FreqRedef_21(TSPX_TmSltNotZero 1, TCV_chtype, TCV_mae1, TCV_Strt, TCV_n))		
		ltree_send_immass			
40		[TSPC_SvcOnTCH]			
41		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_nfh(TCV_Rr, TCV_Fn, '00001'B, TCV_ia_ts, TSPX_TscDef, TimingAdv(30), C_arfcnC, C_normal_paging))		
42		[NOT TSPC_SvcOnTCH]			
43		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_27(TCV_Rr, TCV_Fn, TCV_ia_ts, C_arfcnC, TimingAdv(30)))		
		ltree_setFreqRedefPara			
44		[TCV_Cnt1 = 1]			
45		(TCV_cchdescr := C_cchd_e_407, TCV_mae1:='00011100'B, TCV_mae2:='00000000'B, TCV_n := 3)			
46		[TCV_Cnt1 = 2]			

47		(TCV_cchdescr := C_cchd_e_408, TCV_mae1:='00000000'B, TCV_mae2:='11111000'B, TCV_n := 5)			
48		[TCV_Cnt1 = 3]			
49		(TCV_cchdescr := C_cchd_e_409, TCV_mae1:='00000000'B, TCV_mae2:='00001111'B, TCV_n := 4)			
50		[TCV_Cnt1 = 4]			
51		(TCV_cchdescr := C_cchd_e_415, TCV_mae1:='01100011'B, TCV_n := 4)			
52		[TCV_Cnt1 = 5]			
53		(TCV_cchdescr := C_cchd_e_411, TCV_mae1:='00000001'B, TCV_mae2:='11110000'B, TCV_n := 5)			
54		[TCV_Cnt1 = 6]			
55		(TCV_cchdescr := C_cchd_e_414, TCV_mae1:='00001110'B, TCV_n := 3)			
		local_ChConfig			
56		[TSPC_SvcOnTCH]			
57		+FullRateCh_A_1_nociph(C_Immass, TCV_ia_ts, TSPX_TscDef, ChMod_sign, FreqTCH(C_arfcnC), FreqTCH(C_arfcnC), TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
58		(TCV_ch := TCV_chTch)			
59		+SysInfo_5bisSending(TCV_sacchTch, TCV_sysinfo5bis)			
60		[NOT TSPC_SvcOnTCH]			
61		(TCV_chtype := INT_TO_BIT((8 + BIT_TO_INT(TSPX_SDCCH8SubDef)),5))			
62		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TCV_slot, TCV_tsc, ChMod_sign, FreqTCH(C_arfcnC), FreqTCH(C_arfcnC), TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3)			
63		+SysInfo_5bisSending(TCV_sacch8, TCV_sysinfo5bis)			

Detailed Comments: Ma1) Mobile Allocation with 1 octets
Ma2) Mobile Allocation with 2 octets

Test Case Dynamic Behaviour

Test Case Name: TC_26_10_3_1

Group: GSM_L3_MS_v4220/EGSMSignalling/

Purpose: 1) To verify that the MS in MM state "idle, updated" and in RR idle mode with a TMSI assigned, when made to initiate a call for a selected teleservice that is supported ba the MS as declared in a PICS/PIXIT statement, starts to initiate an immediate assignment procedure bz sending the CHANNEL REQUEST message.

2) To verify that subsequently after receipt of an IMMEDIATE ASSIGNMENT message allocating an SDCCH, after completion of establishment of the main signalling link, after having sent a CM SERVICE REQUEST message, after having successfully performed authenticiation and cipher mode setting procedures, after having sent a SETUP message, after having received a CALL PROCEEDING message followed by an ALERTING message and an ASSIGNMENT COMMAND message allocating an appropriate TCH, the MS sends an ASSIGNMENT COMPLETE message.

3) To verify that subsequently, after the suite of actions specified in test purposes 1 and 2, the MS after receiving a CONNECT message returns a CONNECT ACKNOWLEDGE message.

4) To verify that after the suite of actions specified in test purposes 1 and 2, the MS after receiving a CONNECT message attaches the user connection to the radion path. (This is checked by verifying that there is a point in time after transmission of the first L2 frame containing the (complete) CONNECT message, where the MS is sending appropriate speech or data frames whenever it doesn't have to transmit or acknowledge an I frame on layer 2 of the FACCH.)

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+ltree_SelectTeleservice			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_22, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_46, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_egsm, C_NCCP_2)			
4		(TCV_chTch := TCV_ch, TCV_chdescr_arfcn:= C_BCCHcarrierB_ho, TCV_ia_ts:= TSPX_TmSltG, TCV_asscmd_ts:= TSPX_TmSltC)			
5		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TCV_ia_ts, TCV_tsc, ChMod_sign, FreqSDCCH8_e, FreqSDCCH8_e, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
6		+SysInfo_5bisSending(TCV_sacch8, TCV_sysinfo5bis)			
7		+InitCall(TCV_Service)			
8		+local_continue			
		local_continue			
9		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
10		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
11		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_27(TCV_Rr, TCV_Fn, TCV_ia_ts, TCV_chdescr_arfcn, TimingAdv(0)))		
12		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
13		ACTIVATE(OtherEventsFail)			Restore Normal default

14		+Authentication(TCV_ch, TCV_cksn, TSPX_RANDDef)			
15		+Ciphering_on(TCV_ch)			
16		+ltree_ccest			
		ltree_SelectTeleservice			
17		[TSPC_Serv_TS11]			
18		+BasicServiceMO(C_Telephony, C_Full)			
19		[TSPC_Serv_TS61]			
20		+BasicServiceMO(C_AltSpchFax, C_Full)			
21		[TSPC_Serv_TS62]			
22		+BasicServiceMO(C_AutoFax, C_Full)			
		ltree_ccest			
23		(TCV_Null := OM_FreeResource(TCV_chTch))			
24		+FullRateCh_A_1(C_Ass, TCV_asscmd_ts, TCV_tsc, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcnEgsm, C_arfcnEgsm, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
25		+SysInfo_5bisSending(TCV_sacchTch, TCV_sysinfo5bis)			
26		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
27		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
28		L!DL_DatRqAlert	AlertSnd(TCV_ch, Alerting_01(TCV_TI))		
29		(TCV_AssCmd := AsgnCmd_nfh('00001'B, TCV_asscmd_ts, TSPX_TscDef, 9, C_arfcnEgsm, CellChDes_Omit, TCV_ChMod, StartingTm_omit, CphMod_omit))			
30		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
31		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
32		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
33		(TCV_Null := OO_TCHThroConnCHK(), TCV_Null := OO_TermCall())			
34		L?DL_DatInDisc	DiscRcv(TCV_chTch, DisconnR(TCV_TI0, Cause_Def))		
35		L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_08(TCV_TI))		
36		L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TCV_TI0))		
37		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_26_10_3_2

Group: GSM_L3_MS_v4220/EGSMSignalling/

Purpose: 1) To verify that the MS in the "idle, no IMSI" state (no SIM inserted) when made to call the number 112, sends a CHANNEL REQUEST message with establishment cause "emergency call".

2) To verify that after assignment of a dedicated channel the first layer message sent by the MS on the assigned dedicated channel is a CM SERVICE REQUEST message specifiying the correct CKSN and TMSI, with CM Service Type "emergency call establishment".

3) To verify that after receipt of a CM SERVICE ACCEPT message from the SS, the MS sends an EMERGENCY SETUP message.

4) To verify that subsequently, the SS having sent a CALL PROCEEDING message and then an ALERT message and having initiated the assignment procedure, the MS performs correctly that assignment procedure.

5) To verify subsequent correct performance of a connect procedure.

6) To verify that subsequently the MS has throught connected the TCH in both directions.

7) To verify that the call is cleared correctly.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(C_EmgCall, TSPX_EmgCallRate)			
3		(TCV_ia_ts:= TSPX_TmSltG)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1nonComb, C_BPM_3, C_T3212_1, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_Omit, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Ass, TCV_ia_ts, TCV_tsc, ChMod_sign, FreqSDCCH8_e, FreqSDCCH8_e, TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3)			1.
6		+local_TrafficChConf			
7	body	+InitCall(TCV_Service)			
8	L_0570	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_16)	(P)	2.
9		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
10		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_27(TCV_Rr, TCV_Fn, TCV_ia_ts, TCV_chdescr_arfcn, TimingAdv(0)))		
11	L_0571	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_06(TSPX_CKSNDef))	(P)	3.
12		ACTIVATE(OtherEventsFail)			Restore Normal default
13		+local_continue			
		local_continue			
14		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch,		

			CMServiceAcp_01)		
15		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
16		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
17		L!DL_DatRqAlert	AlertSnd(TCV_ch, Alerting_01(TCV_TI))		
18		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
19		(TCV_AssCmd.ch1d_at.arfcn:= INT_TO_BIT(C_arfcnEgsm,10))			
20		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
21		+local_continue1			
		local_continue1			
22		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
23	L_0572	L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))	(P)	
24		(TCV_Res := OO_TCHThroConnCHK())			
25	L_0573	[NOT TCV_Res]		(F)	6.
26		+PostMainLinkRel(TCV_chTch)			
27	L_0574	[TCV_Res]		(P)	7.
28		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
29	L_0575	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	
30		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
31		+PostMainLinkRel(TCV_chTch)			
		local_TrafficChConf			
32		[TCV_ChRate = C_Full]			
33		+FullRateCh_A_1_nociph(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH(C_arfcnEgsm), FreqTCH(C_arfcn_tchAd), TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3)			4.
34		[TCV_ChRate = C_Half]			
35		+HalfRateCh_A_1_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH(C_arfcnEgsm), FreqTCH(C_arfcn_tchAd), TimingAdv(0), C_BABR_0, C_cch_1nonComb, C_BPM_3)			5.

Detailed Comments:

1. To setup a physical channel
2. CHANNEL REQUEST with "emergency call establishment" cause received.
3. CM SERVICE REQUEST with "emergency call establishment" service type correct TMSI and CKSN received.
4. To setup aphysical channel as full rate traffic channel.
5. To setup aphysical channel as half rate traffic channel.
6. The TCH channel is not through connected, fail.
7. The TCH channel is through connected.

Test Group SS

Test Case Dynamic Behaviour

Test Case Name: TC_31_2_1_1_1

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that the MS correctly requests a supplementary service transaction for registration of call forwarding in CHANNEL REQUEST message.

2) To check that the MS correctly requests a supplementary service transaction for registration of call forwarding in the subsequent CM-SERVICE REQUEST.

3) To check that the MS sends a REGISTER message containing the invoke of the RegisterSS operation with the expected parameter values for registration of call forwarding.

4) To check that upon receipt of the result of the operation (in a RELEASE COMPLETE message), the MS provides the appropriate user indication (as described by the manufacturer).

These checks are done for :

a) CFNRy, for basic service group speech,

b) CFU, for basic service group all facsimile.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+part1			
4		+part2			
		part1			
5		(TCV_Null := OO_InitSS("**61*00431234*11*5#"))			2. international prefix + Country code
6		+ChannelReqtree			
7		+localtree			
8		+Checktree(C_RegCFNRy)			
		part2			
9		(TCV_Null := OO_InitSS("**21*1234*13#"))			5.
10		+ChannelReqtree			
11		+localtree1			
12		+Checktree(C_RegCFU)			
		localtree			
13		L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, RegisterSS_01), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].registerSSComponents.registerSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(RegisterSS_01)))		
14		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(RegisterSSRslt_01(TCV_InvkId))))		6.

		localtree1			
15		L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, RegisterSS_02), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].registerSSComponents.registerSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(RegisterSS_02)))		
16		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(RegisterSSRslt_02(TCV_InvkId))))		6.
		ChannelReqtree			
17		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
18		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
19		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
20		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
21		ACTIVATE(OtherEventsFail)			Restore Normal default
22		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
		Checktree(par:INTEGER)			
23		(TCV_Res := OO_SSresultCHK(par))			
24	L_0576	[TCV_Res]		(P)	3.
25		+PostMainLinkRel(TCV_ch)			
26	L_0577	[NOT TCV_Res]		(F)	4.
27		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4 with default parameters.
2. To initiate a registration of call forwarding service for CFNRy (speech).
3. The user indication is correct.
4. The user indication is wrong.
5. To initiate a registration of call forwarding service for CFU (all facsimile).
6. To return the ReturnResult of SSoperation by RELEASE COMPLETE message.

Test Case Dynamic Behaviour

Test Case Name: TC_31_2_1_1_2

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that the MS correctly requests the establishment of a parallel MM transaction for supplementary service transaction of registration of call forwarding, sending a CM-SERVICE REQUEST.

2) To check that the MS sends a REGISTER message containing the invoke of the RegisterSS operation with the expected parameter values for registration of call forwarding.

3) To check that upon receipt of the RELEASE COMPLETE message related to the present SS transaction, the first transaction remains unaffected.

4) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (as described by the Manufacturer).

These checks are performed with a call transaction already established for :

a) CFB, for all asynchronous services, the RELEASE COMPLETE message being sent with a facility IE containing a return_error(error) where error is "Bearer Service not provisioned".

b) CF, for all facsimile, the RELEASE COMPLETE message being sent with a facility IE containing a reject(invoke_problem) where invoke_problem is "resource limitation".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcH, TSPX_MOChRateH)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
6		(TCV_Null := OO_InitSS("**67*00431234*21#"))			3. international prefix + Country code
7		+part1			
8		(TCV_Null := OO_InitSS("**002*00431234*13#"))			4. international prefix + Country code
9		+part2			
		part1			
10		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
11		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
12		L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1)	Register_03(RegisterPdu_03(facilityIErcviei(RegisterSS_03)))		

13		(TCV_n := OC_PosinSet(TCV_Comp, RegisterSS_03), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].registerSSComponents.registerSS_InvokeComp.invokeID, 1))			
14		+localtree			
		localtree			
15		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(RegisterSSErr_01(TCV_InvkId))))		
16		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
17	L_0578	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
		part2			
18		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
19		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
20		L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1)	Register_03(RegisterPdu_03(facilityIErcviei(RegisterSS_04)))		
21		(TCV_n := OC_PosinSet(TCV_Comp, RegisterSS_04), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].registerSSComponents.registerSS_InvokeComp.invokeID, 1))			
22		+localtree1			
		localtree1			
23		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(RegisterSSRej_01(TCV_InvkId))))		
24		(TCV_Res := OO_SSresultCHK(C_RegCF))			
25	L_0579	[TCV_Res]		(P)	
26		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
27	L_0580	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
28		+PostMainLinkRel(TCV_chTch)			
29	L_0581	[NOT TCV_Res]		(F)	
30		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
31	L_0582	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
32		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:	1. To setup traffic channel. 2. To establish a mobile originating call. 3. To initiate a registration of call forwarding service for CFB (all asynchronous service). 4. To initiate a registration of call forwarding service for CF (all facsimile).

Test Case Dynamic Behaviour

Test Case Name: TC_31_2_1_2_1

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that the MS correctly requests supplementary service transaction for erasure of call forwarding in CHANNEL REQUEST message.

2) To check that the MS correctly requests supplementary service transaction for erasure of call forwarding in the subsequent CM-SERVICE REQUEST.

3) To check that the MS sends a REGISTER message containing the invoke of the EraseSS operation with the expected parameter values for erasure of call forwarding.

4) To check that upon receipt of the result of the operation (in a RELEASE COMPLETE message), the MS provides the appropriate user indication (as described by the manufacturer).

These checks are done for :

a) CFC, for basic service group all facsimile.

b) CFNRc, for all basic service groups.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_Null := OO_InitSS("##004**13#"))			2.
4		+part1			
5		(TCV_Null := OO_InitSS("##62#"))			3.
6		+part2			
		part1			
7		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
8		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
9		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
10		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
11		ACTIVATE(OtherEventsFail)			Restore Normal default
12		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
13		+localtree			
		localtree			
14		L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, EraseSS_01), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].eraseSSComponents.eraseSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(EraseSS_01)))		
15		L!DL_DatRqRelCmp	RelComSnd(TCV_ch,		4.

			ReleaseCmp_09(TCV_TI, facilityIEtsndiei(EraseSSRslt_01(TCV_InvkId))))		
16		+Checktree(C_ErsCFC)			
		part2			
17		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
18		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
19		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
20		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
21		ACTIVATE(OtherEventsFail)			Restore Normal default
22		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
23		+localtree1			
		localtree1			
24		L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, EraseSS_02), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].eraseSSComponents.eraseSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(EraseSS_02)))		
25		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(EraseSSRslt_02(TCV_InvkId))))		5.
26		+Checktree(C_ErsCFNRc)			
		Checktree(par:INTEGER)			
27		(TCV_Res := OO_SSresultCHK(par))			
28	L_0583	[TCV_Res]		(P)	6.
29		+PostMainLinkRel(TCV_ch)			
30	L_0584	[NOT TCV_Res]		(F)	7.
31		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4 with default parameters.
2. To initiate an EraseSS operation of call forwarding service for CFC (all facsimile).
3. To initiate an EraseSS operation of call forwarding service for CFNRc (all basic services).
4. To return the ReturnResult of SSoperation by RELEASE COMPLETE message.
5. To return the ReturnResult of SSoperation by RELEASE COMPLETE message.
6. The user indication is correct.
7. The user indication is wrong.

Test Case Dynamic Behaviour

Test Case Name: TC_31_2_1_2_2

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that the MS correctly requests the establishment of a parallel MM transaction for supplementary service transaction of erasure of call forwarding, sending a CM-SERVICE REQUEST.

2) To check that the MS sends a REGISTER message containing the invoke of the EraseSS operation with the expected parameter values for erasure of call forwarding.

3) To check that upon receipt of the RELEASE COMPLETE message related to the present SS transaction, the first transaction remains unaffected.

4) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (as described by the Manufacturer).

These checks are performed with a call transaction already established for :

a) CFU, for speech, the RELEASE COMPLETE message being sent with a facility IE containing a return_error(error) where error is "Teleservice not provisionned".

b) CFNRy, for all facsimile, the RELEASE COMPLETE message being sent with a facility IE containing a reject(invoke_problem) where invoke_problem is "resource limitation".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcI, TSPX_MOChRateI)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
6		(TCV_Null := OO_InitSS("##21**11#"))			3.
7		+part1			
8		(TCV_Null := OO_InitSS("##61**13#"))			4.
9		+part2			
		part1			
10		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
11		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
12		+localtree			
		localtree			
13		L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, EraseSS_03), TCV_InvkId	Register_03(RegisterPdu_03(facilityIErcviei(EraseSS_03)))		

		:= OC_Asn1intToOct(TCV_Comp.[TCV_n].eraseSSComponents.eraseSS_InvokeComp.invokeID, 1))			
14		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(EraseSSErr_01(TCV_InvkId))))		
15		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
16	L_0585	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
		part2			
17		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
18		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
19		+localtree1			
		localtree1			
20		L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, EraseSS_04), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].eraseSSComponents.eraseSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(EraseSS_04)))		
21		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(EraseSSRej_01(TCV_InvkId))))		
22		(TCV_Res := OO_SSresultCHK(C_ErsCFNRy))			
23	L_0586	[TCV_Res]		(P)	
24		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
25	L_0587	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
26		+PostMainLinkRel(TCV_chTch)			
27	L_0588	[NOT TCV_Res]		(F)	
28		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup traffic channel.
2. To establish a mobile originating call.
3. To initiate an erasure of call forwarding service for CFU (speech).
4. To initiate an erasure of call forwarding service for CFNRy (all facsimile).

Test Case Dynamic Behaviour

Test Case Name: TC_31_2_1_3

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that the MS correctly requests a supplementary service transaction for activation of call forwarding in CHANNEL REQUEST message.

2) To check that the MS correctly requests a supplementary service transaction for activation of call forwarding in the subsequent CM-SERVICE REQUEST.

3) To check that the MS sends a REGISTER message containing the invoke of the ActivateSS operation with the expected parameter values for activation of call forwarding.

4) To check that upon receipt of the result of the operation (in a RELEASE COMPLETE message), the MS provides the appropriate user indication (as described by the manufacturer).

These checks are done for :

a) CF, for basic service group "all synchronous services".

b) CFU, for all basic service groups.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_Null := OO_InitSS("*002**22#"))			2.
4		+part1			
5		(TCV_Null := OO_InitSS("*21#"))			3.
6		+part2			
		part1			
7		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
8		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
9		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
10		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
11		ACTIVATE(OtherEventsFail)			Restore Normal default
12		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
13		+localtree			
		localtree			
14		L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, ActivateSS_01), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].activateSSComponents.activateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(ActivateSS_01)))		

15		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(ActivateSSRslt_01(TCV_InvkId))))		
16		+Checktree(C_ActCF)			
		part2			
17		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
18		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
19		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
20		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
21		ACTIVATE(OtherEventsFail)			Restore Normal default
22		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
23		+localtree1			
		localtree1			
24		L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, ActivateSS_02), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].activateSSComponents.activateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(ActivateSS_02)))		
25		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(ActivateSSRslt_02(TCV_InvkId))))		5.
26		+Checktree(C_ActCFU)			
		Checktree(par:INTEGER)			
27		(TCV_Res := OO_SSresultCHK(par))			6.
28	L_0589	[TCV_Res]		(P)	
29		+PostMainLinkRel(TCV_ch)			
30	L_0590	[NOT TCV_Res]		(F)	
31		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4 with default parameters.
2. To initiate an ActivateSS operation of call forwarding service for CF (all synchronous services).
3. To initiate an ActivateSS operation of call forwarding service for CFU (all basic services).
4. To return the ReturnResult of SSoperation by FACILITY message.
5. To return the ReturnResult of SSoperation by RELEASE COMPLETE message.
6. To check whether the user indication is correct.

Test Case Dynamic Behaviour

Test Case Name: TC_31_2_1_4

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that the MS correctly requests a supplementary service transaction for deactivation of call forwarding in CHANNEL REQUEST message.

2) To check that the MS correctly requests a supplementary service transaction for deactivation of call forwarding in the subsequent CM-SERVICE REQUEST.

3) To check that the MS sends a REGISTER message containing the invoke of the DeactivateSS operation with the expected parameter values for deactivation of call forwarding.

4) To check that upon receipt of the result of the operation (in a RELEASE COMPLETE message), the MS provides the appropriate user indication (as described by the manufacturer).

These checks are done for :

a) CFC, for basic service group speech.

b) CFNRc, for basic service group all facsimile.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_Null := OO_InitSS("#004**11#"))			2.
4		+part1			
5		(TCV_Null := OO_InitSS("#62**13#"))			3.
6		+part2			
		part1			
7		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
8		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
9		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
10		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
11		ACTIVATE(OtherEventsFail)			Restore Normal default
12		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
13		+localtree			
		localtree			
14		L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, DeactivateSS_01), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].deactivateSSComponents.deactivateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(DeactivateSS_01)))		
15		L!DL_DatRqRelCmp	RelComSnd(TCV_ch,		

			ReleaseCmp_09(TCV_TI, facilityIEtsndiei(DeactivateSSRslt_01(TCV_InvkId))))		
16		+Checktree(C_DeactCFC)			
		part2			
17		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
18		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
19		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
20		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
21		ACTIVATE(OtherEventsFail)			Restore Normal default
22		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
23		+localtree1			
		localtree1			
24		L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, DeactivateSS_02), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].deactivateSSComponents.deactivateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(DeactivateSS_02)))		
25		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(DeactivateSSRslt_02(TCV_InvkId))))		5.
26		+Checktree(C_DeactCFNRc)			
		Checktree(par:INTEGER)			
27		(TCV_Res := OO_SSresultCHK(par))			6.
28	L_0591	[TCV_Res]		(P)	
29		+PostMainLinkRel(TCV_ch)			
30	L_0592	[NOT TCV_Res]		(F)	
31		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4 with default parameters.
2. To initiate a DeactivateSS operation of call forwarding service for CFC (speech).
3. To initiate a DeactivateSS operation of call forwarding service for CFNRc (all facsimile).
4. To return the ReturnResult of SSoperation by FACILITY message.
5. To return the ReturnResult of SSoperation by RELEASE COMPLETE message.
6. To check whether the user indication is correct.

Test Case Dynamic Behaviour

Test Case Name: TC_31_2_1_6_1

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that the MS correctly requests a supplementary service transaction for interrogation of a specific call forwarding in CHANNEL REQUEST message.

2) To check that the MS correctly requests a supplementary service transaction for interrogation of call forwarding in the subsequent CM-SERVICE REQUEST.

3) To check that the MS sends a REGISTER message containing the invoke of the InterrogateSS operation with the expected parameter values for interrogation of call forwarding.

4) To check that upon receipt of the result of the operation (in a RELEASE COMPLETE message), the MS provides the appropriate user indication (as described by the manufacturer).

These checks are done for :

a) CFB, for all basic service groups.

b) CFC, for basic service group speech.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_Null := OO_InitSS("*#67#"))			2.
4		+part1			
5		(TCV_Null := OO_InitSS("*#61**11#"))			3.
6		+part2			
		part1			
7		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
8		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
9		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
10		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
11		ACTIVATE(OtherEventsFail)			Restore Normal default
12		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
13		+localtree			
		localtree			
14		L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, InterrogateSS_01), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].interrogateSSComponents.interrogateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(InterrogateSS_01)))		

15		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(InterrogateSSRslt_01(TCV_InvkId))))		
16		+Checktree(C_InterrogCFB)			
		part2			
17		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
18		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
19		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
20		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
21		ACTIVATE(OtherEventsFail)			Restore Normal default
22		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
23		+localtree1			
		localtree1			
24		L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, InterrogateSS_02), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].interrogateSSComponents.interrogateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(InterrogateSS_02)))		
25		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(InterrogateSSRslt_02(TCV_InvkId))))		5.
26		+Checktree(C_InterrogCFC)			
		Checktree(par:INTEGER)			
27		(TCV_Res := OO_SSresultCHK(par))			6.
28	L_0593	[TCV_Res]		(P)	
29		+PostMainLinkRel(TCV_ch)			
30	L_0594	[NOT TCV_Res]		(F)	
31		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To setup a physical channel as BCCH, CCCH and SDCCH4 with default parameters.
2. To initiate a InterrogateSS operation of call forwarding service for CFB (all basice services).
3. To initiate a InterrogateSS operation of call forwarding service for CFNRy (Speech).
4. To return the ReturnResult of SSoperation by FACILITY message.
5. To return the ReturnResult of SSoperation by RELEASE COMPLETE message.
6. To check whether the user indication is correct.

Test Case Dynamic Behaviour

Test Case Name: TC_31_2_1_6_2

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that the MS correctly requests the establishmnt of a parallel MM transaction for supplementary service transaction of interrogation of a specific call forwarding service, sending a CM-SERVICE REQUEST.

2) To check that the MS sends a REGISTER message containing the invoke of the InterrogateSS operation with the expected parameter values for interrogation of call forwarding.

3) To check that upon receipt of the RELEASE COMPLETE message related to the present SS transaction, the first transaction remains unaffected.

4) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (as described by the Manufacturer).

These checks are performed with a call transaction already established for :

a) CFNRc, for all basic service group, the RELEASE COMPLETE message being sent with a facility IE containing a return_error(error) where error is "SS not available".

b) CFB, for all facsimile, the RELEASE COMPLETE message being sent with a facility IE containing a reject(invoke_problem) where invoke_problem is "resource limitation".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcJ, TSPX_MOChRateJ)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
6		(TCV_Null := OO_InitSS("*#62#"))			3.
7		+part1			
8		(TCV_Null := OO_InitSS("*#67**13#"))			4.
9		+part2			
		part1			
10		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
11		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
12		+localtree			
		localtree			
13		L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n :=	Register_03(RegisterPdu_03(facilityIErcviei(InterrogateSS_03)))		

		OC_PosinSet(TCV_Comp, InterrogateSS_03), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].interrogateSSComponents.interrogateSS_InvokeComp.invokeID, 1))			
14		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(InterrogateSSErr_01(TCV_InvkId))))		
15		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
16	L_0595	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
		part2			
17		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
18		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
19		+localtree1			
		localtree1			
20		L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, InterrogateSS_04), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].interrogateSSComponents.interrogateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(InterrogateSS_04)))		
21		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(InterrogateSSRej_01(TCV_InvkId))))		
22		+Checktree(C_InterrogCFB)			
		Checktree(par:INTEGER)			
23		(TCV_Res := OO_SSresultCHK(par))			
24	L_0596	[TCV_Res]		(P)	
25		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
26	L_0597	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
27		+PostMainLinkRel(TCV_chTch)			
28	L_0598	[NOT TCV_Res]		(F)	
29		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To setup traffic channel.
2. To establish a mobile originating call.
3. To initiate an interrogation of call forwarding service for CFNRc (all basic services).
4. To initiate an interrogation of call forwarding service for CFB (all facsimile).

Test Case Dynamic Behaviour

Test Case Name: TC_31_2_1_7_1_1

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that, in state U7 or U10, upon receipt of a FACILITY message notifying an incoming call, the MS shall provide the appropriate user indication (which is to be described by the manufacturer).

2) To check that when the notification of incoming call is received while the MS is in CC state U7 and U10 of another incoming call, it has no effect on its state.

These checks are performed in the case of CFB.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcA, TSPX_MTChRateA)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			Paging
6	body	L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		
7		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
8		L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))		
9		+ltree_connectTch			
		ltree_connectTch			
10		[TSPX_Immconn]			
11		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
12		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
13		+ltree_U10			
14		[NOT TSPX_Immconn]			
15		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
16		L?DL_DatInAlert	AlertRcv(AlertingInd_01(TI_01))		
17		L!DL_DatRqFac	FacilitySnd(TCV_chTch, FacilityPdu_25(TI_02, facilityIEtsnd(NotificationSS_01)))		2.
18		(TCV_Res := OO_SSresultCHK(C_NotifyCFB))			
19		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(		

			TI_02))		
20	L_0599	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TI_01, C_U7))	(P)	U7
21		(TCV_Null := OO_HookOff())			
22		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
23	L_0600	[TCV_Res]		(P)	Notified
24		+ltree_U10			
25	L_0601	[NOT TCV_Res]		(F)	Not notified
26		+ltree_U10			
		ltree_U10			
27		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
28		L!DL_DatRqFac	FacilitySnd(TCV_chTch, FacilityPdu_25(TI_02, facilityIEtsnd(NotificationSS_01)))		3.
29		(TCV_Res := OO_SSresultCHK(C_NotifyCFB))			
30		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TI_02))		
31	L_0602	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TI_01, C_U10))	(P)	U10
32		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TI_02))		
33		+PostMainLinkRel(TCV_chTch)			
34	L_0603	[TCV_Res]		(P)	Notified
35	L_0604	[NOT TCV_Res]		(F)	Not notified

Detailed Comments:
1. To setup physical channels for BCCH, CCCH, SDCCH4 and full rate traffic channels.
2. To send a FACILITY message containing NotifySS invocation while the MS is in U7 state.
3. To send a FACILITY message containing NotifySS invocation while the MS is in U10 state.

Test Case Dynamic Behaviour

Test Case Name: TC_31_2_1_7_1_2

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that when an outgoing call is being established, if the ALERTING message is received with the facility information element containing an SS notification, the MS correctly reaches CC state U4. This is tested for CFU.

2) As an outgoing call is being established, if the ALERTING message is received with the facility information element containing an SS notification, the MS provides the appropriate user indication (which is to be described by the manufacturer). This is tested for CFU.

3) As an outgoing call is being established, if the CONNECT message is received with the facility information element containing an SS notification, the MS normally sends a CONNECT ACK message and enter CC state U10. This is tested for CFC.

4) As an outgoing call is being established, if the CONNECT message is received with the facility information element containaining an SS notification (for CFU or CFC), the MS provides the appropriate user indication (which is to be described by the manufacturer). This is tested for CFC.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcJ, TSPX_MOChRateJ)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+InitCall(TCV_Service)			
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
9		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
12		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
13		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
14		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			

15		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
16		+continue			
		continue			
17		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_04(TCV_TI, facilityIEtsndiei(NotificationSS_02)))		2.
18		(TCV_Res := OO_SSresultCHK(C_NotifyCFU))			
19	L_0605	[TCV_Res]		(P)	
20		+localtree			
21	L_0606	[NOT TCV_Res]		(F)	
22		+localtree			
		localtree			
23		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
24	L_0607	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U4))	(P)	
25		L!DL_DatRqConn	ConnSnd(TCV_chTch , Connect_05(TCV_TI, facilityIEtsndiei(NotificationSS_03)))		3.
26		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
27		(TCV_Res := OO_SSresultCHK(C_NotifyCFC))			
28	L_0608	[TCV_Res]		(P)	
29		+localtree1			
30	L_0609	[NOT TCV_Res]		(F)	
31		+localtree1			
		localtree1			
32		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
33	L_0610	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
34		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
35		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:
1. To setup traffic channel.
2. To send an ALERTING message containing NotifySS invocation while the MS is in U4 state.
3. To send a CONNECT message containing NotifySS invocation while the MS is in U10 state.

Test Case Dynamic Behaviour

Test Case Name: TC_31_2_1_7_2

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that, upon receipt of the SETUP message containing a notification indication that the call is a forwarded one, the MS correctly continues call establishment and enters CC state U6.

2) Upon receipt of the SETUP message containing a notification indication that the call is a forwarded one, the MS provides the appropriate user indication (which is to be described by the manufacturer).

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcE, TSPX_MTChRateE)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
5		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
6		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
7		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
8		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
9		ACTIVATE(OtherEventsFail)			Restore Normal default
10		(TCV_Setup_mt.fie := facilityIEtsndiei(NotificationSS_04))			
11		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		2.
12		L!DL_DatRqCcstEnq	CCStESnd(TCV_ch, CCStatusEq_01(TI_02))		
13	L_0611	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TI_01, C_U6))	(P)	
14		(TCV_Res := OO_SSresultCHK(C_NotifyCFNRc))			
15	L_0612	[TCV_Res]		(P)	
16		+localtree			
17	L_0613	[NOT TCV_Res]		(F)	
18		+localtree			
		localtree			
19		L!DL_DatRqRelCmp	RelComSnd(		

20		+PostMainLinkRel(TCV_ch)	TCV_ch, ReleaseCmp_08(TI_02))		

Detailed Comments:	1. To setup a physical channel as BCCH, CCCH and SDCCH4. 2. To send a setup message containing facility IE (notification, forwarded call).

Test Case Dynamic Behaviour

Test Case Name: TC_31_6_1_1

Group: GSM_L3_MS_v4220/SS/

Purpose:
1) To verify that when the MS receives the AOCC parameters in a Facility IE which is contained in the CONNECT message and when a TCH has already been assigned, the MS returns a FACILITY message containing the acknowledgement within 1 second.

2) To verify that when the MS receives the AOCC parameters in a Facility IE which is contained in a CONNECT message and when a TCH has already been assigned, the MS stores the correct value in the ACM field of the SIM.

3) To verify that the when the call has no volume related component the MS ignores non-zero AOCC e5, e6 parameters sent to it.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(1800)			
2		+BasicServiceMO(TSPX_MOBscSvcF, TSPX_MOChRateF)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		(TCV_Null := OO_ACMReading())			1.
5		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+execution1			
7		+execution2			
8		+execution3			
9		+execution4			
10		+execution5			
		execution1			
11		+EstablishFacMO(90000, FwdChAdvSS_01, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
12		+localtree("43")			
		execution2			
13		+EstablishFacMO(90000, FwdChAdvSS_02, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
14		+localtree("100")			
		execution3			
15		+EstablishFacMO(90000, FwdChAdvSS_03, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
16		+localtree("2000")			
		execution4			
17		+EstablishFacMO(90000, FwdChAdvSS_04, TSPX_SDCCH4SubDef, TSPX_TmSltDef,			

		TSPX_TscDef, TSPX_TCHHSubDef)			
18		+localtree("89 or 90")			
		execution5			
19		+EstablishFacMO(90000, FwdChAdvSS_05, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
20		+localtree("50 or 62.5")			
		localtree(val:IA5String)			
21		?TIMEOUT T_dly			
22		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
23		L?DL_DatInRel	ReleaseRcv(Release_02)		
24		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
25		(TCV_Res := OO_ACMIncCHK(val))			
26		+PostMainLinkRel(TCV_chTch)			
27	L_0614	[TCV_Res]		(P)	
28	L_0615	[NOT TCV_Res]		(F)	

Detailed Comments:

1. To read and note the value of ACM on SIM at the begining of the test,
2. To send Facility IE of ForwardChargeAdvice using definite form mixed with indefinite form.
3. To check whether the increment of the value of ACM on SIM is the expected value.

Test Case Dynamic Behaviour

Test Case Name: TC_31_6_1_2

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To verify that when the MS receives certain AOCC e-parameters in a Facility IE which is contained in a FACILITY message sent after the CONNECT message and when a TCH has already been assigned, the MS returns a FACILITY message containing the acknowledgement within 1 second.

2) To verify that when the MS receives the AOCC parameters in a Facility IE which is contained in a FACILITY message and when a TCH has already been assigned, the MS stores the correct value in the ACM field of the SIM.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(1800)			
2		+BasicServiceMT(TSPX_MTBscSvcA, TSPX_MTChRateA)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		(TCV_Null := OO_ACMReading())			1.
5		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6	body	+execution(FwdChAdvSS_06, "0")			
7		+execution(FwdChAdvSS_07, "100")			
8		+execution(FwdChAdvSS_08, "43")			
9		+execution(FwdChAdvSS_09, "89 or 90")			
10		+execution(FwdChAdvSS_10, "50 or 62.5")			
		execution(comp:Component_T; para:IA5String)			
11		+PreEnterCCstateU8_34_SS			
12		L!DL_DatRqFac START T_dly(90000), START T_dly1(10000)	FacilitySnd(TCV_chTch, FacilityPdu_25(TI_02, facilityIEtsnd(comp)))		2.
13	L_0616	L?DL_DatInFac (TCV_Fn1 := DL_DatInFac.fn) CANCEL T_dly1	FacilityRcv(FacilityPdu_26(TI_01, facilityIErcv(FwdChAdvRslt_01)))	(P)	6.
14		(TCV_Fn := OM_ReturnFn(TCV_chTch))			5.
15		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
16		+Check_Time(C_T_Wait_FAC)			
17		+post(para)			
18	L_0617	?TIMEOUT T_dly1		(F)	3.
19		+PostLinkRelEnd(TCV_chTch)			
		post(para:IA5String)			
20		?TIMEOUT T_dly			

21		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TI_02, Cause_01, ProgInd_omit, UuInfo_omit))		
22		L?DL_DatInRel	ReleaseRcv(Release_02)		
23		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TI_02))		
24		+PostMainLinkRel(TCV_chTch)			
25		(TCV_Res := OO_ACMIncCHK(para))			
26	L_0618	[TCV_Res]		(P)	4.
27	L_0619	[NOT TCV_Res]		(F)	

Detailed Comments:

1. To read and note the value of ACM on SIM at the beginning of the test,
2. To send Facility IE of ForwardChargeAdvice using definite form mixed with indefinite form.
3. The expected FACILITY message does not return, fail.
4. To check whether the increment of the value of ACM on SIM is correct.
5. Get the first frame number for transmitting of FACILITY message containing AOC information.
6. TCV_Fn1 contains the first frame number of the received FACILITY message with AOC acknowledge from mobile.

Test Case Dynamic Behaviour

Test Case Name: TC_31_6_1_5

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To verify that when the MS receives new AOCC parameters mid-way through a call in a Facility IE which is contained within a FACILITY message the MS returns a FACILITY message containing the acknowledgement within 1 second.

2) To verify that when the MS receives new charging information mid-way through a call in the form of a Facility IE contained within a FACILITY message the MS correctly indicates the total charge considering both sets of charging information

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+BasicServiceMO(TSPX_MOBscSvcJ, TSPX_MOChRateJ)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		(TCV_Null := OO_ACMReading())			1.
5		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
6		+ltree_body			
		ltree_body			
7		+InitCall(TCV_Service)			
8		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
9		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
10		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
11		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
12		ACTIVATE(OtherEventsFail)			Restore Normal default
13		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
14		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
15		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
16		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
17		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
18		L!DL_DatRqAlert	AlertSnd(TCV_chTch,		

			Alerting_01(TCV_TI))		
19		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
20		+continue			
		continue			
21		+localtree(FwdChAdvSS_11, 80000)			
22		+localtree1(FwdChAdvSS_12, 100000)			
23		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
24		L?DL_DatInRel	ReleaseRcv(Release_02)		
25		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
26		+PostMainLinkRel(TCV_chTch)			
27		+post			
		localtree(comp:Component_T; t:INTEGER)			
28		L!DL_DatRqFac START T_dly(t), START T_dly1(10000)	FacilitySnd(TCV_chTch, FacilityPdu_25(TCV_TI, facilityIEtsnd(comp)))		2.
29		+AOC_CHK_FAC(TCV_TI0)			
30		?TIMEOUT T_dly			
		localtree1(comp:Component_T; t:INTEGER)			
31		L!DL_DatRqFac START T_dly(t), START T_dly1(10000)	FacilitySnd(TCV_chTch, FacilityPdu_25(TCV_TI, facilityIEtsnd(comp)))		2.
32	L_1046	L?DL_DatInFac (TCV_Fn := OM_ReturnFn(TCV_chTch), TCV_Fn1 := DL_DatInFac.fn) CANCEL T_dly1	FacilityRcv(FacilityPdu_26(TCV_TI0, facilityIErcv(FwdChAdvRslt_01)))	(P)	
33		+Check_Time(C_T_Wait_FAC)			
34		?TIMEOUT T_dly			
35	L_1047	?TIMEOUT T_dly1		(F)	
36		CANCEL T_dly			
37		+PostLinkRelEnd(TCV_chTch)			
		post			
38		(TCV_Res := OO_ACMIncCHK("65"))			3.
39	L_0620	[TCV_Res]		(P)	
40	L_0621	[NOT TCV_Res]		(F)	

Detailed Comments:

1. To read and note the value of ACM on SIM at the beginning of the test,
2. To send second CAI.
3. To check whether the increment of the value of ACM on SIM is 65.

Test Case Dynamic Behaviour

Test Case Name: TC_31_6_1_6

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To verify that when the MS receives a Facility IE in which certain e-parameters are set to zero the total charge accumulated is the same as that when the same e-parameters are completely omitted from the Facility IE.

2) To verify the operation of a shortened channel release procedure where the SS does not send DISCONNECT but only the RELEASE COMPLETE and CHANNEL RELEASE messages or just the CHANNEL RELEASE message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(1200)			
2		+BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		(TCV_Null := OO_ACMReading())			1.
6		+execution1			
7		+execution2			
8		+execution3			
		execution1			
9		+start(FwdChAdvSS_13)			
10		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
11		L?DL_DatInRel	ReleaseRcv(Release_02)		
12		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
13		+localtree			
		execution2			
14		+start(FwdChAdvSS_14)			
15		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
16		+localtree			
		execution3			
17		+start(FwdChAdvSS_14)			
18		+localtree			

		localtree			
19		+PostMainLinkRel(TCV_chTch)			
20		(TCV_Res := OO_ACMIncCHK("20"))			4.
21	L_0622	[TCV_Res]		(P)	
22	L_0623	[NOT TCV_Res]		(F)	
		start(comp:Component_T)			
23		+InitCall(TCV_Service)			
24		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
25		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
26		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
27		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
28		ACTIVATE(OtherEventsFail)			Restore Normal default
29		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
30		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
31		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
32		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
33		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
34		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
35		L!DL_DatRqFac START T_dly(90000)	FacilitySnd(TCV_chTch, FacilityPdu_25(TCV_TI, facilityIEtsnd(comp)))		2.
36		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
37		+branch			
38		?TIMEOUT T_dly			
		branch			
39	L_0624	L?DL_DatInFac	FacilityRcv(FacilityPdu_26(TCV_TI0, facilityIErcv(FwdChAdvRslt_01)))	(P)	3.
40		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
41		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
42	L_0625	L?DL_DatInFac	FacilityRcv(FacilityPdu_26(TCV_TI0, facilityIErcv(FwdChAdvRslt_01)))	(P)	3.

Detailed Comments:

1. To read and note the value of ACM on SIM at the beginning of the test,
2. To send Facility IE of ForwardChargeAdvice using definite form mixed with indefinite form.
3. The expected FACILITY message received, pass.

4. To check whether the increment of the value of ACM on SIM is 20.

Test Case Dynamic Behaviour

Test Case Name: TC_31_6_1_7

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To verify that when the MS invokes a Call Hold call and hence receives Facility IEs containing AOCC e-parameters for each chargeable call the MS returns a FACILITY message containing the AOCC acknowledgement within 1 second of transmission of each set of e-parameters.

2) To verify that when the MS invokes a Call Hold call and hence receives Facility IEs containing CAI elements for each chargeable call the CCM records the sum of all the charges for the services currently being used and hence that the ME inserts the correct charge in the ACM field of the SIM.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+BasicServiceMO(TSPX_MOBscSvcC, TSPX_MOChRateC)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		(TCV_Null := OO_ACMReading())			1.
6		+EstablishFacMO(180000, FwdChAdvSS_15, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
7		+localtree			
		localtree			
8		+InitCall(TCV_Service)			Place the first call on hold
9		ACTIVATE(DTMFsig, OtherEventsFail)			Accept DTMF signalling
10		L?DL_DatInHold	HoldRcv(Holdpdu_01(TCV_TI0))		
11		ACTIVATE(OtherEventsFail)			
12		L!DL_DatRqHoldAck	HoldAckSnd(TCV_chTch, HoldAckpdu_01(TCV_TI))		
13		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_01)		
14		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
15		+SetupRcvMo2(SetupInd_01)			
16		L!DL_DatRqCallProc	CallProc(TCV_chTch, TCV_CallProc)		
17		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI2))		
18		L!DL_DatRqConn START T_dly2(90000), START T_dly1(	ConnSnd(TCV_chTch, Connect_05(		3.

		10000)	TCV_TI2, facilityIEtsndiei(FwdChAdvSS_16)))		
19		+AOC_CHK_FAC(TCV_TI1)			
20		+localtree1			
		localtree1			
21		?TIMEOUT T_dly2			
22		+releasecall(TCV_TI2)			4.
23		?TIMEOUT T_dly			
24		+releasecall(TCV_TI)			5.
25		+releaselink			
		releasecall(ti:TI)			
26		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(ti, Cause_01, ProgInd_omit, UuInfo_omit))		
27		L?DL_DatInRel	ReleaseRcv(Release_02)		
28		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(ti))		
		releaselink			
29		+PostMainLinkRel(TCV_chTch)			
30		(TCV_Res := OO_ACMIncCHK("54"))			6.
31	L_0626	[TCV_Res]		(P)	
32	L_0627	[NOT TCV_Res]		(F)	

Detailed Comments:

1. To read and note the value of ACM on SIM at the beginning of the test,
2. To send Facility IE of ForwardChargeAdvice using definite form mixed with indefinite form.
3. To send second CAI.
4. To release call C.
5. To release call B.
6. To check whether the increment of the ACM value on SIM is 54.

Test Case Dynamic Behaviour

Test Case Name: TC_31_6_1_8

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To verify that when the MS invokes a Multi-party call and hence receives Facility IEs containing AOCC e-parameters for each chargeable call the MS returns a FACILITY message containing the AOCC acknowledgement within 1 second of transmission of each set of e-parameters.

2) To verify that when the MS originates a Multi-party call and hence receives Facility IEs containing CAI elements for each chargeable call the CCM records the sum of all the charges for the services currently being used and hence the ME inserts the correct charge in the ACM field of the SIM.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+BasicServiceMO(TSPX_MOBscSvcI, TSPX_MOChRateI)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		(TCV_Null := OO_ACMReading())			1.
6		+EstablishFacMO(180000, FwdChAdvSS_17, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
7		+localtree			
		localtree			
8		+InitCall(TCV_Service)			3.
9		ACTIVATE(DTMFsig, OtherEventsFail)			Accept DTMF signalling
10		L?DL_DatInHold	HoldRcv(Holdpdu_01(TCV_TI0))		
11		ACTIVATE(OtherEventsFail)			Restore Normal Defaults
12		L!DL_DatRqHoldAck	HoldAckSnd(TCV_chTch, HoldAckpdu_01(TCV_TI))		
13		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_01)		
14		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
15		+SetupRcvMo2(SetupInd_01)			
16		L!DL_DatRqCallProc	CallProc(TCV_chTch, TCV_CallProc)		
17		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI2))		
18		L!DL_DatRqConn START T_dly2(90000), START T_dly1(10000)	ConnSnd(TCV_chTch , Connect_05(TCV_TI2, facilityIEtsndiei(		4.

			FwdChAdvSS_18)))		
19		+AOC_CHK_FAC(TCV_TI1)			
20		+localtree1			
		localtree1			
21		(TCV_Null := OO_MptyCall())			5.
22		ACTIVATE(DTMFsig, OtherEventsFail)			Accept DTMF signalling
23		L?DL_DatInFac (TCV_Comp := DL_DatInFac.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, BldMptySS_01), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].registerSSComponents.registerSS_InvokeComp.invokeID, 1), TCV_TI3 := DL_DatInFac.msg.ti, TCV_TI3.ti_f := '1'B)	FacilityRcv(FacilityPdu_26(TCV_TI0, facilityIErcv(BldMptySS_01)))		
24		+ltree_release			
25		L?DL_DatInFac (TCV_Comp := DL_DatInFac.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, BldMptySS_01), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].registerSSComponents.registerSS_InvokeComp.invokeID, 1), TCV_TI3 := DL_DatInFac.msg.ti, TCV_TI3.ti_f := '1'B)	FacilityRcv(FacilityPdu_26(TCV_TI1, facilityIErcv(BldMptySS_01)))		
26		+ltree_release			
		ltree_release			
27		ACTIVATE(OtherEventsFail)			Restore Normal Defaults
28		L!DL_DatRqFac	FacilitySnd(TCV_chTch, FacilityPdu_25(TCV_TI3, facilityIEtsnd(BuildMptySSRslt_01(TCV_InvkId))))		
29		?TIMEOUT T_dly2			
30		+releasecall(TCV_TI2)			
31		?TIMEOUT T_dly			
32		+releasecall(TCV_TI)			
33		+releaselink			
		releasecall(ti: TI)			
34		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(ti, Cause_01, ProgInd_omit, UuInfo_omit))		
35		L?DL_DatInRel	ReleaseRcv(Release_02)		
36		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(ti))		
		releaselink			
37		+PostMainLinkRel(TCV_chTch)			
38		(TCV_Res := OO_ACMIncCHK("134"))			6.
39	L_0628	[TCV_Res]		(P)	
40	L_0629	[NOT TCV_Res]		(F)	

Detailed Comments:

1. To read and hote the value of ACM on SIM at the beginning of the test,
2. To send Facility IE of ForwardChargeAdvice using definite form mixed with indefinite form.
3. To hold the first call and make a second call.
4. To send CAI for the second call.
5. To build the multi party call.
6. To check whether the increment of the value of ACM on SIM is 134.

Test Case Dynamic Behaviour

Test Case Name: TC_31_6_2_1
Group: GSM_L3_MS_v4220/SS/
Purpose: 1) To verify that when the SIM is removed from the ME during an active AOCC call the ME immediately terminates the call.

2) To verify that when the SIM is removed during an active AOCC call the ME has written the total charge up to that point in the call to the ACM field of the SIM.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcD, TSPX_MOChRateD)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		(TCV_Null := OO_ACMReading())			1.
6	body	+EstablishFacMO(90000, FwdChAdvSS_19, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
7		+localtree			
		localtree			
8		?TIMEOUT T_dly			
9		START T_dly(60000)			
10		(TCV_Null := OO_SIMRmv())			3.
11		L?DL_DatInDisc CANCEL T_dly	DiscRcv(TCV_chTch, DisconnR(TCV_TI0, Cause_Def))		
12		L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_03(TCV_TI))		
13	L_0630	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TCV_TI0))	(P)	
14		+PostMainLinkRel(TCV_chTch)			
15		+localtree1			
16	L_0631	L?DL_DatInRelCmp CANCEL T_dly	RelComRcv(ReleaseCmp_03(TCV_TI0))	(P)	
17		+PostMainLinkRel(TCV_chTch)			
18		+localtree1			
19	L_0632	L?DL_RelIn CANCEL T_dly	DLRelInd_01	(P)	
20		+localtree1			
21	L_0633	?TIMEOUT T_dly		(F)	
22		+PostLinkRelEnd(TCV_chTch)			
		localtree1			

23		(TCV_Res := OO_ACMIncCHK("30"))			4.
24	L_0634	[TCV_Res]		(P)	
25	L_0635	[NOT TCV_Res]		(F)	

Detailed Comments:

1. To read and note the value of ACM on SIM at the beginning of the test,
2. To send Facility IE of ForwardChargeAdvice using definite form mixed with indefinite form.
3. To remove the SIM without power off and 90 seconds after CAI sent.
4. To check whether the increment of the value of ACM on SIM is 30.

Test Case Dynamic Behaviour

Test Case Name: TC_31_6_2_2
Group: GSM_L3_MS_v4220/SS/
Purpose: To verify that when the power supply of the MS is removed during an active AOCC call the ME has written the total charge up to that point in the call to the ACM field of the SIM.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+BasicServiceMO(TSPX_MOBscSvcE, TSPX_MOChRateE)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		(TCV_Null := OO_ACMReading())			1.
6		+execution1			
7		+execution2			
		execution1			
8		+main			
9		(TCV_Null := OO_SwitchOff())			3.
10		(TCV_Res := OO_ACMIncCHK("30"))			4.
11	L_0636	[TCV_Res]		(P)	
12	L_0637	[NOT TCV_Res]		(F)	
		execution2			
13		(TCV_Null := OO_SwitchOn())			5.
14		START T_dly(20000)			
15		?TIMEOUT T_dly			
16		+main			
17		(TCV_Null := OO_PowerDown())			6.
18		(TCV_Res := OO_ACMIncCHK("30"))			4.
19	L_0638	[TCV_Res]		(P)	
20	L_0639	[NOT TCV_Res]		(F)	
		main			
21		+EstablishFacMO(90000, FwdChAdvSS_19, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
22		?TIMEOUT T_dly			

Detailed Comments:
1. To read and note the value of ACM on SIM at the beginning of the test,
2. To send Facility IE of ForwardChargeAdvice using definite form mixed with indefinite form.
3. To switch off the MS 90 seconds after CAI sent.
4. To check whether the increment of the value of ACM on SIM is 30.
5. To switch on the MS and wait for the MS back to idle state.
6. To remove battery pack 90 seconds after CAI sent.

Test Case Dynamic Behaviour

Test Case Name: TC_31_6_2_3

Group: GSM_L3_MS_v4220/SS/

Purpose: To verify that when the MS goes out of radio coverage area and an active call is dropped the ME has written the total charge up to that point in the call to the ACM field of the SIM.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcH, TSPX_MOChRateH)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		(TCV_Null := OO_ACMReading())			1.
6		+EstablishFacMO(90000, FwdChAdvSS_19, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
7		?TIMEOUT T_dly			
8		(TCV_Null := OM_StopCell(C_CellA))			3.
9		(TCV_Res := OO_ACMIncCHK("30"))			4.
10	L_0640	[TCV_Res]		(P)	
11	L_0641	[NOT TCV_Res]		(F)	

Detailed Comments:

1. To read and note the value of ACM on SIM at the beginning of the test,
2. To send Facility IE of ForwardChargeAdvice using definite form mixed with indefinite form.
3. To switch off the cell A 90 seconds after CAI sent.
4. To check whether the increment of the value of ACM on SIM is 30.

Test Case Dynamic Behaviour

Test Case Name: TC_31_6_2_4

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To verify that when the value stored in the ACM becomes equal to or exceeds its maximum value, the ACMM, any outgoing calls in progress for which a non-zero CAI exists are terminated by the ME, once the chargeable interval determinined by the CAI has elapsed, with an appropriate indication given to the user.

2) To verify that when the value stored in the ACM becomes equal to or exceeds its maximum limit, the ACMM, the making of non-emergency calls is inhibited

3) To verify that when the value stored in the ACM becomes equal to or exceeds its maximum limit, the ACMM, the making of emergency calls is uninhibited

Default: OtherEventsFail

Comments: The ACM is reset to zero and the ACMmax is set to 2 units before starting the test

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+BasicServiceMO(TSPX_MOBscSvcI, TSPX_MOChRateI)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		(TCV_Null := OO_ACMSetting())			1.
6		+execution1			
7		+execution2			
8		+execution3			
		execution1			
9		+EstablishFacMO(100000, FwdChAdvSS_20, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
10		L?DL_DatInDisc READTIMER T_dly(TCV_Time), CANCEL T_dly	DiscRcv(TCV_chTch, DisconnR(TCV_TI0, Cause_Def))		3.
11	L_0642	[(88000 <= TCV_Time) AND (TCV_Time <= 92000)]		(P)	4.
12		+localtree1			
13	L_0643	[(88000 > TCV_Time) OR(TCV_Time > 92000)]		(F)	
14		+localtree1			
		localtree1			
15		L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_08(TCV_TI))		
16		L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TCV_TI0))		
17		+PostMainLinkRel(TCV_chTch)			

18		(TCV_Res := OO_ACMIncCHK("2"))			5.
19	L_0644	[TCV_Res]		(P)	
20	L_0645	[NOT TCV_Res]		(F)	
		execution2			
21		+InitCall(TCV_Service)			
22		START T_dly(5000)			
23	L_0646	?TIMEOUT T_dly		(P)	
24	L_0647	L?DL_RacInChRq CANCEL T_dly	ChReq(ChRequest_02)	(F)	
		execution3			
25		+BasicServiceMO(C_EmgCall, TSPX_EmgCallRate)			
26		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
27		+InitCall(C_EmgCall)			6.
28		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_16)		
29		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
30		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
31		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
32		ACTIVATE(OtherEventsFail)			Restore Normal default
33		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
34		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
35		+continue1			
		continue1			
36		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
37		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
38		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
39		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
40		L!DL_DatRqConn START T_dly1(10000), START T_dly(120000)	ConnSnd(TCV_chTch, Connect_05(TCV_TI, facilityIEtsndiei(FwdChAdvSS_21)))		7.
41		+AOC_CHK_FAC(TCV_TI0)			
42		?TIMEOUT T_dly			9.
43		(TCV_Null := OO_TermCall())			
44		+localtree2			
		localtree2			
45	L_0648	L?DL_DatInDisc	DiscRcv(TCV_chTch, DisconnR(TCV_TI0, Cause_Def))	(P)	
46		L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_08(		

			TCV_TI))		
47		L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TCV_TI0))		
48		+PostMainLinkRel(TCV_chTch)			
49		(TCV_Res := OO_ACMIncCHK("0"))			8.
50	L_0649	[TCV_Res]		(P)	
51	L_0650	[NOT TCV_Res]		(F)	

Detailed Comments:

1. To reset the ACM = 0 and set the ACMmax to 2.
2. To send Facility IE of ForwardChargeAdvice using definite form mixed with indefinite form.
3. The call is terminated when the ACM reaches the ACMmax (cause value #68).
4. The time duration is 90 +- 2 seconds, pass.
5. To check whether the ACM increment is 2.
6. To make an emergency call.
7. To send Facility IE of ForwardChargeAdvice using definite form mixed with indefinite form.
8. To check whether the value of ACM is still 2.
9. The time duration is 120 seconds.

Test Case Dynamic Behaviour

Test Case Name: TC_31_6_2_5

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To verify that when the value stored in the ACM becomes equal to or exceeds its maximum value, the ACMM, any mobile terminating calls in progress for which a non-zero CAI exists are terminated by the ME, once the chargeable interval determinined by the CAI has elapsed, with an appropriate indication given to the user.

2) To verify that when the value stored in the ACM becomes equal to or exceeds its maximum value, the ACMM, and an incoming call is received for which subsequently a non-zero CAI is received, then the call is terminated by the ME with an appropriate indication given to the user.

3) To verify that when the value stored in the ACM becomes equal to or exceeds its maximum limit,
the ACMM, the receiving of calls for which the CAI is zero is uninhibited.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+BasicServiceMT(TSPX_MTBscSvcA, TSPX_MTChRateA)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		(TCV_Null := OO_ACMSetting())			1.
6	body	+execution1			
7		+execution2			
8		+execution3			
		execution1			
9		+init			
10		+branchA(FwdChAdvSS_20)			
11		+branchF			
12		+TimerCheck			
13		+step20("2")			5.
		execution2			
14		+init			
15		+branchB			
16		+branchC			
17		+branchG			
18		+step20("0")			8.
19		+branchD			
20		+branchG			
21		+step20("0")			8.
22		+branchE			
23		+branchG			
24		+step20("0")			8.
		execution3			
25		+init			

26		+branchA(FwdChAdvSS_21)			
27		+branchH			
28		+step20("0")			8.
		init			
29		+PreEnterCCstateU8_34_SS			
30		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
		branchA(comp:Component_T)			
31		L!DL_DatRqFac START T_dly(120000)	FacilitySnd(TCV_chTch, FacilityPdu_25(TI_02, facilityIEtsnd(comp)))		2., 7.
32	L_0651	L?DL_DatInFac	FacilityRcv(FacilityPdu_26(TI_01, facilityIErcv(FwdChAdvRslt_01)))	(P)	
		branchB			
33		L!DL_DatRqFac	FacilitySnd(TCV_chTch, FacilityPdu_25(TI_02, facilityIEtsnd(FwdChAdvSS_20)))		2.
		branchC			
34	L_0652	L?DL_DatInFac	FacilityRcv(FacilityPdu_26(TI_01, facilityIErcv(FwdChAdvRslt_01)))	(P)	
35		L?DL_DatInDisc	DiscRcv(TCV_chTch, DisconnR(TI_01, Cause_Def))		6.
		branchD			
36		L?DL_DatInDisc	DiscRcv(TCV_chTch, DisconnR(TI_01, Cause_Def))		6.
37	L_0653	L?DL_DatInFac	FacilityRcv(FacilityPdu_26(TI_01, facilityIErcv(FwdChAdvRslt_01)))	(P)	
		branchE			
38		L?DL_DatInDisc	DiscRcv(TCV_chTch, Disconn_10(TI_01))		6., with ReturnResult
		branchF			
39		L?DL_DatInDisc READTIMER T_dly(TCV_Time), CANCEL T_dly	DiscRcv(TCV_chTch, DisconnR(TI_01, Cause_Def))		3.
40		L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_08(TI_02))		
41		L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TI_01))		
42		+PostMainLinkRel(TCV_chTch)			
		branchG			
43		L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_08(TI_02))		
44		L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_25(TI_01))		

45		+PostMainLinkRel(TCV_chTch)			
		branchH			
46		?TIMEOUT T_dly			9.
47		(TCV_Null := OO_TermCall())			
48		L?DL_DatInDisc (TCV_Fn1 := DL_DatInDisc.fn)	DiscRcv(TCV_chTch, DisconnR(TI_01, Cause_Def))		3.
49		L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_08(TI_02))		
50		L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TI_01))		
51		+PostMainLinkRel(TCV_chTch)			
		step20(val:IA5String)			
52		(TCV_Res := OO_ACMIncCHK(val))			
53	L_0654	[TCV_Res]		(P)	
54	L_0655	[NOT TCV_Res]		(F)	
		TimerCheck			
55	L_0656	[(88000 <= TCV_Time) AND (TCV_Time <= 92000)]		(P)	4.
56	L_0657	[(88000 > TCV_Time) OR(TCV_Time > 92000)]		(F)	4.

Detailed Comments:

1. To reset the ACM = 0 and set the ACMmax to 2.
2. To send Facility IE of ForwardChargeAdvice using definite form mixed with indefinite form.
3. The call is terminated when the ACM reaches the ACMmax.
4. To check whether the time duration is 90 +- 2 seconds.
5. To check whether the ACM increment is 2.
6. The call is unsuccessful.
7. To send Facility IE of ForwardChargeAdvice with zero CAI.
8. To check whether the value of ACM is still 2.
9. The time duration is 120 seconds.

Test Case Dynamic Behaviour

Test Case Name: TC_31_8_1_1

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that the MS correctly requests a supplementary service transaction for registration of a password for all call restriction services in CHANNEL REQUEST message.

2) To check that the MS correctly requests a supplementary service transaction for registration of a password for all call restriction services in the subsequent CM-SERVICE REQUEST.

3) To check that the MS sends a REGISTER message containing the invoke of the RegisterPassword operation with the expected parameter values for registration of a password for all barring services.

4) To check that when the mobile subscriber wants to register a new password, the old password the new and the repeat of the new password shall be entered into the MS before the MS sends to the network a CHANNEL REQUEST..

5) To check that the MS is able to send a password by sending a FACILITY message in accordance to the user request (MMI actions).

6) To check that upon receipt of the result of the procedure, contained in RELEASE COMPLETE message, the MS provides the appropriate user indication (as described by the manufacturer).

These checks are done for:

all barring services, the result of the operation being sent in a RELEASE COMPLETE message

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_Null := OO_InitSS("**03*330*1234*9876*9876#"))			2.
4		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
5		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
6		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
7		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
8		ACTIVATE(OtherEventsFail)			Restore Normal default
9		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
10		+continue			
		continue			
11		L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, RegPasswdSS_01), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].registerPasswordComponents.registerPassword_InvokeComp.inv	Register_03(RegisterPdu_03(facilityIErcviei(RegPasswdSS_01)))		

		okeID, 1))			
12		L!DL_DatRqFac	FacilitySnd(TCV_ch, FacilityPdu_25_ci(TCV_TI, facilityIEtsnd(GetPasswdSS_01(TCV_InvkId))))		
13		L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI0, facilityIErcv(GetPasswdRslt_01)))		3
14		L!DL_DatRqFac	FacilitySnd(TCV_ch, FacilityPdu_25_ci(TCV_TI, facilityIEtsnd(GetPasswdSS_02(TCV_InvkId))))		
15		L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI0, facilityIErcv(GetPasswdRslt_02)))		4
16		L!DL_DatRqFac	FacilitySnd(TCV_ch, FacilityPdu_25_ci(TCV_TI, facilityIEtsnd(GetPasswdSS_03(TCV_InvkId))))		
17	L_0658	L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI0, facilityIErcv(GetPasswdRslt_02)))	(P)	5
18		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(RegPasswdSSRslt_01(TCV_InvkId))))		
19		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To set up physical channel as BCCH, CCCH and SDCCH4.
2. To initiate the Registration of password supplementary service.
3. Received the old password.
4. Received the new password.
5. Received the repeat of new password.

Test Case Dynamic Behaviour

Test Case Name: TC_31_8_1_2_1

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that, when a call transaction is already established, the MS correctly requests the establishment of a parallel MM transaction for supplementary service transaction of registration of a password for all call restriction services, sending a CM-SERVICE REQUEST.

2) To check that the MS sends a REGISTER message containing the invoke of the RegisterPassword operation with the expected parameter values for registration of a password for all call restriction services.

3) To check that upon receipt of the RELEASE COMPLETE message related to the present SS transaction, the first transaction remains unaffected.

4) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (as described by the Manufacturer).

Those checks are performed with a call transaction already established for:

all call restriction services, the RELEASE COMPLETE message being sent at the beginning of the procedure with a facility IE containing a return_error(error) where error is "SS subscription violation".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
6		(TCV_Null := OO_InitSS("**03*330*1234*9876*9877#"))			3.
7		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
8		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
9		+localtree			
		localtree			
10	L_0659	L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, RegPasswdSS_01), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].registerPasswordComponents.registerPassword_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(RegPasswdSS_01)))	(P)	
11		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch,		4.

			ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(RegPasswdSSErr_01 (TCV_InvkId))))		
12		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
13	L_0660	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
14		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To set up traffic channel.
2. To establish a mobile originating call, bring the MS into state U10.
3. To initiate RegisterPassword supplementary service.
4. To reject the RegisterPassword supplementary service invocation.

Test Case Dynamic Behaviour

Test Case Name: TC_31_8_1_2_2

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that, when a call transaction is already established, the MS correctly requests the establishment of a parallel MM transaction for supplementary service transaction of registration of a password for all call restriction services, sending a CM-SERVICE REQUEST.

2) To check that the MS sends a REGISTER message containing the invoke of the RegisterPassword operation with the expected parameter values for registration of a password for all call restriction services.

3) To check that upon receipt of the RELEASE COMPLETE message related to the present SS transaction, the first transaction remains unaffected.

4) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (as described by the Manufacturer).

Those checks are performed with a call transaction already established for :

all call restriction services, the RELEASE COMPLETE message being sent at the end of the procedure with a facility IE containing a return_error(error) where error is "NegativePasswordCheck".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcC, TSPX_MOChRateC)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
6	body	(TCV_Null := OO_InitSS("**03*330*1234*9876*9876#"))			3.
7		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
8		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
9		+continue			
		continue			
10		L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, RegPasswdSS_01), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].registerPasswordComponents.registerPassword_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(RegPasswdSS_01)))		
11		L!DL_DatRqFac	FacilitySnd(TCV_chTch, FacilityPdu_25_ci(TCV_TI2,		

			facilityIEtsnd(GetPasswdSS_01(TCV_InvkId)))		
12		L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI1, facilityIErcv(GetPasswdRslt_01)))		4
13		L!DL_DatRqRelCmp	RelComSnd(TCV_ch Tch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(RegPasswdSSErr_02 (TCV_InvkId))))		5.
14		+MMI_indic_chk			
15		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
16	L_0661	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
17		+PostMainLinkRel(TCV_chTch)			
		MMI_indic_chk			
18		(TCV_Res := OO_SSresultCHK(C_RegPswd))			
19	L_0662	[TCV_Res]		(P)	
20	L_0663	[NOT TCV_Res]		(F)	

Detailed Comments:

1. To set up traffic channel.
2. To establish a mobile originating call, bring the MS into state U10.
3. To initiate the Registration of password supplementary service.
4. Received the wrong password.
5. Negative password check, indefinite form.
6. Check whether the MS provides correct MMI user indication.

Test Case Dynamic Behaviour

Test Case Name: TC_31_8_1_2_3

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that, when a call transaction is already established, the MS correctly requests the establishment of a parallel MM transaction for supplementary service transaction of registration of a password for all call restriction services, sending a CM-SERVICE REQUEST.

2) To check that the MS sends a REGISTER message containing the invoke of the RegisterPassword operation with the expected parameter values for registration of a password for all call restriction services.

3) To check that upon receipt of the RELEASE COMPLETE message related to the present SS transaction, the first transaction remains unaffected.

4) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (as described by the Manufacturer).

Those checks are performed with a call transaction already established for :

all call restriction services, the RELEASE COMPLETE message being sent at the end of the procedure with a facility IE containing a return_error(error) where error is "PasswordRegistrationFailure" with diagnostic "new password mismatch".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcB, TSPX_MOChRateB)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
6		(TCV_Null := OO_InitSS("**03*330*1234*9876*9877#"))			3.
7		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
8		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
9		+continue			
		continue			
10		L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, RegPasswdSS_01), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].registerPasswordComponents.registerPassword_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(RegPasswdSS_01)))		
11		L!DL_DatRqFac	FacilitySnd(TCV_chTch, FacilityPdu_25_ci(		

			TCV_TI2, facilityIEtsnd(GetPasswdSS_01(TCV_InvkId))))		
12		L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI1, facilityIErcv(GetPasswdRslt_01)))		4.
13		L!DL_DatRqFac	FacilitySnd(TCV_chTch, FacilityPdu_25_ci(TCV_TI2, facilityIEtsnd(GetPasswdSS_02(TCV_InvkId))))		
14		L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI1, facilityIErcv(GetPasswdRslt_02)))		5.
15		L!DL_DatRqFac	FacilitySnd(TCV_chTch, FacilityPdu_25_ci(TCV_TI2, facilityIEtsnd(GetPasswdSS_03(TCV_InvkId))))		
16		L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI1, facilityIErcv(GetPasswdRslt_03)))		6.
17		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(RegPasswdSSErr_03(TCV_InvkId))))		7.
18		+MMI_indic_chk			
19		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
20	L_0664	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
21		+PostMainLinkRel(TCV_chTch)			
		MMI_indic_chk			
22		(TCV_Res := OO_SSresultCHK(C_RegPswd))			
23	L_0665	[TCV_Res]		(P)	
24	L_0666	[NOT TCV_Res]		(F)	

Detailed Comments:

1. To set up traffic channel.
2. To establish a mobile originating call, bring the MS into state U10.
3. To initiate the Registration of password supplementary service.
4. Received the old password.
5. Received the new password.
6. TReceived a different new password.
7. New password mismatch.

Test Case Dynamic Behaviour

Test Case Name: TC_31_8_3_1

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that the MS correctly requests a supplementary service transaction for activation of a specific call restriction service in CHANNEL REQUEST message.

2) To check that the MS correctly requests a supplementary service transaction for activation of call restriction service in the subsequent CM-SERVICE REQUEST.

3) To check that the MS sends a REGISTER message containing the invoke of the ActivateSS operation with the expected parameter values for activation of a specific call restriction service.

4) To check that upon receipt of FACILITY message requiring the password, the MS provides the appropriate user indication (as described by the manufacturer).

5) To check that the MS is able to send a password by sending a FACILITY message in accordance to the user request (MMI actions).

6) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (which is to be described by the manufacturer).

These checks are done for:

a) BAOC, for basic service group "all synchronous services" the result of the operation being sent in a FACILITY message.

b) BICRoam, for all basic service groups, the result of the operation being sent in a RELEASE COMPLETE message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_Null := OO_InitSS("*33**22#"))			2.
4		+part1			
5		(TCV_Null := OO_InitSS("*351#"))			3.
6		+part2			
		part1			
7	L_0667	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	
8		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
9		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
10		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
11		ACTIVATE(OtherEventsFail)			Restore Normal default
12		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
13		+localtree			
		localtree			
14	L_066	L?DL_DatInRegister (TCV_TI :=	Register_03(	(P)	

	8	DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, ActivateSS_03), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].activateSSComponents.activateSS_InvokeComp.invokeID, 1))	RegisterPdu_03(facilityIErcviei(ActivateSS_03)))		
15		L!DL_DatRqFac	FacilitySnd(TCV_ch, FacilityPdu_25_ci(TCV_TI, facilityIEtsnd(GetPasswdSS_01(TCV_InvkId))))		4.
16		(TCV_Null := OO_EnterPswd("1234"))			5.
17	L_0669	L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI0, facilityIErcv(GetPasswdRslt_01)))	(P)	
18		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(ActivateSSRslt_03(TCV_InvkId))))		
19		+Checktree(C_ActBOAC)			
20		+PostMainLinkRel(TCV_ch)			
		part2			
21	L_0670	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	
22		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
23		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
24		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
25		ACTIVATE(OtherEventsFail)			Restore Normal default
26		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
27		+localtree1			
		localtree1			
28	L_0671	L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, ActivateSS_04), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].activateSSComponents.activateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(ActivateSS_04)))	(P)	
29		L!DL_DatRqFac	FacilitySnd(TCV_ch, FacilityPdu_25_ci(TCV_TI, facilityIEtsnd(GetPasswdSS_01(TCV_InvkId))))		4.
30		+Checktree(C_RegPswd)			
31		(TCV_Null := OO_EnterPswd("1234"))			5.
32		L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI0, facilityIErcv(GetPasswdRslt_01)))		
33		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(		

			ActivateSSRslt_04(TCV_InvkId))))		
34		+Checktree(C_ActBICRoam)			
35		+PostMainLinkRel(TCV_ch)			
		Checktree(par:INTEGER)			
36		(TCV_Res := OO_SSresultCHK(par))			
37	L_0672	[TCV_Res]		(P)	
38	L_0673	[NOT TCV_Res]		(F)	

Detailed Comments:

1. To setup physical channel as BCCH, CCCH and SDCCH4.
2. To initiate Activation for BAOC.
3. To initiate Activation for BICRoam.
4. To send GetPassword invocation to the MS.
5. To enter password at the MMI.

Test Case Dynamic Behaviour

Test Case Name: TC_31_8_3_2_1

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that, when a call transaction is already established, the MS correctly requests the establishment of a parallel MM transaction for supplementary service transaction of specific call barring service, sending a CM-SERVICE REQUEST.

2) To check that the MS sends a REGISTER message containing the invoke of the ActivateSS operation with the expected parameter values for specific call barring service.

3) To check that upon receipt of the RELEASE COMPLETE message related to the present SS transaction, the first transaction remains unaffected.

4) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (as described by the Manufacturer).

Those checks are performed with a call transaction already established for :

BOIC, the RELEASE COMPLETE message being sent at the beginning of the procedure with a facility IE containing a return_error(error) where error is "SS subscription violation".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcD, TSPX_MOChRateD)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
6		(TCV_Null := OO_InitSS("*331#"))			3.
7		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
8		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
9		+localtree			
		localtree			
10	L_0674	L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, ActivateSS_05), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].activateSSComponents.activateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(ActivateSS_05)))	(P)	
11		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(ActivateSSErr_01(		4.

			TCV_InvkId))))		
12		(TCV_Res := OO_SSresultCHK(C_ActBOIC))			
13	L_0675	[TCV_Res]		(P)	5.
14		+localtree1			
15	L_0676	[NOT TCV_Res]		(F)	
16		+localtree1			
		localtree1			
17		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
18	L_0677	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
19		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup traffic channel.
2. To establish a mobile originating call to bring the MS into state U10.
3. To initiate the ActivateSS for BOIC.
4. To send ReturnError for the invocation of ActivateSS.
5. The user indication is correct, pass.

Test Case Dynamic Behaviour

Test Case Name: TC_31_8_3_2_2

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that, when a call transaction is already established, the MS correctly requests the establishment of a parallel MM transaction for supplementary service transaction of activation of one specific call restriction service, sending a CM-SERVICE REQUEST.

2) To check that the MS sends a REGISTER message containing the invoke of the ActivateSS operation with the expected parameter values for activation of one specific call restriction service.

3) To check that upon receipt of the RELEASE COMPLETE message related to the present SS transaction, the first transaction remains unaffected.

4) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (as described by the Manufacturer).

Those checks are performed with a call transaction already established for :

BAIC, the RELEASE COMPLETE message being sent at the end of the procedure with a facility IE containing a return_error(error) where error is "NegativePasswordCheck".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcE, TSPX_MOChRateE)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
6		(TCV_Null := OO_InitSS("*35#"))			3.
7	L_0678	L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)	(P)	
8		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
9		+localtree			
		localtree			
10	L_0679	L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, ActivateSS_06), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].activateSSComponents.activateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(ActivateSS_06)))	(P)	
11		L!DL_DatRqFac	FacilitySnd(TCV_chTch, FacilityPdu_25_ci(TCV_TI2, facilityIEtsnd(GetPasswdSS_01(		

			TCV_InvkId))))		
12		(TCV_Res := OO_SSresultCHK(C_RegPswd))			
13	L_0680	[TCV_Res]		(P)	
14		+localtree1			
15	L_0681	[NOT TCV_Res]		(F)	
16		+localtree1			
		localtree1			
17		(TCV_Null := OO_EnterPswd("1234"))			4.
18		L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI1, facilityIErcv(GetPasswdRslt_01)))		
19		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(ActivateSSErr_02(TCV_InvkId))))		5.
20		(TCV_Res := OO_SSresultCHK(C_ActBAIC))			
21	L_0682	[TCV_Res]		(P)	6.
22		+localtree2			
23	L_0683	[NOT TCV_Res]		(F)	
24		+localtree2			
		localtree2			
25		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
26	L_0684	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
27		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup traffic channel.
2. To establish a mobile originating call, bring the MS into state U10.
3. To initiate the Activation supplementary service.
4. To enter password.
5. ReturnError indication negative password checking.
6. The user indication is correct, pass.

Test Case Dynamic Behaviour

Test Case Name: TC_31_8_4_1

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that the MS correctly requests a supplementary service transaction for deactivation of a group of call barring services in CHANNEL REQUEST message.

2) To check that the MS correctly requests a supplementary service transaction for deactivation of a group of call barring services in the subsequent CM-SERVICE REQUEST.

3) To check that the MS sends a REGISTER message containing the invoke of the DeactivateSS operation with the expected parameter values for deactivation of a group of call restriction services.

4) To check that upon receipt of FACILITY message requiring the password, the MS provides the appropriate user indication (as described by the manufacturer).

5) To check that the MS is able to send a password by sending a FACILITY message in accordance to the user request (MMI actions).

6) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (which is to be described by the manufacturer).

These checks are done for:

a) all restrictions, for basic service group "speech".

b) barring of outgoing calls, for all facsimile.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_Null := OO_InitSS("#330**11#"))			2.
4		+part1			
5		(TCV_Null := OO_InitSS("#333**13#"))			3.
6		+part2			
		part1			
7	L_0685	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	
8		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
9		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
10	L_0686	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)	(P)	
11		ACTIVATE(OtherEventsFail)			Restore Normal default
12		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
13		+localtree			
		localtree			
14	L_0687	L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI,	Register_03(RegisterPdu_03(	(P)	

		TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, DeactivateSS_03), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].deactivateSSComponents.deactivateSS_InvokeComp.invokeID, 1))	facilityIErcviei(DeactivateSS_03)))		
15		L!DL_DatRqFac	FacilitySnd(TCV_ch, FacilityPdu_25_ci(TCV_TI, facilityIEtsnd(GetPasswdSS_01(TCV_InvkId))))		
16		(TCV_Res := OO_SSresultCHK(C_RegPswd))			
17	L_0688	[TCV_Res]		(P)	
18		+localtree1			
19	L_0689	[NOT TCV_Res]		(F)	
20		+localtree1			
		localtree1			
21		(TCV_Null := OO_EnterPswd("1234"))			4.
22		L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI0, facilityIErcv(GetPasswdRslt_01)))		
23		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(DeactivateSSRslt_03(TCV_InvkId))))		
24		+PostMainLinkRel(TCV_ch)			
		part2			
25	L_0690	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	
26		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
27		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
28	L_0691	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)	(P)	
29		ACTIVATE(OtherEventsFail)			Restore Normal default
30		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
31		+localtree2			
		localtree2			
32	L_0692	L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, DeactivateSS_04), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].deactivateSSComponents.deactivateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(DeactivateSS_04)))	(P)	
33		L!DL_DatRqFac	FacilitySnd(TCV_ch, FacilityPdu_25_ci(TCV_TI, facilityIEtsnd(GetPasswdSS_01(TCV_InvkId))))		
34		(TCV_Res := OO_SSresultCHK(C_RegPswd))			
35	L_0693	[TCV_Res]		(P)	

36		+localtree3			
37	L_0694	[NOT TCV_Res]		(F)	
38		+localtree3			
		localtree3			
39		(TCV_Null := OO_EnterPswd("1234"))			
40	L_0695	L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI0, facilityIErcv(GetPasswdRslt_01)))	(P)	
41		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(DeactivateSSRslt_04(TCV_InvkId))))		
42		(TCV_Res := OO_SSresultCHK(C_DeactBO))			
43	L_0696	[TCV_Res]		(P)	
44		+PostMainLinkRel(TCV_ch)			
45	L_0697	[NOT TCV_Res]		(F)	
46		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To setup physical channel as BCCH, CCCH and SDCCH4.
2. To initiate Deactivation for B.
3. To initiate Deactivation for BO.
4. To send GetPassword invocation to the MS.
5. To enter password at the MMI.

Test Case Dynamic Behaviour

Test Case Name: TC_31_8_4_2_1

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that, when a call transaction is already established, the MS correctly requests the establishment of a parallel MM transaction for supplementary service transaction of a group of call barring services, sending a CM-SERVICE REQUEST.

2) To check that the MS sends a REGISTER message containing the invoke of the DeactivateSS operation with the expected parameter values for a group of call barring services.

3) To check that upon receipt of the RELEASE COMPLETE message related to the present SS transaction, the first transaction remains unaffected.

4) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (as described by the Manufacturer).

These checks are performed with a call transaction already established for :

BI, the RELEASE COMPLETE message being sent at the beginning of the procedure with a facility IE containing a return_error(error) where error is "SS subscription violation".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcF, TSPX_MOChRateF)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
6		(TCV_Null := OO_InitSS("#353#"))			3.
7		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
8		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
9		+localtree			
		localtree			
10	L_0698	L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, DeactivateSS_05), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].deactivateSSComponents.deactivateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(DeactivateSS_05)))	(P)	
11		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2,		4.

			facilityIEtsndiei(DeactivateSSErr_01(TCV_InvkId)))		
12		(TCV_Res := OO_SSresultCHK(C_DeactBI))			
13	L_0699	[TCV_Res]		(P)	5.
14		+localtree1			
15	L_0700	[NOT TCV_Res]		(F)	
16		+localtree1			
		localtree1			
17		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
18	L_0701	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
19		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup traffic channel.
2. To establish a mobile originating call to bring the MS into state U10.
3. To initiate the DeactivateSS for BI.
4. To send ReturnError for the invocation of DeactivateSS.
5. The user indication is correct, pass.

Test Case Dynamic Behaviour

Test Case Name: TC_31_8_4_2_2

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that, when a call transaction is already established, the MS correctly requests the establishment of a parallel MM transaction for supplementary service transaction of deactivation of a group of call restriction services, sending a CM-SERVICE REQUEST.

2) To check that the MS sends a REGISTER message containing the invoke of the DeactivateSS operation with the expected parameter values for deactivation of a group of call restriction service.

3) To check that upon receipt of the RELEASE COMPLETE message related to the present SS transaction, the first transaction remains unaffected.

4) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (as described by the Manufacturer).

Those checks are performed with a call transaction already established for :

BOICExHC, the RELEASE COMPLETE message being sent at the end of the procedure with a facility IE containing a return_error(error) where error is "NegativePasswordCheck".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcG, TSPX_MOChRateG)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
6		(TCV_Null := OO_InitSS("#332#"))			3.
7	L_0702	L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)	(P)	
8		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
9		+localtree			
		localtree			
10	L_0703	L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, DeactivateSS_06), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].deactivateSS Components.deactivateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(DeactivateSS_06)))	(P)	
11		L!DL_DatRqFac	FacilitySnd(TCV_chTch, FacilityPdu_25_ci(TCV_TI2, facilityIEtsnd(GetPasswdSS_01(		

			TCV_InvkId))))		
12		(TCV_Res := OO_SSresultCHK(C_RegPswd))			
13	L_0704	[TCV_Res]		(P)	
14		+localtree1			
15	L_0705	[NOT TCV_Res]		(F)	
16		+localtree1			
		localtree1			
17		(TCV_Null := OO_EnterPswd("1234"))			4.
18		L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI1, facilityIErcv(GetPasswdRslt_01)))		
19		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(DeactivateSSErr_02(TCV_InvkId))))		5.
20		(TCV_Res := OO_SSresultCHK(C_DeactBOICExHC))			
21	L_0706	[TCV_Res]		(P)	6.
22		+localtree2			
23	L_0707	[NOT TCV_Res]		(F)	
24		+localtree2			
		localtree2			
25		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
26	L_0708	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
27		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup traffic channel.
2. To establish a mobile originating call, bring the MS into state U10.
3. To initiate the deactivation supplementary service.
4. To enter password.
5. ReturnError indication negative password checking.
6. The user indication is correct, pass.

Test Case Dynamic Behaviour

Test Case Name: TC_31_8_6_1

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that the MS correctly requests a supplementary service transaction for interrogation of a specific call barring service in CHANNEL REQUEST message.

2) To check that the MS correctly requests a supplementary service transaction for interrogation of a call barring service in the subsequent CM-SERVICE REQUEST.

3) To check that the MS sends a REGISTER message containing the invoke of the InterrogateSS operation with the expected parameter values for interrogation of one call restriction service.

4) To check that upon receipt of FACILITY message requiring the password, the MS provides the appropriate user indication (as described by the manufacturer).

5) To check that the MS is able to send a password by sending a FACILITY message in accordance to the user request (MMI actions).

6) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (which is to be described by the manufacturer).

These checks are done for :

a) BAIC, the result of the operation being a Basic Service code sent in a FACILITY message.

b) BOICExHC, the result of the operation being a SS-status sent in a RELEASE COMPLETE message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_Null := OO_InitSS("*#35#"))			2.
4		+part1			
5		(TCV_Null := OO_InitSS("*#332#"))			3.
6		+part2			
		part1			
7	L_0709	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	
8		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
9		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
10	L_0710	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)	(P)	
11		ACTIVATE(OtherEventsFail)			Restore Normal default
12		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
13		+localtree			
		localtree			
14	L_071	L?DL_DatInRegister (TCV_TI :=	Register_03(	(P)	

	1	DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, InterrogateSS_07), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].interrogateSSComponents.interrogateSS_InvokeComp.invokeID, 1))	RegisterPdu_03(facilityIErcviei(InterrogateSS_07)))		
15		L!DL_DatRqFac	FacilitySnd(TCV_ch, FacilityPdu_25_ci(TCV_TI, facilityIEtsnd(GetPasswdSS_01(TCV_InvkId))))		
16		(TCV_Res := OO_SSresultCHK(C_RegPswd))			
17	L_0712	[TCV_Res]		(P)	
18		+localtree1			
19	L_0713	[NOT TCV_Res]		(F)	
20		+localtree1			
		localtree1			
21		(TCV_Null := OO_EnterPswd("1234"))			4.
22		L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI0, facilityIErcv(GetPasswdRslt_01)))		
23		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(InterrogateSSRslt_03(TCV_InvkId))))		
24		+PostMainLinkRel(TCV_ch)			
		part2			
25	L_0714	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	
26		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
27		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
28	L_0715	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)	(P)	
29		ACTIVATE(OtherEventsFail)			Restore Normal default
30		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
31		+localtree2			
		localtree2			
32	L_0716	L?DL_DatInRegister (TCV_TI := DL_DatInRegister.msg.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, InterrogateSS_08), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].interrogateSSComponents.interrogateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(InterrogateSS_08)))	(P)	
33		L!DL_DatRqFac	FacilitySnd(TCV_ch, FacilityPdu_25_ci(TCV_TI, facilityIEtsnd(GetPasswdSS_01(TCV_InvkId))))		

34		(TCV_Res := OO_SSresultCHK(C_RegPswd))			
35	L_0717	[TCV_Res]		(P)	
36		+localtree3			
37	L_0718	[NOT TCV_Res]		(F)	
38		+localtree3			
		localtree3			
39		(TCV_Null := OO_EnterPswd("1234"))			
40	L_0719	L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI0, facilityIErcv(GetPasswdRslt_01)))	(P)	
41		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(InterrogateSSRslt_04(TCV_InvkId))))		
42		(TCV_Res := OO_SSresultCHK(C_InterrogBOICExHC))			
43	L_0720	[TCV_Res]		(P)	
44		+PostMainLinkRel(TCV_ch)			
45	L_0721	[NOT TCV_Res]		(F)	
46		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

1. To setup physical channel as BCCH, CCCH and SDCCH4.
2. To initiate Interrogation for BAIC.
3. To initiate Interrogation for BOICExHC.
4. To send GetPassword invocation to the MS.
5. To enter password at the MMI.

Test Case Dynamic Behaviour

Test Case Name: TC_31_8_6_2

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To check that the MS correctly requests the establishement of a parallel MM transaction for supplementary service transaction of interrogation of a specific call barring service message, sending a CM-SERVICE REQUEST.

2) To check that the MS sends a REGISTER message containing the invoke of the InterrogateSS operation with the expected parameter values for interrogation of call barring.

3) To check that upon receipt of the RELEASE COMPLETE message related to the present SS transaction, the first transaction remains unaffected.

4) To check that upon receipt of the RELEASE COMPLETE message, the MS provides the appropriate user indication (as described by the Manufacturer).

These checks are performed with a call transaction already established for :

a) BICRoam, the RELEASE COMPLETE message being sent with a facility IE containing a return_error(error) where error is "SS not available".

b) BOIC, the RELEASE COMPLETE message being sent with a facility IE containing a reject(invoke_problem) where invoke_problem is "resource limitation".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcH, TSPX_MOChRateH)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			1.
5		+Est_MO_Call(TimingAdv(0), TCV_ChRate, TSPX_RANDDef, TSPX_SDCCH4SubDef, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			2.
6		(TCV_Null := OO_InitSS("*#351#"))			3.
7		+part1			
8		(TCV_Null := OO_InitSS("*#331#"))			4.
9		+part2			
		part1			
10		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
11		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
12		+localtree			
		localtree			
13		L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n :=	Register_03(RegisterPdu_03(facilityIErcviei(InterrogateSS_05)))		

		OC_PosinSet(TCV_Comp, InterrogateSS_05), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].interrogateSSComponents.interrogateSS_InvokeComp.invokeID, 1))			
14		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(InterrogateSSErr_02(TCV_InvkId))))		
15		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
16	L_0722	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
		part2			
17		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
18		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
19		+localtree1			
		localtree1			
20		L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, InterrogateSS_06), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].interrogateSSComponents.interrogateSS_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcviei(InterrogateSS_06)))		
21		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(InterrogateSSRej_02(TCV_InvkId))))		
22		(TCV_Res := OO_SSresultCHK(C_InterrogBOIC))			
23	L_0723	[TCV_Res]		(P)	
24		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
25	L_0724	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
26		+PostMainLinkRel(TCV_chTch)			
27	L_0725	[NOT TCV_Res]		(F)	
28		L!DL_DatRqCcstEnq	CCStESnd(TCV_chTch, CCStatusEq_01(TCV_TI))		
29	L_0726	L?DL_DatInCcst	CCStRcv(TCV_chTch, CCStatus_14(TCV_TI0, C_U10))	(P)	
30		+PostMainLinkRel(TCV_chTch)			

Detailed Comments:

1. To setup traffic channel.
2. To establish a mobile originating call.
3. To initiate an interrogation for BICRoam.
4. To initiate an interrogation for BOIC.

Test Case Dynamic Behaviour

Test Case Name: TC_31_8_7

Group: GSM_L3_MS_v4220/SS/

Purpose: To check that upon receipt of the RELEASE COMPLETE message the MS provides the appropriate user indication (as described by the manufacturer).

This is tested in the case of barring of incoming calls.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcB, TSPX_MOChRateB)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		+InitCall(TCV_Service)			
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
9		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
12		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
13		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(NotificationSS_05)))		
14		(TCV_Res := OO_SSresultCHK(C_NotifyBI))			
15	L_0727	[TCV_Res]		(P)	
16		+PostMainLinkRel(TCV_ch)			
17	L_0728	[NOT TCV_Res]		(F)	
18		+PostMainLinkRel(TCV_ch)			

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_31_9_1_1

Group: GSM_L3_MS_v4220/SS/

Purpose: 1)To verify that the mobile station invokes an USSD request by sending a REGISTER message to the network containing a ProcessUnstructruredSS-Request invoke component. This message will contain the alphabet indicator set to "SMS default alphabet" and the lenguage indicator set to "language unspecified". The ussd-string parameter shall contain the following digits and symbols depending on the operation initiated:

Activation *NN(N)# (no supplementary information included)
*NN(N)*SI#(one field of supplementary information included)
*NN(N)*SIA*SIB#(two fields of supplementary information included)

Dectivation #NN(N)# (no supplementary information included)
#NN(N)*SI#(one field of supplementary information included)
#NN(N)*SIA*SIB#(two fields of supplementary information included)

Interrogation *#NN(N)# (no supplementary information included)
*#NN(N)*SI#(one field of supplementary information included)
*#NN(N)*SIA*SIB#(two fields of supplementary information included)

Registration **NN(N)# (no supplementary information included)
**NN(N)*SI#(one field of supplementary information included)
**NN(N)*SIA*SIB#(two fields of supplementary information included)

Erasure ##NN(N)# (no supplementary information included)
##NN(N)*SI#(one field of supplementary information included)
##NN(N)*SIA*SIB#(two fields of supplementary information included)

Operations not yet defined in GSM 02.30 (see 2)

NN(N) features a set of service codes which have not yet been allocated for GSM supplementary services (see GSM 02.30 for service codes already specified).
N is a digit within 1..9 and SI, SIA, SIB strings of characters.

2)To check that the entry of 1 or 2 characters defined in the GSM 03.38 default alphabet followed by "SEND" shall be interpreted by the MS as an USSD request unless the MS is not engaged in a call and the first of the two character entry followed by "SEND " is a "1".

3) To verify that, for supplementary service procedures independent of any call, the initating side must establish a MM-connection between the network and the mobile station according to the rules in TS GSM 4.07 and 4.08.

4) To verify that, within a call the MS shall transmit a USSD request if any. See TS GSM 4.07 and 4.08 for the hanndling of multiple MM connections.

5) To check that upon receipt of the RELEASE COMPLETE message, the MS shall display the information contained to the user in a way described by the manufacturer.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcC, C_Full)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			

4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		(TCV_PreviousOctets:= 'A2120201'O, TCV_FollowingOctets:= '300E02013B0FC6F038CD4ED3F391E712'O, TCV_counter_c:=1, TCV_counter_k:=1)			USSD String for FACILITY message (random): "Facility OK"
6		REPEAT ltree_c_loop UNTIL [TCV_counter_c>17]			
		ltree_c_loop			
7		+ltreeSetLoopParameters			
8		(TCV_Null:=OO_InitSS(TCV_UssdString))			
9		+ltree_MMConnection			
10		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
11		ACTIVATE(OtherEventsFail)			Restore Normal default
12		+ltree_UssdOperation(TCV_ch)			
13		+PostMainLinkRel(TCV_ch)			
14		+ltree_continue			
		ltree_continue			
15		+ltree_MsOrigCall			
16		(TCV_Null := OO_InitSS(TCV_UssdString))			
17		+DTMFSignalling(OC_LengthOfString(TCV_UssdString), TCV_TI0, TCV_TI, TCV_chTch)			
18		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
19		+ltree_UssdOperation(TCV_chTch)			
20		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
21		L?DL_DatInRel	ReleaseRcv(Release_02)		
22		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
23		+PostMainLinkRel(TCV_chTch)			
24		(TCV_counter_c := TCV_counter_c+1)			
		ltree_MMConnection			
25		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
26		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
27		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
		ltree_MsOrigCall			
28		+InitCall(TCV_Service)			
29		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
30		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
31		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot,		

			TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
32		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
33		ACTIVATE(OtherEventsFail)			Restore Normal default
34		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
35		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
36		L!DL_DatRqCallProc	CallProc(TCV_chTch, TCV_CallProc)		
37		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
38		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
39		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
		ltree_UssdOperation(ch: LOGICCH)			
40		L!DL_DatRqCmsAcp	CMSerAcp(ch, CMServiceAcp_01)		
41	L_0729	L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, ProcessUSSDReq_01(TCV_UssdString)), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].processUnstructuredSSRequestComponents.processUnstructuredSSRequest_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcv(ProcessUSSDReq_01(TCV_UssdString))))	(P)	
42		L!DL_DatRqRelCmp	RelComSnd(ch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(USSDReq_01(TCV_InvkId, TCV_PreviousOctets, TCV_FollowingOctets, TCV_UssdString))))		
		ltreeSetLoopParameters			
43		[TCV_counter_c=1]			
44		(TCV_UssdString:="*60#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			activation with no supplement. information
45		[TCV_counter_c=2]			
46		(TCV_UssdString:="*201*35#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			activation with one field of sup. information
47		[TCV_counter_c=3]			
48		(TCV_UssdString:="*70*635*562#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			activation with two fields of sup. information
49		[TCV_counter_c=4]			
50		(TCV_UssdString:="#60#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			deactivation with no supplement. information
51		[TCV_counter_c=5]			
52		(TCV_UssdString:="#201*35#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			deactivation with one field of sup. information
53		[TCV_counter_c=6]			
54		(TCV_UssdString:="#70*635*562#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			deactivation with two fields of sup. information

55		[TCV_counter_c=7]			
56		(TCV_UssdString:="*#60#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			interrogation with no supplement. information
57		[TCV_counter_c=8]			
58		(TCV_UssdString:="*#201*35#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			interrogation with one field of sup. information
59		[TCV_counter_c=9]			
60		(TCV_UssdString:="*#70*635*562#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			interrogation with two fields of sup. information
61		[TCV_counter_c=10]			
62		(TCV_UssdString:="**60#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			registration with no supplement. information
63		[TCV_counter_c=11]			
64		(TCV_UssdString:="**201*35#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			registration with one field of sup. information
65		[TCV_counter_c=12]			
66		(TCV_UssdString:="**70*635*562#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			registration with two fields of sup. information
67		[TCV_counter_c=13]			
68		(TCV_UssdString:="##60#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			erasure with no supplement. information
69		[TCV_counter_c=14]			
70		(TCV_UssdString:="##201*35#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			erasure with one field of sup. information
71		[TCV_counter_c=15]			
72		(TCV_UssdString:="##70*635*562#", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			erasure with two fields of sup. information
73		[TCV_counter_c=16]			
74		(TCV_UssdString:="7", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			acc. requirement 1a (GSM 11.10)
75		[TCV_counter_c=17]			
76		(TCV_UssdString:="26", TCV_PreviousOctets:="O, TCV_FollowingOctets:="O)			acc. requirement 1a (GSM 11.10)
77	L_0730	[TCV_counter_c>17]		I	
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_31_9_1_2

Group: GSM_L3_MS_v4220/SS/

Purpose: 1)To verify that if a mobile initiated USSD request using protocole version 2 is rejected by the network, and the reason for the rejection is indicated either by the problem code "unrecognized operation" or a cause "facility rejected", the mobile station shall assume that the network only supports protocole version 1 of USSD operation. The mobile station shall re-attempt the request by using the appropriate protocole version 1 USSD operation without a SS version indicator if the unstructured data entered by teh user can be coded as an IA5 string.

2)To check that, upon receipt of the RELEASE COMPLETE message, the MS shall provide the appropriate user indication (which is to be described by the manufacturer). If ussd-string information is included this shall be given to the user (in a way described by the manufacturer).

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMO(TSPX_MOBscSvcC, C_Full)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		(TCV_counter_k:=1, TCV_UssdString:="*70*635*562#")			
6	body	+ltree_part1			
7		+ltree_part2			
8		+ltree_part3			
		ltree_part1			
9		(TCV_Null:=OO_InitSS(TCV_UssdString))			
10		+ltree_MMConnection			
11		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
12		ACTIVATE(OtherEventsFail)			Restore Normal default
13		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
14	L_0731	L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, ProcessUSSDReq_01(TCV_UssdString)), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].processUnstructuredSSRequestComponents.processUnstructuredSSRequest_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcv(ProcessUSSDReq_01(TCV_UssdString))))	(P)	Invoke ProcessUnstructeredUSS-Request
15		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(ProcessUSSRequest_02(TCV_InvkId, TCV_PreviousOctets, TCV_FollowingOctets		Return result ProcessUnstructeredUSS-Data

			))))		
16		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
17		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
18	L_0732	L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, ProcessUSSData_01(TCV_UssdString)), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].processUnstructuredSSRequestComponents.processUnstructuredSSRequest_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcv(ProcessUSSData_01(TCV_UssdString))))	(P)	Invoke ProcessUnstructeredUSS-Data
19		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(ProcessUSSData_02(TCV_InvkId, TCV_PreviousOctets, TCV_FollowingOctets))))		Return result ProcessUnstructeredUSS-Data
20		+PostMainLinkRel(TCV_ch)			
		ltree_part2			
21		(TCV_Null:=OO_InitSS(TCV_UssdString))			
22		+ltree_MMConnection			
23		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_08)		
24		ACTIVATE(OtherEventsFail)			Restore Normal default
25		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
26	L_0733	L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, ProcessUSSDReq_01(TCV_UssdString)), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].processUnstructuredSSRequestComponents.processUnstructuredSSRequest_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcv(ProcessUSSDReq_01(TCV_UssdString))))	(P)	Invoke ProcessUnstructeredUSS-Request
27		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_42(TCV_TI2))		Facility rejected
28		+PostMainLinkRel(TCV_ch)			
29		+ltree_MMConnection			
30		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
31		ACTIVATE(OtherEventsFail)			Restore Normal default
32		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
33	L_0734	L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, ProcessUSSData_01(TCV_UssdString)), TCV_InvkId := OC_Asn1intToOct(TCV_Com	Register_03(RegisterPdu_03(facilityIErcv(ProcessUSSData_01(TCV_UssdString))))	(P)	Invoke ProcessUnstructeredUSS-Data

		p.[TCV_n].processUnstructuredSSRequestComponents.processUnstructuredSSRequest_InvokeComp.invokeID, 1))			
34		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(ProcessUSSData_02(TCV_InvkId, TCV_PreviousOctets, TCV_FollowingOctets))))		Return result ProcessUnstructeredUSS-Data
35		+PostMainLinkRel(TCV_ch)			
		ltree_part3			
36		REPEAT ltree_k_loop UNTIL [TCV_counter_k>10]			
		ltree_k_loop			
37		+ltreeSetLoopParameters			
38		+ltree_MsOrigCall			
39		(TCV_Null:=OO_InitSS(TCV_UssdString))			
40		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_08)		
41		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
42	L_0735	L?DL_DatInRegister (TCV_TI1 := DL_DatInRegister.msg.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_Comp := DL_DatInRegister.msg.fie.components_1, TCV_n := OC_PosinSet(TCV_Comp, ProcessUSSDReq_01(TCV_UssdString)), TCV_InvkId := OC_Asn1intToOct(TCV_Comp.[TCV_n].processUnstructuredSSRequestComponents.processUnstructuredSSRequest_InvokeComp.invokeID, 1))	Register_03(RegisterPdu_03(facilityIErcv(ProcessUSSDReq_01(TCV_UssdString))))	(P)	Invoke ProcessUnstructeredUSS-Request
43		+ltree_ReleaseCmpAccCounterK			
44		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI2, Cause_01, ProgInd_omit, UuInfo_omit))		
45		L?DL_DatInRel	ReleaseRcv(Release_02)		
46		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI2))		
47		+PostMainLinkRel(TCV_chTch)			
48		(TCV_counter_k:= TCV_counter_k+1)			
		ltree_ReleaseCmpAccCounterK			
49		[TCV_counter_k<5]			
50		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(ProcessUSSRequest_02(TCV_InvkId, TCV_PreviousOctets, TCV_FollowingOctets))))		Return error ProcessUnstructeredUSS-Request
51		[TCV_counter_k>4]			
52		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch,		Reject ProcessUnstructeredUSS

			ReleaseCmp_09(TCV_TI2, facilityIEtsndiei(ProcessUSSRequest _02(TCV_InvkId, TCV_PreviousOctets, TCV_FollowingOctets)))		-Request
		ltree_MMConnection			
53		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
54		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
55		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
		ltree_MsOrigCall			
56		+InitCall(TCV_Service)			
57		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
58		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
59		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_nfh(TCV_Rr, TCV_Fn, TCV_chtype, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), TCV_tch_arfcn, C_normal_paging))		
60		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
61		ACTIVATE(OtherEventsFail)			Restore Normal default
62		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
63		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
64		L!DL_DatRqCallProc	CallProc(TCV_chTch, TCV_CallProc)		
65		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
66		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
67		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
		ltreeSetLoopParameters			
68		[TCV_counter_k=1]			
69		(TCV_PreviousOctets:='A3060201'O, TCV_FollowingOctets:='020122'O)			RETURN ERR Error code system failure
70		[TCV_counter_k=2]			
71		(TCV_PreviousOctets:='A3060201'O, TCV_FollowingOctets:='020123'O)			RETURN ERR Error code data missing
72		[TCV_counter_k=3]			
73		(TCV_PreviousOctets:='A3060201'O, TCV_FollowingOctets:='020147'O)			RETURN ERR Error code unknown alpabet
74		[TCV_counter_k=4]			
75		(TCV_PreviousOctets:='A3060201'O, TCV_FollowingOctets:='020124'O)			RETURN ERR Error code unexpected

					data value
76		[TCV_counter_k=5]			
77		(TCV_PreviousOctets:='A4060201'O, TCV_FollowingOctets:='800100'O)			REJECT Gen. Problem unrecognized component
78		[TCV_counter_k=6]			
79		(TCV_PreviousOctets:='A4060201'O, TCV_FollowingOctets:='800101'O)			REJECT Gen. Problem Mistyped component
80		[TCV_counter_k=7]			
81		(TCV_PreviousOctets:='A4060201'O, TCV_FollowingOctets:='800102'O)			REJECT Gen. Problem badly structured component
82		[TCV_counter_k=8]			
83		(TCV_PreviousOctets:='A4060201'O, TCV_FollowingOctets:='800102'O)			REJECT Invoke Problem Mistyped parameter
84		[TCV_counter_k=9]			
85		(TCV_PreviousOctets:='A4060201'O, TCV_FollowingOctets:='800103'O)			REJECT Invoke Problem resource limitation
86		[TCV_counter_k=10]			
87		(TCV_PreviousOctets:='A4060201'O, TCV_FollowingOctets:='800104'O)			REJECT Invoke Problem initiating release
88	L_0736	[TCV_counter_k>10]		I	
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_31_9_2_1

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To verify that for a USSD request, the MS shall display the text provided and await user input. If the user enters a response, the MS shall acknowledge the operation by sending a FACILITY message containing an empty result component to the network

2) To verify that the MS includes alphabet and language indicators in the response to the network. The alphabet indicator shall indicate "SMS default alphabet". The language indicator shall indicate "language unspecified".

3)To check that the MS is able to process the opeation during a call or out of a call.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcF, C_Full)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+CCConfigTCH_nociph(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, TCV_ChMod.mode, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd)			
5		(TCV_UssdString:="Transaction OK")			
6	body	+ltree_body			
		ltree_body			
7		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
8		L!DL_DatRqRegister(TCV_TI.ti_v:='100'B, TCV_TI.ti_f:='0'B, TCV_TI0.ti_v:='100'B, TCV_TI0.ti_f:='1'B, TCV_InvkId:='00'O)	RegisterReq(TCV_ch, RegisterPdu_34(TCV_TI, facilityIEtsnd(NotificationSS_06(TCV_InvkId, TCV_UssdString))))		
9		+CheckUssdStringDisplayed(TCV_UssdString)			
10	L_0737	L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI0, facilityIErcv(NotificationSS_07(TCV_InvkId))))	(P)	
11		+CC_EstMsTermCall(TCV_ChRate, TSPX_TCHHSubDef)			
12		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_47(TCV_TI))		
13		L!DL_DatRqRegister(TCV_TI2.ti_v:='001'B, TCV_TI2.ti_f:='0'B, TCV_TI1.ti_v:='001'B, TCV_TI1.ti_f:='1'B, TCV_InvkId:='01'O)	RegisterReq(TCV_chTch, RegisterPdu_34(TCV_TI2, facilityIEtsnd(NotificationSS_06(TCV_InvkId, TCV_UssdString))))		
14		+CheckUssdStringDisplayed(TCV_UssdString)			

15	L_0738	L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI1, facilityIErcv(NotificationSS_07(TCV_InvkId))))	(P)	
16		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_47(TCV_TI2))		
17		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TI_02, Cause_01, ProgInd_omit, UuInfo_omit))		
18		L?DL_DatInRel	ReleaseRcv(Release_02)		
19		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TI_02))		
20		+ChanRel_end(TCV_chTch)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_31_9_2_2

Group: GSM_L3_MS_v4220/SS/

Purpose: To verify that when the mobile station receives an USSD operation in parallel to any call independent supplementary transaction, it responds with a return error component in a RELEASE COMPLETE message, containing "USSD-Busy" error.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_UssdString:="Transaction OK")			
4	body	+ltree_body			
		ltree_body			
5		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
6		L!DL_DatRqRegister(TCV_TI.ti_v:='000'B, TCV_TI.ti_f:='0'B, TCV_TI0.ti_v:='000'B, TCV_TI0.ti_f:='1'B, TCV_InvkId0:='00'O)	RegisterReq(TCV_ch, RegisterPdu_34(TCV_TI, facilityIEtsnd(NotificationSS_06(TCV_InvkId0, TCV_UssdString))))		
7		+CheckUssdStringDisplayed(TCV_UssdString)			
8	L_0739	L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI0, facilityIErcv(NotificationSS_07(TCV_InvkId0))))	(P)	
9		L!DL_DatRqRegister(TCV_TI2.ti_v:='001'B, TCV_TI2.ti_f:='0'B, TCV_TI1.ti_v:='001'B, TCV_TI1.ti_f:='1'B, TCV_InvkId1:='01'O)	RegisterReq(TCV_ch, RegisterPdu_34(TCV_TI2, facilityIEtsnd(NotificationSS_06(TCV_InvkId1, TCV_UssdString))))		
10	L_0740	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_49(TCV_TI1, TCV_InvkId1))	(P)	
11		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(NotificationSS_08(TCV_InvkId0))))		
12		+ChanRel_end(TCV_ch)			

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_31_9_2_3

Group: GSM_L3_MS_v4220/SS/

Purpose: 1) To verify that for a USSD notification, the MS shall display the text provided and await user input. If the user enters a response, the MS shall return the response to the network, maintaining the transaction.

2) To verify that the MS includes alphabet and language indicators in the response to the network. The alphabet indicator shall indicate "SMS default alphabet". The language indicator shall indicate "language unspecified".

3)To check that the MS is able to process the opeation during a call or out of a call.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+BasicServiceMT(TSPX_MTBscSvcG, C_Full)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
4		+FullRateCh_A_1_nociph(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH(C_arfcn_tchA), FreqTCH(C_arfcn_tchAd), TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
5		(TCV_UssdString1 := "Type *70*635*562# and send", TCV_UssdString2 := "*70*635*562#")			
6	body	+ltree_body			
		ltree_body			
7		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
8		L!DL_DatRqRegister(TCV_TI.ti_v:='010'B, TCV_TI.ti_f:='0'B, TCV_TI0.ti_v:='010'B, TCV_TI0.ti_f:='1'B, TCV_InvkId:='00'O)	RegisterReq(TCV_ch, RegisterPdu_34(TCV_TI, facilityIEtsnd(USSDReq_03(TCV_InvkId, TCV_UssdString1))))		
9		+CheckUssdStringDisplayed(TCV_UssdString1)			
10		(TCV_Null:=OO_InitSS(TCV_UssdString2))			
11	L_0741	L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI0, facilityIErcv(USSDReq_04(TCV_InvkId, TCV_UssdString2))))	(P)	
12		+CC_EstMsTermCall(C_Full, TSPX_TCHHSubDef)			
13		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_47(TCV_TI))		
14		L!DL_DatRqRegister(TCV_TI2.ti_v:='001'B, TCV_TI2.ti_f:='0'B, TCV_TI1.ti_v:='001'B, TCV_TI1.ti_f:='1'B, TCV_InvkId:='01'O)	RegisterReq(TCV_chTch, RegisterPdu_34(TCV_TI2, facilityIEtsnd(USSDReq_03(TCV_InvkId,		

			TCV_UssdString1))))		
15		+CheckUssdStringDisplayed(TCV_UssdString1)			
16		(TCV_Null := OO_InitSS(TCV_UssdString2))			
17		+DTMFSignalling(OC_LengthOfString(TCV_UssdString2), TI_01, TI_02, TCV_chTch)			
18	L_0742	L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI1, facilityIErcv(USSDReq_04(TCV_InvkId, TCV_UssdString2))))	(P)	
19		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_47(TCV_TI2))		
20		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TI_02, Cause_01, ProgInd_omit, UuInfo_omit))		
21		L?DL_DatInRel	ReleaseRcv(Release_02)		
22		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TI_02))		
23		+ChanRel_end(TCV_chTch)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_31_9_2_4

Group: GSM_L3_MS_v4220/SS/

Purpose: To verify that when the mobile station receives an USSD operation in parallel to any call independent supplementary transaction, it responds with a return error component in a RELEASE COMPLETE message, containing "USSD-Busy" error.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		(TCV_UssdString1 := "Type *70*635*562# and send", TCV_UssdString2 := "*70*635*562#")			
4	body	+ltree_body			
		ltree_body			
5		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
6		L!DL_DatRqRegister(TCV_TI.ti_v:='000'B, TCV_TI.ti_f:='0'B, TCV_TI0.ti_v:='000'B, TCV_TI0.ti_f:='1'B, TCV_InvkId0:='00'O)	RegisterReq(TCV_ch, RegisterPdu_34(TCV_TI, facilityIEtsnd(NotificationSS_06(TCV_InvkId0, TCV_UssdString1))))		
7		+CheckUssdStringDisplayed(TCV_UssdString1)			
8		(TCV_Null:=OO_InitSS(TCV_UssdString2))			
9	L_0743	L?DL_DatInFac	FacilityRcv(FacilityPdu_26_ci(TCV_TI0, facilityIErcv(USSDReq_04(TCV_InvkId0, TCV_UssdString2))))	(P)	
10		L!DL_DatRqRegister(TCV_TI2.ti_v:='001'B, TCV_TI2.ti_f:='0'B, TCV_TI1.ti_v:='001'B, TCV_TI1.ti_f:='1'B, TCV_InvkId1:='01'O)	RegisterReq(TCV_ch, RegisterPdu_34(TCV_TI2, facilityIEtsnd(NotificationSS_06(TCV_InvkId1, TCV_UssdString1))))		
11	L_0744	L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_50(TCV_TI1, TCV_InvkId1))	(P)	
12		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_09(TCV_TI, facilityIEtsndiei(USSDReq_06(TCV_InvkId0))))		
13		+ChanRel_end(TCV_ch)			

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_31_10

Group: GSM_L3_MS_v4220/SS/

Purpose: To check that the entry of two digits in the form !X (X in the set 0..9) followed by SEND is accepted by the mobile station in idle mode as a normal call establishment for the 1X number. It is checked that the MS sends a CHANNEL REQUEST, sends CM SERVICE REQUEST message for mobile originated call (after having received an IMMEDIATE ASSIGNMENT), and then sends the SETUP message containing the 1X phone number as called number (after having received the CM SERVICE ACCEPT message)

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
3		+test("10")			
4		+test("11")			
5		+test("12")			
6		+test("13")			
7		+test("14")			
8		+test("15")			
9		+test("16")			
10		+test("17")			
11		+test("18")			
12		+test("19")			
		test(num:IA5String)			
13		(TCV_Null := OO_InitSS(num))			2.
14		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
15		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
16		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
17		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_04)		
18		ACTIVATE(OtherEventsFail)			Restore Normal default
19		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
20		L?DL_DatInSetup(TCV_TI := DL_DatInSetup.msg.ti, TCV_CalledNum := DL_DatInSetup.msg.cdpn, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B)	SetupRcv(SetupInd_01)		
21		L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_08(TCV_TI))		
22		+PostMainLinkRel(TCV_ch)			
23		+ltree_check(num)			3.
24		START T_dly(10000)			
25		?TIMEOUT T_dly			
		ltree_check(num:IA5String)			

26		(TCV_Res := OC_CalledNumCHK(TCV_CalledNum.digits, num))			3.
27	L_0745	[TCV_Res]		(P)	
28	L_0746	[NOT TCV_Res]		(F)	

Detailed Comments:

1. To setup physical channel as BCCH, CCCH and SDCCH4.
2. To dial the two digits number.
3. To check whether the received called party number is the same as the dialled number.

Test Group SM

Test Case Dynamic Behaviour					
Test Case Name:		TC_34_2_1			
Group:		GSM_L3_MS_v4220/SM/			
Purpose:		To verify the ability of a MS to receive and decode the SMS where provided for the point to point service.			
Default:		OtherEventsFail			
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		START T_guard(1200)			
2		+ltree_emptystorage(TRUE)			
3		(TCV_asscmd_ts := TSPX_TmSltDef, TCV_RPOA_MT:='1111111111'O, TCV_TPOA1:='3333333333'O, TCV_Rpmr := '00'O)			
4		+BasicServiceMT(TSPX_MTBscSvcE, C_Full)			
5		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_48, BcchFreqLst_Omit, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
6		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
7	body	+ltree_part1			1
8		+ltree_part2			2
9		+ltree_part3			3
10		+ltree_part4			4
		ltree_part1			
11		+ltree_Environment1			
12		+ltree_sms1(TCV_ch)			
13		+ltree_CheckMessage(TCV_ch, FALSE)			
14		+ltree_Environment1			
15		+ltree_sms3(TCV_ch)			
16		+ltree_CheckMessage(TCV_ch, FALSE)			
17		+ltree_Environment1			
18		+ltree_sms4(TCV_ch, 5)			
19		+ltree_CheckMessage(TCV_ch, TRUE)			
		ltree_part2			
20		+ltree_Environment2			
21		+ltree_sms1(TCV_chSms)			
22		+ltree_CheckMessage(TCV_chTch, TRUE)			
23		+ltree_Environment2			
24		+ltree_sms3(TCV_chSms)			
25		+ltree_CheckMessage(TCV_chTch, TRUE)			
26		+ltree_Environment2			
27		+ltree_sms4(TCV_chSms, 15)			
28		+ltree_CheckMessage(TCV_chTch, TRUE)			
		ltree_part3			

29		+ltree_Environment2			
30		+ltree_ClearTchChannel_SS			
31		+ltree_sms1(TCV_chSms)			
32		+ltree_CheckMessage(TCV_chTch, TRUE)			
		ltree_part4			
33		+ltree_Environment2			
34		+ltree_ClearTchChannel_MS			
35		+ltree_CheckMessage(TCV_chTch, TRUE)			
		ltree_Environment1			
36		+RRmtcallprepare(TimingAdv(0), TSPX_RANDDef)			
37		(TCV_CPDataRetx := 0, TCV_ti_v_2 := '000'B, TCV_chSms := TCV_ch)			
38		+ltree_set_sapi3_SDCCH			
		ltree_Environment2			
39		+Est_MT_CallNonFH(TimingAdv(0), C_Full, TSPX_TCHHSubDef, TSPX_RANDDef)			
40		(TCV_CPDataRetx := 0, TCV_ti_v_2 := '000'B, TCV_chSms := C_SACCHF_A_1)			
41		+ltree_set_sapi3_SACCH			
		ltree_set_sapi3_SDCCH			
42		L!DL_EstRq	DLEstRq(TCV_ch)		SABM(SAPI=3)
43		L?DL_EstCo	DLEstCo(TCV_ch)		UA(SAPI=3)
		ltree_set_sapi3_SACCH			
44		L!DL_EstRq	DLEstRq(TCV_chSms)		SABM(SAPI=3)
45		L?DL_EstCo	DLEstCo(TCV_chSms)		UA(SAPI=3)
		ltree_ClearTchChannel_SS			
46		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TI_02, Cause_01, ProgInd_omit, UuInfo_omit))		
47	L_0747	L?DL_DatInRel	ReleaseRcv(Release_10(TI_01))	(P)	
48		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TI_02))		
		ltree_ClearTchChannel_MS			
49		(TCV_Null := OO_TermCall())			
50	L_0748	L?DL_DatInDisc	DiscRcv(TCV_chTch, DisconnR(TI_01, Cause_Def))	(P)	
51		L!DL_DatRqCpData START T_dly(25000)	DatRqCpData(CpDataPdu_01(TI_07(TCV_ti_v_2), CpData_01(TCV_TPOA1, TCV_RPOA_MT, TCV_Rpmr, C_Tzone0)), TCV_chSms)		CP-DATA including RP-Data(SMS DELIVER) n->ms
52		L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_08(TI_02))		
53		L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TI_01))		
54	L_0749	?TIMEOUT T_dly		(F)	

55		L?DL_DatInCpDataAck START T_dly(60000)	DatInCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v_2)))		CP-ACK ms->n
56	L_0750	?TIMEOUT T_dly		(F)	
57		L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))		CP-DATA including RP-Ack ms->n
58		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		CP-ACK n->ms
59		+NoReaction(TSPX_TC1M * 1000)			Step 128
		ltree_sms1(ch: LOGICCH)			
60		+ltree_sms2(ch)			
61		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		CP-ACK n->ms
62		+NoReaction(TSPX_TC1M * 1000)			Step 17, 71, 113
		ltree_sms2(ch: LOGICCH)			
63		L!DL_DatRqCpData (TCV_SMcntns := DL_DatRqCpData.msg.CPdata.rpdata.rpusrdat.tpdeliver.ud) START T_dly(25000)	DatRqCpData(CpDataPdu_01(TI_07(TCV_ti_v_2), CpData_01(TCV_TPOA1, TCV_RPOA_MT, TCV_Rpmr, C_Tzone0)), TCV_chSms)		CP-DATA including RP-Data(SMS DELIVER) n->ms
64	L_0751	?TIMEOUT T_dly		(F)	
65		L?DL_DatInCpDataAck START T_dly(60000)	DatInCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v_2)))		CP-ACK ms->n
66	L_0752	?TIMEOUT T_dly		(F)	
67		L?DL_DatInCpData CANCEL T_dly	DatInCpData(ch, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))		CP-DATA including RP-Ack ms->n
		ltree_sms3(ch: LOGICCH)			
68		+ltree_sms2(ch)			
69		START T_dly(TCV_T_TwiceTC1M)			
70	L_0753	?TIMEOUT T_dly		(F)	First CP-DATA(Rp-Ack) not acknowledged
71		L?DL_DatInCpData CANCEL T_dly	DatInCpData(ch, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))		CP-DATA(RP-Ack) retransmitted
72		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		Second CP-DATA(Rp-Ack) acknowledged
73		+NoReaction(TSPX_TC1M * 1000)			Step 38, 84
		ltree_sms4(ch: LOGICCH; time: INTEGER)			
74		+ltree_sms2(ch)			
75		REPEAT ltree_sms5(ch) UNTIL [TCV_CPDataRetx = TSPX_MaxCPDataRetx]			CP-DATA(RP-Ack) retransmitted
76		START T_dly(((TSPX_TC1M + time) * 1000))			
77	L_0754	L?DL_DatInCpData CANCEL T_dly	DatInCpData(ch, CpDataPdu_02(TI_08(TCV_ti_v_2),	(F)	RP-Ack shall not be sent more than TSPX_MaxCPDataRetx

			CpData_02(TCV_Rpmr)))		times
78	L_0755	?TIMEOUT T_dly		(P)	
		ltree_sms5(ch: LOGICCH)			
79		START T_dly(TCV_T_TwiceTC1M)			
80	L_0756	?TIMEOUT T_dly		(F)	
81		L?DL_DatInCpData CANCEL T_dly	DatInCpData(ch, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))		CP-DATA (RP-Ack) ms->n
82		(TCV_CPDataRetx := TCV_CPDataRetx + 1)			
		ltree_CheckMessage(ch: LOGICCH; Emptying:BOOLEAN)			
83		+ChanRel(ch)			
84		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey), TCV_Res := OO_CheckMessageDisplayed(160, TCV_SMcntns))			
85	L_0757	[NOT TCV_Res]		(F)	
86		+ltree_emptystorage(Emptying)			
87	L_0758	[TCV_Res]		(P)	
88		+ltree_emptystorage(Emptying)			
		ltree_emptystorage(Emptying:BOOLEAN)			
89		[Emptying]			
90		(TCV_Null := OO_EmptyMessageStorage())			
91		[NOT Emptying]			

Detailed Comments:

1: Parts a) to f) of the test procedure as described in GSM 11.10, 34.2.1
2: Parts g) to j) of the test procedure as described in GSM 11.10, 34.2.1
3: Part k) of the test procedure as described in GSM 11.10, 34.2.1
4: Part l) of the test procedure as described in GSM 11.10, 34.2.1

Test Case Dynamic Behaviour

Test Case Name: TC_34_2_2

Group: GSM_L3_MS_v4220/SM/

Purpose: To verify the MS ability to correctly send a short message where the SMS is provided for the point to point service. It also verifies the MS capability to simultaneously receive a network originated SM whilst sending a mobile originated SM.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(1200)			
2		(TCV_Null:=OO_EmptyMessageStorage(), TCV_asscmd_ts := TSPX_TmSltDef, TCV_RPOA_MT:='1111111111'O, TCV_TPOA1:='5555555555'O)			
3		+BasicServiceMT(TSPX_MTBscSvcF, C_Full)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
6	body	+ltree_part1			1
7		+ltree_part2			2
8		+ltree_part3			3
9		+ltree_part4			4
		ltree_part1			
10		+ltree_Environment1_3			
11		+ltree_sms1(TCV_ch)			
12		+ltree_ChanRel(TCV_ch)			
13		+ltree_Environment1_3			
14		+ltree_sms3(TCV_ch, 5)			
15		+ltree_Environment1_3			
16		+ltree_sms4			CP-Error
17		+ltree_ChanRel(TCV_ch)			
		ltree_part2			
18		+ltree_Environment2			
19		+ltree_sms1(TCV_chSms)			
20		+ltree_ChanRel(TCV_chTch)			
21		+ltree_Environment2			
22		+ltree_sms3(TCV_chTch, 15)			
		ltree_part3			
23		+ltree_Environment1_3			
24		+ltree_sms5			MT-SM
25		+ltree_sms2(TCV_ch)			Terminate MO-SM
26		+ltree_ChanRel(TCV_ch)			
27		(TCV_Res := OO_CheckMessageDisplayed(160, TCV_SMcntns))			
28	L_0759	[NOT TCV_Res]		(F)	
29	L_0760	[TCV_Res]		(P)	

		ltree_part4			
30		+ltree_Environment4			
31		+ChanRel_end(TCV_ch)			
		ltree_init			
32		(TCV_Null:=OO_SendMOShortMessage())			
33		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
34		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
35		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
36		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_09)		
37		ACTIVATE(OtherEventsFail)			Restore normal default tree
		ltree_Environment1_3			
38		+ltree_init			
39		+Authentication(TCV_ch, TCV_cksn, TSPX_RANDDef)			
40		+Ciphering_on(TCV_ch)			
41		+ltree_set_sapi3_SDCCH			
		ltree_Environment2			
42		+Wait(C_T_Wait1)			
43		+Est_MT_CallNonFH(TimingAdv(0), C_Full, TSPX_TCHHSubDef, TSPX_RANDDef)			
44		(TCV_Null := OO_SendMOShortMessage())			
45		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_09)		
46		L!DL_DatRqCmsAcp	CMSerAcp(TCV_chTch, CMServiceAcp_01)		
47		+ltree_set_sapi3_SACCH			
		ltree_Environment4			
48		+ltree_init			
49		L!DL_DatRqCmsRej	CMSerRej(TCV_ch, CMServiceRej_03)		Service Option not supported
50		START T_dly(5000)			
51	L_0761	?TIMEOUT T_dly		(P)	
52	L_0762	L?DL_EstIn CANCEL T_dly	DLEstInd_2(TCV_ch)	(F)	SAPI-3 shall NOT be established
		ltree_sms1(ch: LOGICCH)			
53		L?DL_DatInCpData(TCV_Rpmr := DL_DatInCpData.msg.CPdata.rpdata.rpmr, TCV_ti_v := DL_DatInCpData.msg.ti.ti_v)	DatInCpData(ch, CpDataPdu_03(CpData_03(RpData_03)))		CP-DATA(RP-Data SMS SUBMIT)
54		+ltree_sms2(ch)			Terminate MO-SM
		ltree_sms2(ch: LOGICCH)			
55		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v)), TCV_chSms)		CP-ACK n->ms
56		L!DL_DatRqCpData START T_dly(25000)	DatRqCpData(CpDataPdu_02(TI_08(TCV_ti_v), CpData_04(TCV_Rpmr)),		CP-DATA(RP-Ack) n->ms

			TCV_chSms)		
57	L_0763	?TIMEOUT T_dly		(F)	
58		+ChanRel(ch)			
59	L_0764	L?DL_DatInCpDataAck CANCEL T_dly	DatInCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v)))	(P)	CP-ACK ms->n
		ltree_sms3(ch: LOGICCH; t:INTEGER)			
60		L?DL_DatInCpData(TCV_Rpmr := DL_DatInCpData.msg.CPdata.rpdata.rpmr, TCV_ti_v := DL_DatInCpData.msg.ti.ti_v)	DatInCpData(TCV_chSms, CpDataPdu_03(CpData_03(RpData_03)))		CP-DATA(RP-Data SMS SUBMIT)
61		REPEAT ltree_sms6(ch) UNTIL [TCV_CPDataRetx = TSPX_MaxCPDataRetx]			MO-SM is retransmitted
62		START T_dly(((TSPX_TC1M + t) * 1000))			
63	L_0765	L?DL_DatInCpData(TCV_Rpmr := DL_DatInCpData.msg.CPdata.rpdata.rpmr, TCV_ti_v := DL_DatInCpData.msg.ti.ti_v)	DatInCpData(TCV_chSms, CpDataPdu_03(CpData_03(RpData_03)))	(F)	RP-Data(RP Data SMS SUBMIT) shall NOT be sent more than TSPX_MaxCPDataRetx times
64		+ChanRel(ch)			
65		?TIMEOUT T_dly			
66		+ltree_ChanRel(ch)			
		ltree_sms4			
67		L?DL_DatInCpData(TCV_Rpmr := DL_DatInCpData.msg.CPdata.rpdata.rpmr, TCV_ti_v := DL_DatInCpData.msg.ti.ti_v)	DatInCpData(TCV_chSms, CpDataPdu_03(CpData_03(RpData_03)))		CP-DATA(RP-Data SMS SUBMIT)
68		L!DL_DatRqCpError	DatRqCpError(CpErrPdu_01(TI_08(TCV_ti_v)), TCV_chSms)		CP error n->ms "Network Failure"
		ltree_sms5			
69		L?DL_DatInCpData(TCV_Rpmr := DL_DatInCpData.msg.CPdata.rpdata.rpmr, TCV_ti_v := DL_DatInCpData.msg.ti.ti_v)	DatInCpData(TCV_chSms, CpDataPdu_03(CpData_03(RpData_03)))		CP-DATA(RP-Data SMS SUBMIT)
70		+ltree_ti			
71		L!DL_DatRqCpData (TCV_SMcntns := DL_DatRqCpData.msg.CPdata.rpdata.rpusrdat.tpdeliver.ud) START T_dly(25000)	DatRqCpData(CpDataPdu_01(TI_07(TCV_ti_v_2), CpData_01(TCV_TPOA1, TCV_RPOA_MT, TCV_Rpmr, C_Tzone1)), TCV_chSms)		CP-DATA(RP-Data SMS DELIVER), n->ms
72	L_0766	?TIMEOUT T_dly		(F)	
73		+ChanRel(TCV_ch)			
74	L_0767	L?DL_DatInCpDataAck START T_dly(60000)	DatInCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v_2)))	(P)	CP-ACK ms->n
75	L_0768	?TIMEOUT T_dly		(F)	
76		+ChanRel(TCV_ch)			
77	L_0769	L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))	(P)	CP-DATA(RP-Ack) n->ms
78		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		

		ltree_sms6(ch: LOGICCH)			
79		START T_dly(TCV_T_TwiceTC1M)			
80	L_0770	?TIMEOUT T_dly		(F)	
81		+ChanRel(ch)			
82		L?DL_DatInCpData(TCV_Rpmr := DL_DatInCpData.msg.CPdata.rpdata.rpmr, TCV_ti_v := DL_DatInCpData.msg.ti.ti_v) CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_03(CpData_03(RpData_03)))		CP-DATA(RP-Data SMS SUBMIT)
83		(TCV_CPDataRetx := TCV_CPDataRetx + 1)			
		ltree_set_sapi3_SDCCH			
84		L?DL_EstIn	DLEstInd_2(TCV_ch)		
85		(TCV_CPDataRetx := 0, TCV_chSms := TCV_ch)			
		ltree_set_sapi3_SACCH			
86		L?DL_EstIn	DLEstInd_2(C_SACCHF_A_1)		
87		(TCV_CPDataRetx := 0, TCV_chSms := C_SACCHF_A_1)			
		ltree_ti			
88		[TCV_ti_v='000'B]			
89		(TCV_ti_v_2:='001'B)			
90		[NOT(TCV_ti_v='000'B)]			
91		(TCV_ti_v_2:='000'B)			
		ltree_ChanRel(ch: LOGICCH)			
92		+ChanRel(ch)			
93		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			

Detailed Comments:
1: Parts a) to f) of the test procedure as described in GSM 11.10, 34.2.2.3
2: Parts g) to i) of the test procedure as described in GSM 11.10, 34.2.2.3
3: Part j) of the test procedure as described in GSM 11.10, 34.2.2.3
4: Part k) of the test procedure as described in GSM 11.10, 34.2.2.3

Test Case Dynamic Behaviour

Test Case Name: TC_34_2_3

Group: GSM_L3_MS_v4220/SM/

Purpose: 1. To verify that the MS sends the correct acknowledgement when its memory in the SIM becomes full.

2. To verify that the MS sends the correct acknowledgement when its memory in the ME and the SIM becomes full, and sets the "memory exceeded" notification flag in the SIM.

3. To verify that the MS performs the "memory available" procedure when its message store becomes available for receiving short messages, and only at this moment.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_Null := OO_EmptyMessageStorage(), TCV_Null := OO_MSSetupStoreClass1SMInMEMemory(), TCV_Null := OO_ConnectSIMSimulator(), TCV_RPOA_MT:='1111111111'O, TCV_TPOA1:='3333333333'O, TCV_Rpmr:='00'O)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4	body	+ltree_part1			1
5		+ltree_part2			2
6		+ltree_sms3			3
7		+ltree_sms4			4
		ltree_part1			
8		(TCV_ProtErrorUnspec := FALSE, TCV_MemCapExcd := FALSE)			
9		REPEAT ltree_sms1 UNTIL [(TCV_ProtErrorUnspec) OR(TCV_MemCapExcd)]			
		ltree_part2			
10		(TCV_MemCapExcd := FALSE)			
11		REPEAT ltree_sms2 UNTIL [TCV_MemCapExcd]			
		ltree_Environment			
12		+Wait(C_T_Wait1)			
13		+RRmtcallprepare(TimingAdv(0), TSPX_RANDDef)			
14		+ltree_set_sapi3_SDCCH_MT			
15		(TCV_ti_v_2:='000'B, TCV_chSms:=TCV_ch)			
		ltree_Environment2			
16		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
17		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
18		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
19		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_09)		
20		ACTIVATE(OtherEventsFail)			Restore normal default tree

21		+Authentication(TCV_ch, TCV_cksn, TSPX_RANDDef)			
22		+Ciphering_on(TCV_ch)			
23		+ltree_set_sapi3_SDCCH_MO			
24		(TCV_chSms:=TCV_ch)			
		ltree_sms1			
25		+ltree_Environment			
26		L!DL_DatRqCpData START T_dly(25000)	DatRqCpData(CpDataPdu_01(TI_07(TCV_ti_v_2), CpData_05(TCV_TPOA1, TCV_RPOA_MT, TCV_Rpmr, C_Tzone2)), TCV_chSms)		CP-DATA(RP-Data SMS DELIVER), n->ms class 2 MT-SM
27	L_0771	?TIMEOUT T_dly		(F)	
28		+ChanRel(TCV_ch)			
29	L_0772	L?DL_DatInCpDataAck START T_dly(60000)	DatInCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v_2)))	(P)	CP-ACK ms->n
30	L_0773	?TIMEOUT T_dly		(F)	
31		+ChanRel(TCV_ch)			
32	L_0774	L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))	(P)	CP-DATA(RP-Ack) ms->n
33		L!DL_DatRqCpDataAck CANCEL T_dly	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		CP-ACK n->ms
34		+ChanRel(TCV_ch)			
35		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
36	L_0775	L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_07(TCV_Rpmr)))	(P)	CP-DATA(RP-Error: Protocol Error, unspecified)
37		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		CP-ACK n->ms
38		(TCV_ProtErrorUnspec := TRUE)			SMS storage in SIM full
39		+ChanRel(TCV_ch)			
40		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
41	L_0776	L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_08(TCV_Rpmr)))	(P)	CP-DATA(RP-Error: Memory Capacity Exceeded)
42		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		CP-ACK n->ms
43		(TCV_MemCapExcd := TRUE)			SMS storage in ME + SIM full
44		+ChanRel(TCV_ch)			
45		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey), TCV_Res := OO_CheckMCEFOnSIM())			
46	L_0777	[NOT TCV_Res]		(F)	
47	L_077	[TCV_Res]		(P)	

	8				
		ltree_sms2			
48		+ltree_Environment			
49		L!DL_DatRqCpData START T_dly(25000)	DatRqCpData(CpDataPdu_01(TI_07(TCV_ti_v_2), CpData_06(TCV_TPOA1, TCV_RPOA_MT, TCV_Rpmr, C_Tzone2)), TCV_chSms)		CP-DATA(RP-Data SMS DELIVER), n->ms, class 1 MT-SM
50	L_0779	?TIMEOUT T_dly		(F)	
51		+ChanRel(TCV_ch)			
52		L?DL_DatInCpDataAck START T_dly(60000)	DatInCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v_2)))		CP-ACK ms->n
53	L_0780	?TIMEOUT T_dly		(F)	
54		+ChanRel(TCV_ch)			
55		L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))		CP-DATA(RP-Ack) ms->n
56		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		CP-ACK n->ms
57		+ChanRel(TCV_ch)			
58		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
59		L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_08(TCV_Rpmr)))		CP-DATA(RP-Error: Memory Capacity Exceeded)
60		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		CP-ACK n->ms
61		(TCV_MemCapExcd := TRUE)			SMS storage in ME + SIM full
62		+ChanRel(TCV_ch)			
63		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey), TCV_Res := OO_CheckMCEFOnSIM())			
64	L_0781	[NOT TCV_Res]		(F)	
65	L_0782	[TCV_Res]		(P)	
		ltree_sms3			
66		+ltree_Environment			
67		L!DL_DatRqCpData START T_dly(25000)	DatRqCpData(CpDataPdu_01(TI_07(TCV_ti_v_2), CpData_01(TCV_TPOA1, TCV_RPOA_MT, TCV_Rpmr, C_Tzone2)), TCV_chSms)		CP-DATA(RP-Data SMS DELIVER) TP-DCS is 0
68	L_0783	?TIMEOUT T_dly		(F)	
69		+ChanRel(TCV_ch)			
70		L?DL_DatInCpDataAck START T_dly(60000)	DatInCpDataAck(CpDataAckPdu_01(		CP-ACK ms->n

			TI_08(TCV_ti_v_2)))		
71	L_0784	?TIMEOUT T_dly		(F)	
72		+ChanRel(TCV_ch)			
73		L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_08(TCV_Rpmr)))		CP-DATA(RP-Error: Memory Capacity Exceeded)
74		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		CP-ACK n->ms
75		+ChanRel(TCV_ch)			
76		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey), TCV_Null := OO_RemoveOneSM())			
		ltree_sms4			
77		+ltree_Environment2			
78		L?DL_DatInCpData(TCV_ti_v := DL_DatInCpData.msg.ti.ti_v)	DatInCpData(TCV_chSms, CpDataPdu_02(TI_09, CpData_09))		CP-DATA(RP-SMMA), ms->n
79		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v)), TCV_chSms)		CP-ACK n->ms
80		L!DL_DatRqCpData START T_dly(25000)	DatRqCpData(CpDataPdu_02(TI_08(TCV_ti_v), CpData_04(TCV_Rpmr)), TCV_chSms)		CP-DATA(RP-Ack) n->ms
81	L_0785	?TIMEOUT T_dly		(F)	
82		+ChanRel(TCV_ch)			
83		L?DL_DatInCpDataAck CANCEL T_dly	DatInCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v)))		CP-ACK ms->n
84		+ChanRel(TCV_ch)			
85		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey), TCV_Res := OO_CheckMCEFOnSIMUnset())			
86	L_0786	[NOT TCV_Res]		(F)	
87	L_0787	[TCV_Res]		(P)	
88		(TCV_Null := OO_RemoveOneSM())			
89		START T_dly(60000)			
90	L_0788	L?DL_RacInChRq(TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_04)	(F)	The MS shall not attempt to send RP-SSMA
91		+ChanRel(TCV_ch)			
92	L_0789	?TIMEOUT T_dly		(P)	
		ltree_set_sapi3_SDCCH_MT			
93		L!DL_EstRq	DLEstRq(TCV_ch)		
94		L?DL_EstCo	DLEstCo(TCV_ch)		
		ltree_set_sapi3_SDCCH_MO			
95		L?DL_EstIn	DLEstInd_2(TCV_ch)		

Detailed Comments:
1: Part a) of the test procedure as described in GSM 11.10, 34.2.3.3
2: Parts b) to c) of the test procedure as described in GSM 11.10, 34.2.3.3
3: Parts d) to e) of the test procedure as described in GSM 11.10, 34.2.3.3
4: Parts f) to k) of the test procedure as described in GSM 11.10, 34.2.3.3

Test Case Dynamic Behaviour

Test Case Name: TC_34_2_4
Group: GSM_L3_MS_v4220/SM/
Purpose: 1) To verify that the MS is able to accept a SMS-STATUS-REPORT TPDU.

2) To verify that a MS able to use the SMS-COMMAND functionality correctly sends a SMS-COMMAND TPDU with the correct TP-Message-Reference.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_RPOA_MT:='1111111111'O, TCV_TPDA:='5555555555'O)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4	body	+ltree_part1			1
5		+ltree_part2			2
6		+ltree_part3			3
7		+ltree_part4			4
		ltree_part1			
8		(TCV_Null := OO_SendMOShortMessage())			
9		+ltree_PrepareEnvironmentParts1_3_4			
10		+ltree_sms1			Initiate MO-SM, status report requested
11		+ltree_sms2			Terminate MO-SM
		ltree_part2			
12		+ltree_PrepareEnvironmentPart2			
13		+ltree_sms3			MT-SM (SMS Status Report)
		ltree_part3			
14		(TCV_Null := OO_SendSMSCOMMANDEnq())			
15		+ltree_PrepareEnvironmentParts1_3_4			
16		+ltree_sms4			Initiate MO-SM (SMS COMMAND, enquiry)
17		+ltree_sms2			Terminate MO-SM
		ltree_part4			
18		(TCV_Null := OO_SendSMSCOMMANDDel())			
19		+ltree_PrepareEnvironmentParts1_3_4			
20		+ltree_sms5			Initiate MO-SM (SMS COMMAND, deletion)
21		+ltree_sms2			Terminate MO-SM
		ltree_PrepareEnvironmentParts1_3_4			
22		(TCV_chSms:=TCV_ch)			
23		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
24		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
25		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot,		

			TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
26		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_09)		
27		ACTIVATE(OtherEventsFail)			Restore normal default tree
28		+Authentication(TCV_ch, TCV_cksn, TSPX_RANDDef)			
29		+Ciphering_on(TCV_ch)			
30		+ltree_set_sapi3_SDCCH_MO			
		ltree_PrepareEnvironmentPart2			
31		+Wait(C_T_Wait1)			
32		+RRmtcallprepare(TimingAdv(0), TSPX_RANDDef)			
33		+ltree_set_sapi3_SDCCH_MT			
34		(TCV_ti_v_2 := '000'B, TCV_chSms := TCV_ch)			
		ltree_sms1			
35		L?DL_DatInCpData(TCV_Rpmr := DL_DatInCpData.msg.CPdata.rpdata.rpmr, TCV_Tpmr := DL_DatInCpData.msg.CPdata.rpdata.rpusrdat.tpsubmit.mr, TCV_ti_v := DL_DatInCpData.msg.ti.ti_v)	DatInCpData(TCV_chSms, CpDataPdu_03(CpData_03(RpData_08)))		RP-Data
		ltree_sms2			
36		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v)), TCV_chSms)		
37		L!DL_DatRqCpData	DatRqCpData(CpDataPdu_02(TI_08(TCV_ti_v), CpData_04(TCV_Rpmr)), TCV_chSms)		RP-Ack
38		START T_dly(25000)			
39	L_0790	?TIMEOUT T_dly		(F)	
40		+ChanRel(TCV_ch)			
41		L?DL_DatInCpDataAck CANCEL T_dly	DatInCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v)))		
42		+ChanRel_P(TCV_ch)			
43		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
		ltree_sms3			
44		+ltree_PrepareEnvironmentPart2			
45		L!DL_DatRqCpData	DatRqCpData(CpDataPdu_01(TI_07(TCV_ti_v_2), CpData_12(TCV_TPDA, TCV_RPOA_MT, TCV_Tpmr, TCV_Rpmr, C_Tzone3)), TCV_chSms)		RP-Data status report
46		START T_dly(25000)			
47	L_0791	?TIMEOUT T_dly		(F)	
48		+ChanRel(TCV_ch)			
49		L?DL_DatInCpDataAck CANCEL T_dly	DatInCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v_2)))		
50		START T_dly(60000)			
51	L_0792	?TIMEOUT T_dly		(F)	
52		+ChanRel(TCV_ch)			

53		L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))		RP-Ack
54		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		
55		+ChanRel(TCV_ch)			
56		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
		ltree_sms4			
57		L?DL_DatInCpData(TCV_Rpmr := DL_DatInCpData.msg.CPdata.rpdata.rpmr, TCV_Tpmr := DL_DatInCpData.msg.CPdata.rpdata.rpusrdat.tpcommand.mr, TCV_ti_v:=DL_DatInCpData.msg.ti.ti_v)	DatInCpData(TCV_chSms, CpDataPdu_03(CpData_03(RpData_11(TCV_Rpmr, TpCommand_01(TCV_Tpmr)))))		RP-Data, SMS COMMAND, enquiry
		ltree_sms5			
58		L?DL_DatInCpData(TCV_Rpmr := DL_DatInCpData.msg.CPdata.rpdata.rpmr, TCV_ti_v := DL_DatInCpData.msg.ti.ti_v)	DatInCpData(TCV_chSms, CpDataPdu_03(CpData_03(RpData_11(TCV_Rpmr, TpCommand_02(TCV_Tpmr)))))		RP-Data, SMS COMMAND, deletion
		ltree_set_sapi3_SDCCH_MO			
59		L?DL_EstIn	DLEstInd_2(TCV_ch)		
		ltree_set_sapi3_SDCCH_MT			
60		L!DL_EstRq	DLEstRq(TCV_ch)		
61		L?DL_EstCo	DLEstCo(TCV_ch)		
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_34_2_5_1
Group: GSM_L3_MS_v4220/SM/
Purpose: To verify that the MS will accept and display but not store a class 0 message, and that it will accept and display a class 0 message if its message store is full.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_Null:=OO_EmptyMessageStorage(), TCV_RPOA_MT:='1111111111'O, TCV_TPOA1:='3333333333'O, TCV_Rpmr:='00'O)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4	body	+ltree_sms1			1
5		(TCV_Res := OO_RecallAndDisplaySM(160, TCV_SMcntns))			
6	L_0793	[TCV_Res]		(F)	Message store should be empty
7	L_0794	[NOT TCV_Res]		(P)	
8		+ltree_Release			
9		(TCV_MemCapExcd := FALSE)			
10		REPEAT ltree_sms2 UNTIL [TCV_MemCapExcd]			
11		+ltree_Release			
12		+ltree_sms1			3
13		+ChanRel_end(TCV_ch)			
		ltree_Release			
14		+ChanRel(TCV_ch)			
15		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
		ltree_Environment			
16		+Wait(C_T_Wait1)			
17		+RRmtcallprepare(TimingAdv(0), TSPX_RANDDef)			
18		+ltree_set_sapi3_SDCCH_MT			
19		(TCV_ti_v_2 := '000'B, TCV_chSms := TCV_ch)			
		ltree_sms1			
20		+ltree_Environment			
21		L!DL_DatRqCpData (TCV_SMcntns := DL_DatRqCpData.msg.CPdata.rpdata.rpusrdat.tpdeliver.ud) START T_dly(25000)	DatRqCpData(CpDataPdu_01(TI_07(TCV_ti_v_2), CpData_10(TCV_TPOA1, TCV_RPOA_MT, TCV_Rpmr, C_Tzone4)), TCV_chSms)		RP-Data Initiate class 0 MT-SM
22	L_0795	?TIMEOUT T_dly		(F)	
23		L?DL_DatInCpDataAck START T_dly(60000)	DatInCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v_2)))		
24	L_079	?TIMEOUT T_dly		(F)	

	6				
25		L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))		RP-Ack
26		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		
27		(TCV_Res := OO_CheckMessageDisplayed(160, TCV_SMcntns))			
28	L_0797	[NOT TCV_Res]		(F)	
29	L_0798	[TCV_Res]		(P)	
		ltree_sms2			
30		+ltree_Environment			
31		L!DL_DatRqCpData START T_dly(25000)	DatRqCpData(CpDataPdu_01(TI_07(TCV_ti_v_2), CpData_06(TCV_RPOA_MT, TCV_TPOA1, TCV_Rpmr, C_Tzone4)), TCV_chSms)		RP-Data Initiate class 1 MT-SM
32	L_0799	?TIMEOUT T_dly		(F)	
33		L?DL_DatInCpDataAck START T_dly(60000)	DatInCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v_2)))		
34	L_0800	?TIMEOUT T_dly		(F)	
35		L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))		RP-Ack
36		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		
37		L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_08(TCV_Rpmr)))		RP-Error: Memory Capacity Exc.
38		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		
39		(TCV_MemCapExcd := TRUE)			
		ltree_set_sapi3_SDCCH_MT			
40		L!DL_EstRq	DLEstRq(TCV_ch)		
41		L?DL_EstCo	DLEstCo(TCV_ch)		
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_34_2_5_2
Group: GSM_L3_MS_v4220/SM/
Purpose: To verify that the MS acts correctly on receiving a class 1 message, i.e. that it stores the message in the ME or SIM and sends an acknowledgement (at RP and CP-Layer).
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_Null:=OO_EmptyMessageStorage(), TCV_Null:=OO_MSSetupStoreClass1SMInMEMemory(), TCV_RPOA_MT:='1111111111'O, TCV_TPOA1:= '3333333333'O, TCV_Rpmr:='00'O)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4	body	+ltree_Environment			1
5		+ltree_sms1			
		ltree_Environment			
6		+RRmtcallprepare(TimingAdv(0), TSPX_RANDDef)			
7		+ltree_set_sapi3_SDCCH_MT			
8		(TCV_ti_v_2:='000'B, TCV_chSms:=TCV_ch)			
		ltree_sms1			
9		L!DL_DatRqCpData (TCV_SMcntns := DL_DatRqCpData.msg.CPdata.rpdata.rpusrdat.tpdeliver.ud)	DatRqCpData(CpDataPdu_01(TI_07(TCV_ti_v_2), CpData_06(TCV_RPOA_MT, TCV_TPOA1, TCV_Rpmr, C_Tzone5)), TCV_chSms)		RP-Data Initiate class 1 MT-SM
10		START T_dly(25000)			
11	L_0801	?TIMEOUT T_dly		(F)	
12		+ChanRel(TCV_chTch)			
13	L_0802	L?DL_DatInCpDataAck CANCEL T_dly	DatInCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v_2)))	(P)	
14		START T_dly(60000)			
15	L_0803	?TIMEOUT T_dly		(F)	
16		+ChanRel(TCV_chTch)			
17	L_0804	L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))	(P)	RP-Ack
18		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		
19		+ChanRel(TCV_ch)			
20		(TCV_Res := OO_RecallAndDisplaySM(160, TCV_SMcntns))			
21	L_080	[NOT TCV_Res]		(F)	

22	5 L_0806	[TCV_Res]		(P)	
		ltree_set_sapi3_SDCCH_MT			
23		L!DL_EstRq	DLEstRq(TCV_ch)		
24		L?DL_EstCo	DLEstCo(TCV_ch)		
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_34_2_5_3

Group: GSM_L3_MS_v4220/SM/

Purpose: To verify that the MS acts correctly on receiving a class 2 message, i.e. that it stores the message correctly in the SIM, and if this is not possible, returns a protocol error message, with the correct error cuase, to the network. There are 2 cases. (1) If the MS supports storing of short messages in the SIM and in the ME, and storage in the ME is not full, and the short message cannot be stored in the SIM, the error cause shall be "protocol error, unspecified". (2) If the MS supports storing of short message in the SIM and not in the ME and short message cannot be stored in the SIM, the error cause shall be "memory capacity exceeded".

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_Null := OO_EmptyMessageStorage(), TCV_Null := OO_MSSetupStoreClass1SMInMEMemory(), TCV_Null := OO_ConnectSIMSimulator(), TCV_RPOA_MT := '1111111111'O, TCV_TPOA1:= '3333333333'O, TCV_Rpmr:= '00'O)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4	body	+ltree_sms1			1
5		+ltree_sms2			
6		+ltree_Release			
7		+ltree_sms1			
8		+ltree_sms3			
9		+ChanRel_end(TCV_ch)			
		ltree_Release			
10		+ChanRel(TCV_ch)			
11		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
		ltree_Environment			
12		+Wait(C_T_Wait1)			
13		+RRmtcallprepare(TimingAdv(0), TSPX_RANDDef)			
14		+ltree_set_sapi3_SDCCH_MT			
15		(TCV_ti_v_2:='000'B, TCV_chSms:=TCV_ch)			
		ltree_sms1			
16		+ltree_Environment			
17		L!DL_DatRqCpData START T_dly(25000)	DatRqCpData(CpDataPdu_01(TI_07(TCV_ti_v_2), CpData_05(TCV_TPOA1, TCV_RPOA_MT, TCV_Rpmr, C_Tzone0)), TCV_chSms)		CP-DATA(RP-Data SMS DELIVER), n->ms class 2 MT-SM
18	L_0807	?TIMEOUT T_dly		(F)	
19		L?DL_DatInCpDataAck START T_dly(60000)	DatInCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v_2)))		CP-ACK ms->n

		ltree_sms2			
20		(TCV_Res := OO_SIMSimulAttIndOK())			ME attempt to strore SM on SIM? Status OK ('9000')
21	L_0808	[NOT TCV_Res]		(F)	
22		[TCV_Res]			
23	L_0809	?TIMEOUT T_dly		(F)	
24		L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))		CP-DATA(RP-Ack) ms->n
25		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		CP-ACK n->ms
		ltree_sms3			
26		(TCV_Res := OO_SIMSimulAttIndMemProblem())			ME attempt to strore SM on SIM? Status 'memory problem' ('9240')
27	L_0810	[NOT TCV_Res]		(F)	
28		[TCV_Res]			
29		[TSPC_StoreRcvSMSME]			
30	L_0811	?TIMEOUT T_dly		(F)	
31		L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_07(TCV_Rpmr)))		CP-DATA(RP-Error: Protocol Error, unspecified)
32		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		CP-ACK n->ms
33		[NOT TSPC_StoreRcvSMSME]			
34	L_0812	?TIMEOUT T_dly		(F)	
35		L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_08(TCV_Rpmr)))		CP-DATA(RP-Error: Memory capacity exceeded)
36		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		CP-ACK n->ms
		ltree_set_sapi3_SDCCH_MT			
37		L!DL_EstRq	DLEstRq(TCV_ch)		
38		L?DL_EstCo	DLEstCo(TCV_ch)		
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_34_2_7
Group: GSM_L3_MS_v4220/SM/
Purpose: To verify the correct implementation of the replace mechanism for Replace Short Messages.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_Null:=OO_EmptyMessageStorage(), TCV_TPOA1:='1111111111'O, TCV_TPOA2:='2222222222'O, TCV_RPOA1:='3333333333'O, TCV_RPOA2:='4444444444'O, TCV_Rpmr:='00'O)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4		REPEAT ltree_RandomTypes UNTIL [NOT(TCV_SMTypeM = TCV_SMTypeN)]			
5	body	+ltree_part1			
6		+ChanRel_end(TCV_ch)			
		ltree_RandomTypes			
7		(TCV_SMTypeM:=OC_Random(1, 7), TCV_SMTypeN:=OC_Random(1, 7))			
		ltree_part1			
8		+ltree_sms1("First short message. ")			
9		+ltree_Release			
10		+ltree_sms1("Second short message. ")			
11		+ltree_Release			
12		+ltree_sms1("Third short message. ")			
13		+ltree_Release			
14		+ltree_sms1("Fourth short message. ")			
15		+ltree_Release			
16		+ltree_sms1("Fifth short message. ")			
17		(TCV_Res := OO_CheckAllSMPresentBut4th())			
18	L_0813	[NOT TCV_Res]		(F)	
19	L_0814	[TCV_Res]		(P)	
		ltree_Release			
20		+ChanRel(TCV_ch)			
21		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
		ltree_Environment			
22		+Wait(C_T_Wait1)			
23		+RRmtcallprepare(TimingAdv(0), TSPX_RANDDef)			
24		+ltree_set_sapi3_SDCCH_MT			
25		(TCV_ti_v_2:='000'B, TCV_chSms:=TCV_ch)			
		ltree_sms1(text: IA5String)			
26		+ltree_Environment			
27		L!DL_DatRqCpData START T_dly(25000)	DatRqCpData(CpDataPdu_16(		RP-Data

			TCV_TPOA1, TCV_RPOA1, TCV_SMTypeN, text, TI_07(TCV_ti_v_2), TCV_Rpmr, C_Tzone1), TCV_chSms)		
28	L_0815	?TIMEOUT T_dly		(F)	
29		L?DL_DatInCpDataAck START T_dly(60000)	DatInCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v_2)))		
30	L_0816	?TIMEOUT T_dly		(F)	
31		L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))		RP-Ack
32		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		
		ltree_set_sapi3_SDCCH_MT			
33		L!DL_EstRq	DLEstRq(TCV_ch)		
34		L?DL_EstCo	DLEstCo(TCV_ch)		
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_34_2_8

Group: GSM_L3_MS_v4220/SM/

Purpose: To verify that the MS is able to send a Reply Short Message back to the correct originating SME even if in the meantime it receives another Short Message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_Null := OO_EmptyMessageStorage(), TCV_RPOA1 := '1111111111'O, TCV_RPOA2 := '2222222222'O, TCV_RPDA_MT := '3333333333'O, TCV_TPOA1 := '4444444444'O, TCV_TPOA2 := '5555555555'O, TCV_RPOA_MO := TCV_RPDA_MT, TCV_RPDA_MO := TCV_RPOA1, TCV_TPDA := TCV_TPOA1, TCV_Rpmr := '00'O)			n->ms, 1st SM n->ms, 2nd SM n->ms, both n->ms, 1st SM n->ms, 2nd SM ms->n, both ms->n,1st reply ms->n,1st reply
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
4	body	+ltree_part1			
5		+ltree_part2			
6		+ChanRel_end(TCV_ch)			
		ltree_Release			
7		+ChanRel(TCV_ch)			
8		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
		ltree_Environment1			
9		+Wait(C_T_Wait1)			
10		+RRmtcallprepare(TimingAdv(0), TSPX_RANDDef)			
11		+ltree_set_sapi3_SDCCH_MT			
12		(TCV_ti_v_2 := '000'B, TCV_chSms := TCV_ch)			
		ltree_Environment2			
13		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
14		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
15		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
16		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_09)		
17		ACTIVATE(OtherEventsFail)			Restore normal default tree
18		+Authentication(TCV_ch, TCV_cksn, TSPX_RANDDef)			
19		+Ciphering_on(TCV_ch)			
20		+ltree_set_sapi3_SDCCH_MO			
		ltree_part1			

21		+ltree_sms1(TCV_TPOA1, TCV_RPOA1, "The first short message. ")			
22		(TCV_CB1 := TCV_SMcntns)			
23		+ltree_Release			
24		[TSPC_StoreRcvSMSME OR TSPC_StoreRcvSMSSIM]			SM can be stored
25		+ltree_sms1(TCV_TPOA2, TCV_RPOA2, "The second short message. ")			
26		(TCV_CB2 := TCV_SMcntns)			
27		+ltree_Release			
28		[NOT(TSPC_StoreRcvSMSME) AND NOT(TSPC_StoreRcvSMSSIM)]			SM cannot be stored
		ltree_part2			
29		(TCV_CB3 := OO_DisplaySMAndSendReplySM(1, 160, TCV_CB1))			TCV_CB3 holds the 1st reply data
30		+ltree_sms2(TCV_CB3)			Reply SM
31		+ltree_Release			
32		[TSPC_StoreRcvSMSME OR TSPC_StoreRcvSMSSIM]			SM can be stored
33		(TCV_RPDA_MO := TCV_RPOA2, TCV_TPDA := TCV_TPOA2, TCV_CB4 := OO_DisplaySMAndSendReplySM(2, 160, TCV_CB2))			
34		+ltree_sms2(TCV_CB4)			Reply SM
35		[NOT(TSPC_StoreRcvSMSME) AND NOT(TSPC_StoreRcvSMSSIM)]			SM cannot be stored
		ltree_sms1(tpoa1: BCDN; rpoa_mt: BCDN; text: IA5String)			
36		+ltree_Environment1			
37		L!DL_DatRqCpData (TCV_SMcntns := DL_DatRqCpData.msg.CPdata.rpdata.rpusrdat.tpdeliver.ud) START T_dly(25000)	DatRqCpData(CpDataPdu_17(tpoa1, rpoa_mt, text, TI_07(TCV_ti_v_2), TCV_Rpmr, C_Tzone6), TCV_chSms)		RP-Data
38	L_0817	?TIMEOUT T_dly		(F)	
39	L_0818	L?DL_DatInCpDataAck START T_dly(60000)	DatInCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v_2)))	(P)	
40	L_0819	?TIMEOUT T_dly		(F)	
41	L_0820	L?DL_DatInCpData CANCEL T_dly	DatInCpData(TCV_chSms, CpDataPdu_02(TI_08(TCV_ti_v_2), CpData_02(TCV_Rpmr)))	(P)	RP-Ack
42		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v_2)), TCV_chSms)		
		ltree_sms2(tpud:TPUD)			
43		+ltree_Environment2			
44		L?DL_DatInCpData(TCV_Rpmr := DL_DatInCpData.msg.CPdata.rpdata.rpmr, TCV_ti_v := DL_DatInCpData.msg.ti.ti_v)	DatInCpData(TCV_chSms, CpDataPdu_18(TCV_TPDA,TCV_RPDA_MT, tpud))		RP-Data, TP-submit
45		L!DL_DatRqCpDataAck	DatRqCpDataAck(CpDataAckPdu_01(TI_08(TCV_ti_v)), TCV_chSms)		
46		L!DL_DatRqCpData START T_dly(25000)	DatRqCpData(CpDataPdu_02(TI_08(TCV_ti_v), CpData_04(TCV_Rpmr)),		RP-Ack

			TCV_chSms)		
47	L_0821	?TIMEOUT T_dly		(F)	
48	L_0822	L?DL_DatInCpDataAck CANCEL T_dly	DatInCpDataAck(CpDataAckPdu_02(TI_07(TCV_ti_v)))	(P)	
		ltree_set_sapi3_SDCCH_MT			
49		L!DL_EstRq	DLEstRq(TCV_ch)		
50		L?DL_EstCo	DLEstCo(TCV_ch)		
		ltree_set_sapi3_SDCCH_MO			
51		L?DL_EstIn	DLEstInd_2(TCV_ch)		
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_34_3
Group: GSM_L3_MS_v4220/SM/
Purpose: This test verifies that an MS is able to receive SMS-CB messages.
This test verifies that a MS is able to respond to a paging requested during the transmission of a cell broadcast short message.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_CBch := C_CBCH_A)			
3		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_0, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_cbms, C_NCCP_2)			
4	body	+ltree_body			
		ltree_body			
5		+SendSMSCBMessage(SerialNumber_01)			Send cell broadcast message
6		(TCV_Res := OO_CheckCBSMReceived(93, TCV_SMSCBpack))			
7	L_0823	[NOT TCV_Res]		(F)	
8	L_0824	[TCV_Res]		(P)	
9		+SendSMSCBMessage(SerialNumber_02)			Send cell broadcast message
10		(TCV_Res := OO_CheckCBSMReceived(93,TCV_SMSCBpack))			
11	L_0825	[NOT TCV_Res]		(F)	
12	L_0826	[TCV_Res]		(P)	
13		+SendSMSCBMessage(SerialNumber_03)			Send cell broadcast message
14		(TCV_Res := OO_CheckCBSMReceived(93, TCV_SMSCBpack))			
15	L_0827	[NOT TCV_Res]		(F)	
16	L_0828	[TCV_Res]		(P)	
17		+ltree_body2			
		ltree_body2			
18		+SendSMSCBMessage(SerialNumber_01)			Send cell broadcast message
19		(TCV_Res := OO_CheckCBSMReceived(93, TCV_SMSCBpack))			
20	L_0829	[NOT TCV_Res]		(F)	
21	L_0830	[TCV_Res]		(P)	
22		(TCV_Null := OM_SendSMSCBWhilePaging(TCV_CBch))			Send cell broadcast message while paging
23		+SendSMSCBMessage(SerialNumber_02)			Send cell broadcast message
24		+localtree			

25		(TCV_Res := OO_CheckCBSMReceived(93, TCV_SMSCBpack))			
26		[NOT TCV_Res]			
27		+localtree2			
28	L_0831	[TCV_Res]		(P)	
29		+localtree2			
		localtree2			
30		+SendSMSCBMessage(SerialNumber_03)			Send cell broadcast message
31		(TCV_Res := OO_CheckCBSMReceived(93, TCV_SMSCBpack))			
32	L_0832	[NOT TCV_Res]		(F)	
33	L_0833	[TCV_Res]		(P)	
		localtree			
34		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
35		START T_dly(5000)			
36	L_0834	?TIMEOUT T_dly		(F)	
37		L?DL_RacInChRq(TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_17)		
38		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
39		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
40		+ChanRel(TCV_ch)			
		SendSMSCBMessage(serial_number:SERIAL_NUMBER)			
41		L!DL_UdatRqSMSCBData(TCV_CB1 := DL_UdatRqSMSCBData.msg.message_contents)	SMSCBReq(TCV_CBch, SMSCB_01(serial_number))		First block
42		L!DL_UdatRqSMSCBData(TCV_CB2 := DL_UdatRqSMSCBData.msg.message_contents)	SMSCBReq(TCV_CBch, SMSCB_02('0001'B, '0'B, 17, 38))		Second block, message octets 17 to 38
43		L!DL_UdatRqSMSCBData(TCV_CB3 := DL_UdatRqSMSCBData.msg.message_contents)	SMSCBReq(TCV_CBch, SMSCB_02('0010'B, '0'B, 39, 60))		Third block, message octets 39 to 60
44		L!DL_UdatRqSMSCBData(TCV_CB4 := DL_UdatRqSMSCBData.msg.message_contents)	SMSCBReq(TCV_CBch, SMSCB_02('0011'B, '1'B, 61, 82))		Fourth and last block, message octets 61 to 82
45		(TCV_SMSCBpack := SMSCBdata(TCV_CB1, TCV_CB2, TCV_CB3, TCV_CB4))			4 blocks data

Detailed Comments:

Test Step Library

Test Step Group management

Test Step Dynamic Behaviour					
Test Step Name:	StopAllBCCH				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To stop the RF transmission of all BCCH channels in all active cells.				
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			
2		(TCV_Null := OM_StopAllBCCH(C_BCCH_A_1, C_BCCH_B_1, C_BCCH_C_1, C_BCCH_D_1, C_BCCH_E_1, C_BCCH_F_1, C_BCCH_G_1, C_BCCH_H_1))			
3		[TSPC_DCS]			
4		(TCV_Null := OM_StopAllBCCH(C_BCCH_A_1, C_BCCH_B_1, C_BCCH_C_1, C_BCCH_D_1, C_BCCH_E_1, C_BCCH_F_1, C_BCCH_G_1, "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Stopmaindcch(ch_main:LOGICCH; ch_sacch:LOGICCH)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To stop the transmission of main dcch channel and wait till no more uplink SACCH frame.				
Default:	OtherEventsFail_01				
Comments:	`ch_main` identifying a main dcch channel and 'ch_sacch' a sacch channel.				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_Deactivate(ch_main, ch_sacch))			
2		(TCV_Null := OM_StartMsrReport(ch_sacch)) START T_dly1(C_T_mrsrp)			1.
3		(TCV_Res := FALSE)			
4		REPEAT ltree_rcvmsr UNTIL [TCV_Res]			
5		(TCV_Null := OM_StopMsrReport(ch_sacch))			
		ltree_rcvmsr			
6		L?DL_UdatInMsrRpt	MsrRept(MsrReport_02)		2.
7		START T_dly1(C_T_mrsrp)			
8	L_0835	?TIMEOUT T_dly1		(P)	
9		(TCV_Res := TRUE)			
Detailed Comments:		1. Wait until there is no more SACCH frames in the uplink direction. 2. If any measurement report message with any contents has been received, restart the timer again. The loop is watched by the T_guard in the default test step OtherEventsFail_01. This test step is called by TC_26_7_4_3_2, TC_26_7_4_3_3, TC_26_7_4_3_4.			

Test Step Dynamic Behaviour

Test Step Name: CCConfigTCH(sub:B_1; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3; mod:B_8; Freqg, Freqd:FRQPARA; arfcng, arfcnd:INTEGER)

Group: GSM_L3_MS_v4220/management/

Objective: Configure Tester for TCH/H or TCH/F depending on TCV_ChRate.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TCV_ChRate = C_Full]			1.
2		+FullRateCh_A_1(acttype, slot, tsc, ChMod(mod), Freqg, Freqd, arfcng, arfcnd, ta, babr, cch_con, bpm)			
3		[TCV_ChRate = C_Half]			2.
4		+HalfRateCh_A_1(sub, acttype, slot, tsc, ChMod(mod), Freqg, Freqd, arfcng, arfcnd, ta, babr, cch_con, bpm)			

Detailed Comments:
1. Full rate channel needed, to setup a physical channel as full rate traffic channel.
2. Half rate channel needed, to setup the physical channel as half rate traffic channel.

Test Step Dynamic Behaviour

Test Step Name: CCConfigTCH_B(sub:B_1; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3; mod:B_8; Freqg, Freqd:FRQPARA; arfcng, arfcnd:INTEGER)

Group: GSM_L3_MS_v4220/management/

Objective: Configure Tester for TCH/H or TCH/F depending on TCV_ChRate.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TCV_ChRate = C_Full]			1.
2		[arfcng <> C_Noarfcn]			
3		+FullRateCh_B_1(acttype, slot, tsc, ChMod(mod), FreqTCH(arfcng), FreqTCH(arfcnd), ta, babr, cch_con, bpm)			
4		[TSPC_PGSM OR TSPC_EGSM]			
5		(TCV_tch_arfcn:= arfcng)			
6		[TSPC_DCS]			
7		(TCV_tch_arfcn:= arfcnd)			
8		[arfcng = C_Noarfcn]			
9		+FullRateCh_B_1(acttype, slot, tsc, ChMod(mod), Freqg, Freqd, ta, babr, cch_con, bpm)			
10		[TCV_ChRate = C_Half]			2.
11		[arfcng <> C_Noarfcn]			
12		+HalfRateCh_B_1(sub, acttype, slot, tsc, ChMod(mod), FreqTCH(arfcng), FreqTCH(arfcnd), ta, babr, cch_con, bpm)			
13		[TSPC_PGSM OR TSPC_EGSM]			
14		(TCV_tch_arfcn:= arfcng)			
15		[TSPC_DCS]			
16		(TCV_tch_arfcn:= arfcnd)			
17		[arfcng = C_Noarfcn]			
18		+HalfRateCh_B_1(sub, acttype, slot, tsc, ChMod(mod), Freqg, Freqd, ta, babr, cch_con, bpm)			

Detailed Comments:
1. Full rate channel needed, to setup a physical channel as full rate traffic channel.
2. Half rate channel needed, to setup the physical channel as half rate traffic channel.

Test Step Dynamic Behaviour

Test Step Name: CCConfigTCH_nociph(sub:B_1; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3; mod:B_8; Freqg, Freqd:FRQPARA; arfcng, arfcnd:INTEGER)

Group: GSM_L3_MS_v4220/management/

Objective: Configure Tester for TCH/H or TCH/F depending on TCV_ChRate.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TCV_ChRate = C_Full]			1.
2		[arfcng <> C_Noarfcn]			
3		+FullRateCh_A_1_nociph(acttype, slot, tsc, ChMod(mod), FreqTCH(arfcng), FreqTCH(arfcnd), ta, babr, cch_con, bpm)			
4		[TSPC_PGSM OR TSPC_EGSM]			
5		(TCV_tch_arfcn:= arfcng)			
6		[TSPC_DCS]			
7		(TCV_tch_arfcn:= arfcnd)			
8		[arfcng = C_Noarfcn]			
9		+FullRateCh_A_1_nociph(acttype, slot, tsc, ChMod(mod), Freqg, Freqd, ta, babr, cch_con, bpm)			
10		[TCV_ChRate = C_Half]			2.
11		[arfcng <> C_Noarfcn]			
12		+HalfRateCh_A_1_nociph(sub, acttype, slot, tsc, ChMod(mod), FreqTCH(arfcng), FreqTCH(arfcnd), ta, babr, cch_con, bpm)			
13		[TSPC_PGSM OR TSPC_EGSM]			
14		(TCV_tch_arfcn:= arfcng)			
15		[TSPC_DCS]			
16		(TCV_tch_arfcn:= arfcnd)			
17		[arfcng = C_Noarfcn]			
18		+HalfRateCh_A_1_nociph(sub, acttype, slot, tsc, ChMod(mod), Freqg, Freqd, ta, babr, cch_con, bpm)			

Detailed Comments:
1. Full rate channel needed, to setup a physical channel as full rate traffic channel.
2. Half rate channel needed, to setup the physical channel as half rate traffic channel.

Test Step Dynamic Behaviour

Test Step Name: CCConfigTCH_pwr(sub:B_1; acttypeT:BITSTRING; slotT:SN; tscT:TSC; ta:TA; babr, cch_con, bpm:B_3; chmod:B_8; Freqg, Freqd:FRQPARA; mspwr: INTEGER)

Group: GSM_L3_MS_v4220/management/

Objective: Configure Tester for TCH/H or TCH/F depending on TCV_ChRate. Set power level according to parameter.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TCV_ChRate = C_Full]			1.
2		(TCV_chTch := C_FACCHF_A_1, TCV_sacchTch := C_SACCHF_A_1, TCV_chtype := '00001'B)			
3		[TSPC_PGSM OR TSPC_EGSM]			
4		+Config_FACCHF_A_1(63, mspwr, ChMod(chmod), acttypeT, slotT, tscT, Freqg, C_TCHF_ACCHF_1, ta, babr, cch_con, bpm)			
5		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
6		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			
7		[TSPC_DCS]			
8		+Config_FACCHF_A_1(63, mspwr, ChMod(chmod), acttypeT, slotT, tscT, Freqd, C_TCHF_ACCHF_1, ta, babr, cch_con, bpm)			
9		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
10		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			
11		[TCV_ChRate = C_Half]			2.
12		(TCV_chTch := OC_SubchOfFacchh(sub, C_CellA, 1), TCV_sacchTch := OC_SubchOfSacchh(sub, C_CellA, 1))			
13		[TSPC_PGSM OR TSPC_EGSM]			
14		+Config_FACCHH_A_1(63, mspwr, ChMod(chmod), acttypeT, slotT, tscT, Freqg, C_TCHH_ACCHH_1, ta, babr, cch_con, bpm)			
15		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
16		+localtree_selTCHH			
17		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			
18		[TSPC_DCS]			
19		+Config_FACCHH_A_1(63, mspwr, ChMod(chmod), acttypeT, slotT, tscT, Freqd, C_TCHH_ACCHH_1, ta, babr, cch_con, bpm)			
20		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
21		+localtree_selTCHH			
22		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			
		localtree_selTCHH			
23		[sub = '0'B]			
24		(TCV_chtype := '00010'B)			
25		[sub = '1'B]			
26		(TCV_chtype := '00011'B)			

Detailed Comments:
1. Full rate channel needed, to setup a physical channel as full rate traffic channel.
2. Half rate channel needed, to setup the physical channel as half rate traffic channel.

Test Step Dynamic Behaviour					
Test Step Name:	Config_CBMS_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH, SDCCH4 and CBCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, C_FCCH_A, C_SCH_A, C_BCCH_A_1, C_PCH_A_1, C_AGCH_A_1, C_RACH_A_1, C_SDCCH4_A, C_SACCHC4_A, C_CBCH_A))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_A, C_SDCCH40_A, C_SDCCH41_A, C_SDCCH42_A, C_SDCCH43_A, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_A, C_SACCHC40_A, C_SACCHC41_A, C_SACCHC42_A, C_SACCHC43_A, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_FACCHF_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHF_ACCH's of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_FACCHF_A_1, C_SACCHF_A_1, "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_FACCHF_A_2(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHF_ACCH's for instance 2 of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_FACCHF_A_2, C_SACCHF_A_2, "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_FACCHF_B_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHF_ACCH's of cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_FACCHF_B_1, C_SACCHF_B_1, "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_FACCHF_H_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHF_ACCH's of cell H.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellH, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_FACCHF_H_1, C_SACCHF_H_1, "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_FACCHH_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHH_ACCH's of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_FACCHH_A_1, C_SACCHH_A_1, "dummy"))			
2		(TCV_Null := OM_Assoc(C_FACCHH_A_1, C_FACCHH0_A_1, C_FACCHH1_A_1, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_SACCHH_A_1, C_SACCHH0_A_1, C_SACCHH1_A_1, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_FACCHH_A_2(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHH_ACCH's of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_FACCHH_A_2, C_SACCHH_A_2, "dummy"))			
2		(TCV_Null := OM_Assoc(C_FACCHH_A_2, C_FACCHH0_A_2, C_FACCHH1_A_2, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_SACCHH_A_2, C_SACCHH0_A_2, C_SACCHH1_A_2, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_FACCHH_B_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHH_ACCH's of cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_FACCHH_B_1, C_SACCHH_B_1, "dummy"))			
2		(TCV_Null := OM_Assoc(C_FACCHH_B_1, C_FACCHH0_B_1, C_FACCHH1_B_1, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_SACCHH_B_1, C_SACCHH0_B_1, C_SACCHH1_B_1, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, C_FCCH_A, C_SCH_A, C_BCCH_A_1, C_PCH_A_1, C_AGCH_A_1, C_RACH_A_1, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_A_2(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map the second BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", C_BCCH_A_2, C_PCH_A_2, C_AGCH_A_2, C_RACH_A_2, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_A_3(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map the third BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", C_BCCH_A_3, C_PCH_A_3, C_AGCH_A_3, C_RACH_A_3, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_A_4(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map the fourth BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", C_BCCH_A_4, C_PCH_A_4, C_AGCH_A_4, C_RACH_A_4, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_B_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, C_FCCH_B, C_SCH_B, C_BCCH_B_1, C_PCH_B_1, C_AGCH_B_1, C_RACH_B_1, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_B_ho_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3; td, fn:INTEGER)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell B with controlable td and fn.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, td, fn, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, C_FCCH_B, C_SCH_B, C_BCCH_B_1, C_PCH_B_1, C_AGCH_B_1, C_RACH_B_1, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, C_FCCH_A, C_SCH_A, C_BCCH_A_1, C_PCH_A_1, C_AGCH_A_1, C_RACH_A_1, C_SDCCH4_A, C_SACCHC4_A, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_A, C_SDCCH40_A, C_SDCCH41_A, C_SDCCH42_A, C_SDCCH43_A, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_A, C_SACCHC40_A, C_SACCHC41_A, C_SACCHC42_A, C_SACCHC43_A, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_B_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, C_FCCH_B, C_SCH_B, C_BCCH_B_1, C_PCH_B_1, C_AGCH_B_1, C_RACH_B_1, C_SDCCH4_B, C_SACCHC4_B, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_B, C_SDCCH40_B, C_SDCCH41_B, C_SDCCH42_B, C_SDCCH43_B, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_B, C_SACCHC40_B, C_SACCHC41_B, C_SACCHC42_B, C_SACCHC43_B, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_B_ho_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3; td, fn:INTEGER)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B with controlable ta and fn.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, td, fn, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, C_FCCH_B, C_SCH_B, C_BCCH_B_1, C_PCH_B_1, C_AGCH_B_1, C_RACH_B_1, C_SDCCH4_B, C_SACCHC4_B, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_B, C_SDCCH40_B, C_SDCCH41_B, C_SDCCH42_B, C_SDCCH43_B, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_B, C_SACCHC40_B, C_SACCHC41_B, C_SACCHC42_B, C_SACCHC43_B, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_C_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell C.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellC, C_FCCH_C, C_SCH_C, C_BCCH_C_1, C_PCH_C_1, C_AGCH_C_1, C_RACH_C_1, C_SDCCH4_C, C_SACCHC4_C, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_C, C_SDCCH40_C, C_SDCCH41_C, C_SDCCH42_C, C_SDCCH43_C, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_C, C_SACCHC40_C, C_SACCHC41_C, C_SACCHC42_C, C_SACCHC43_C, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_D_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell D				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellD, C_FCCH_D, C_SCH_D, C_BCCH_D_1, C_PCH_D_1, C_AGCH_D_1, C_RACH_D_1, C_SDCCH4_D, C_SACCHC4_D, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_D, C_SDCCH40_D, C_SDCCH41_D, C_SDCCH42_D, C_SDCCH43_D, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_D, C_SACCHC40_D, C_SACCHC41_D, C_SACCHC42_D, C_SACCHC43_D, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_E_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell E				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellE, C_FCCH_E, C_SCH_E, C_BCCH_E_1, C_PCH_E_1, C_AGCH_E_1, C_RACH_E_1, C_SDCCH4_E, C_SACCHC4_E, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_E, C_SDCCH40_E, C_SDCCH41_E, C_SDCCH42_E, C_SDCCH43_E, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_E, C_SACCHC40_E, C_SACCHC41_E, C_SACCHC42_E, C_SACCHC43_E, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_F_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell F				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellF, C_FCCH_F, C_SCH_F, C_BCCH_F_1, C_PCH_F_1, C_AGCH_F_1, C_RACH_F_1, C_SDCCH4_F, C_SACCHC4_F, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_F, C_SDCCH40_F, C_SDCCH41_F, C_SDCCH42_F, C_SDCCH43_F, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_F, C_SACCHC40_F, C_SACCHC41_F, C_SACCHC42_F, C_SACCHC43_F, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_G_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell G				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellG, C_FCCH_G, C_SCH_G, C_BCCH_G_1, C_PCH_G_1, C_AGCH_G_1, C_RACH_G_1, C_SDCCH4_G, C_SACCHC4_G, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_G, C_SDCCH40_G, C_SDCCH41_G, C_SDCCH42_G, C_SDCCH43_G, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_G, C_SACCHC40_G, C_SACCHC41_G, C_SACCHC42_G, C_SACCHC43_G, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_H_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell H				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellH, C_FCCH_H, C_SCH_H, C_BCCH_H_1, C_PCH_H_1, C_AGCH_H_1, C_RACH_H_1, C_SDCCH4_H, C_SACCHC4_H, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_H, C_SDCCH40_H, C_SDCCH41_H, C_SDCCH42_H, C_SDCCH43_H, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_H, C_SACCHC40_H, C_SACCHC41_H, C_SACCHC42_H, C_SACCHC43_H, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH8_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as SDCCH8 channel for instance 1 of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_SDCCH8_A_1, C_SACCHC8_A_1, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH8_A_1, C_SDCCH80_A_1, C_SDCCH81_A_1, C_SDCCH82_A_1, C_SDCCH83_A_1, C_SDCCH84_A_1, C_SDCCH85_A_1, C_SDCCH86_A_1, C_SDCCH87_A_1, C_SACCHC8_A_1, C_SACCHC80_A_1, C_SACCHC81_A_1, C_SACCHC82_A_1, C_SACCHC83_A_1, C_SACCHC84_A_1, C_SACCHC85_A_1, C_SACCHC86_A_1, C_SACCHC87_A_1))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH8_A_2(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as SDCCH8 channel for instance 2 of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_SDCCH8_A_2, C_SACCHC8_A_2, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH8_A_2, C_SDCCH80_A_2, C_SDCCH81_A_2, C_SDCCH82_A_2, C_SDCCH83_A_2, C_SDCCH84_A_2, C_SDCCH85_A_2, C_SDCCH86_A_2, C_SDCCH87_A_2, C_SACCHC8_A_2, C_SACCHC80_A_2, C_SACCHC81_A_2, C_SACCHC82_A_2, C_SACCHC83_A_2, C_SACCHC84_A_2, C_SACCHC85_A_2, C_SACCHC86_A_2, C_SACCHC87_A_2))			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: Config_SDCCH8_B_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_L3_MS_v4220/management/

Objective: To set one physical channel used as SDCCH8 channel for instance 1 of cell B.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_SDCCH8_B_1, C_SACCHC8_B_1, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH8_B_1, C_SDCCH80_B_1, C_SDCCH81_B_1, C_SDCCH82_B_1, C_SDCCH83_B_1, C_SDCCH84_B_1, C_SDCCH85_B_1, C_SDCCH86_B_1, C_SDCCH87_B_1, C_SACCHC8_B_1, C_SACCHC80_B_1, C_SACCHC81_B_1, C_SACCHC82_B_1, C_SACCHC83_B_1, C_SACCHC84_B_1, C_SACCHC85_B_1, C_SACCHC86_B_1, C_SACCHC87_B_1))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: CombinedBCCH_A_CBMS(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_L3_MS_v4220/management/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.

Default: OtherEvents

Comments: time slot = 0, ARFCN = 20 (GSM900) or ARFCN = 590 (DCS1800) cell A for RR testing.

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_CBMS_A_1(par_bspwr,mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_CBMS_A_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			

Detailed Comments:
1. For P-GSM900.
2. For DCS1800.

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_A_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_A_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_B(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_B_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_B_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_B_ho(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3; td, fn:INTEGER)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_B_ho_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm, td, fn)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_B_ho_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm, td, fn)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_C(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell C.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_C_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_C_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_D(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell D.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_D_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_D_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_E(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell E.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_E_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_E_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_F(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell F.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_F_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_F_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_G(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2: INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell G.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_G_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_G_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour

Test Step Name: CombinedBCCH_G_sp(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2, arfcn3:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_L3_MS_v4220/management/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell G.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM]			1.
2		+Config_SDCCH4_G_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_EGSM]			1.
4		+Config_SDCCH4_G_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
5		[TSPC_DCS]			2.
6		+Config_SDCCH4_G_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn3), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			

Detailed Comments:
1. For P-GSM900.
2. For DCS1800.

Test Step Dynamic Behaviour

Test Step Name: CombinedBCCH_H(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_L3_MS_v4220/management/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell H.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_H_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_H_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			

Detailed Comments:
1. For P-GSM900.
2. For DCS1800.

Test Step Dynamic Behaviour					
Test Step Name:	NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as FCHH_SCH_BCCH_CCCH for cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_BCCH_CCCH_A_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_A_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	NonCombinedBCCH_A_2(acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map the second BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_BCCH_CCCH_A_2(63, 19, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnA), C_BCCH_CCCH_2, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_A_2(63, 15, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnAd), C_BCCH_CCCH_2, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	NonCombinedBCCH_A_3(acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set a physical channel and map the third BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Comments:	time slot = 4, ARFCN = TSPX_BCCHcarrierA				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_BCCH_CCCH_A_3(63, 19, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnA), C_BCCH_CCCH_3, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_A_3(63, 15, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnAd), C_BCCH_CCCH_3, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour

Test Step Name: NonCombinedBCCH_A_4(acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_L3_MS_v4220/management/

Objective: To set a physical channel and map the fourth BCCH, CCCH onto the physical channel which represents cell A.

Default: OtherEvents

Comments: time slot = 6, ARFCN = TSPX_BCCHcarrierA

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_BCCH_CCCH_A_4(63, 19, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnA), C_BCCH_CCCH_4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_A_4(63, 15, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnAd), C_BCCH_CCCH_4, ta, babr, cch_con, bpm)			

Detailed Comments:
1. For P-GSM900.
2. For DCS1800.

Test Step Dynamic Behaviour

Test Step Name: NonCombinedBCCH_B(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_L3_MS_v4220/management/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH onto the physical channel which represents cell B for RR testing.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM]			1.
2		+Config_BCCH_CCCH_B_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_B_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm)			

Detailed Comments:
1. For P-GSM900.
1.1 Frequenz and cell_id for cell B in HO cases
2. For DCS1800.
2.1 Frequenz and cell_id for cell B in HO cases

Test Step Dynamic Behaviour

Test Step Name: NonCombinedBCCH_B_ho(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3; td, fn:INTEGER)

Group: GSM_L3_MS_v4220/management/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH onto the physical channel which represents cell B for RR testing.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM]			1.
2		+Config_BCCH_CCCH_B_ho_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm, td, fn)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_B_ho_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm, td, fn)			

Detailed Comments:
1. For P-GSM900.
1.1 Frequenz and cell_id for cell B in HO cases
2. For DCS1800.
2.1 Frequenz and cell_id for cell B in HO cases

Test Step Dynamic Behaviour

Test Step Name: FullRateCh_A_1(acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg, Freqd:FRQPARA; arfcng, arfcnd:INTEGER; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_L3_MS_v4220/management/

Objective: To set one physical channel used as TCHF_ACCH's for instance 1 of cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[arfcng <> C_Noarfcn]			Non hopping
2		+FullRateCh_A_1_nociph(acttypeT, slotT, tscT, chmod, FreqTCH(arfcng), FreqTCH(arfcnd), ta, babr, cch_con, bpm)			
3		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			
4		[TSPC_PGSM OR TSPC_EGSM]			
5		(TCV_tch_arfcn:= arfcng)			
6		[TSPC_DCS]			
7		(TCV_tch_arfcn:= arfcnd)			
8		[arfcng = C_Noarfcn]			Hopping
9		+FullRateCh_A_1_nociph(acttypeT, slotT, tscT, chmod, Freqg, Freqd, ta, babr, cch_con, bpm)			
10		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: FullRateCh_A_1_nociph(acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg, Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_L3_MS_v4220/management/

Objective: To set one physical channel used as TCHF_ACCH's for instance 1 of cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_chTch := C_FACCHF_A_1, TCV_sacchTch := C_SACCHF_A_1, TCV_chtype := '00001'B)			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_FACCHF_A_1(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_TCHF_ACCHF_1, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
5		[TSPC_DCS]			
6		+Config_FACCHF_A_1(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_TCHF_ACCHF_1, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: FullRateCh_A_2(acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_L3_MS_v4220/management/

Objective: To set one physical channel used as TCHF_ACCH's for instance 2 of cell A for TC_26_6_13_1.

Default: OtherEvents

Comments: used as after time channel.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+FullRateCh_A_2_nociph(acttypeT, slotT, tscT, chmod, Freqg,Freqd, ta, babr, cch_con, bpm)			
2		(TCV_Null := OM_CphMd(TCV_chTch1, CphMod_01, TCV_CphKey))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: FullRateCh_A_2_nociph(acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_L3_MS_v4220/management/

Objective: To set one physical channel used as TCHF_ACCH's for instance 2 of cell A for TC_26_6_13_1.

Default: OtherEvents

Comments: used as after time channel.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_chTch1 := C_FACCHF_A_2, TCV_sacchTch1 := C_SACCHF_A_2)			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_FACCHF_A_2(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_TCHF_ACCHF_2, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)			
5		[TSPC_DCS]			
6		+Config_FACCHF_A_2(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_TCHF_ACCHF_2, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)			

Detailed Comments:

Test Step Dynamic Behaviour					
Test Step Name:	FullRateCh_B_1(acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHF_ACCH's for instance 1 of cell B.				
Default:	OtherEvents				
Comments:	FullRateCh_B_1.				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+FullRateCh_B_1_nociph(acttypeT, slotT, tscT, chmod, Freqg, Freqd, ta, babr, cch_con, bpm)			
2		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	FullRateCh_B_1_nociph(acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHF_ACCH's for instance 1 of cell B.				
Default:	OtherEvents				
Comments:	Used in TC_26_6_13_5 for after time channel				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_chTch := C_FACCHF_B_1, TCV_sacchTch := C_SACCHF_B_1)			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_FACCHF_B_1(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_TCHF_ACCHF_1, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5_B, TCV_sysinfo6_B)			
5		[TSPC_DCS]			
6		+Config_FACCHF_B_1(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_TCHF_ACCHF_1, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5_B, TCV_sysinfo6_B)			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	HalfRateCh_A_1(sub:B_1; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg, Freqd:FRQPARA; arfcng, arfcnd:INTEGER; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHH_ACCH's for instance 1 of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[arfcng <> C_Noarfcn]			not hopping
2		+HalfRateCh_A_1_nociph(sub, acttypeT, slotT, tscT, chmod, FreqTCH(arfcng), FreqTCH(arfcnd), ta, babr, cch_con, bpm)			
3		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			
4		[TSPC_PGSM OR TSPC_EGSM]			
5		(TCV_tch_arfcn:= arfcng)			
6		[TSPC_DCS]			
7		(TCV_tch_arfcn:= arfcnd)			
8		[arfcng = C_Noarfcn]			hopping
9		+HalfRateCh_A_1_nociph(sub, acttypeT, slotT, tscT, chmod, Freqg, Freqd, ta, babr, cch_con, bpm)			
10		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	HalfRateCh_A_1_nociph(sub:B_1; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg, Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHH_ACCH's for instance 1 of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_chTch := OC_SubchOfFacchh(sub, C_CellA, 1), TCV_sacchTch := OC_SubchOfSacchh(sub, C_CellA, 1))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_FACCHH_A_1(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_TCHH_ACCHH_1, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
5		+localtree			
6		[TSPC_DCS]			
7		+Config_FACCHH_A_1(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_TCHH_ACCHH_1, ta, babr, cch_con, bpm)			
8		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
9		+localtree			
		localtree			
10		[sub = '0'B]			
11		(TCV_chtype := '00010'B)			
12		[sub = '1'B]			
13		(TCV_chtype := '00011'B)			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	HalfRateCh_A_2(bstring:B_1; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHH_ACCH's for instance 2 of cell A for TC_26_6_13_1.				
Default:	OtherEvents				
Comments:	used for after time channel.				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+HalfRateCh_A_2_nociph(bstring, acttypeT, slotT, tscT, chmod, Freqg, Freqd, ta, babr, cch_con, bpm)			
2		(TCV_Null := OM_CphMd(TCV_chTch1, CphMod_01, TCV_CphKey))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	HalfRateCh_A_2_nociph(bstring:B_1; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHH_ACCH's for instance 2 of cell A for TC_26_6_13_1.				
Default:	OtherEvents				
Comments:	used for after time channel.				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_chTch1 := OC_SubchOfFacchh(bstring, C_CellA, 2), TCV_sacchTch1 := OC_SubchOfSacchh(bstring, C_CellA, 2))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_FACCHH_A_2(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_TCHH_ACCHH_2, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)			
5		[TSPC_DCS]			
6		+Config_FACCHH_A_2(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_TCHH_ACCHH_2, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	HalfRateCh_B_1(bstring:B_1; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHH_ACCH's for instance 1 of cell B				
Default:	OtherEvents				
Comments:	used as after time channel assigned by handover command				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+HalfRateCh_B_1_nociph(bstring, acttypeT, slotT, tscT, chmod, Freqg, Freqd, ta, babr, cch_con, bpm)			
2		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	HalfRateCh_B_1_nociph(bstring:B_1; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as TCHH_ACCH's for instance 1 of cell B				
Default:	OtherEvents				
Comments:	used as after time channel assigned by handover command				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_chTch := OC_SubchOfFacchh(bstring, C_CellB, 1), TCV_sacchTch := OC_SubchOfSacchh(bstring, C_CellB, 1))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_FACCHH_B_1(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_TCHH_ACCHH_1, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5_B, TCV_sysinfo6_B)			
5		[TSPC_DCS]			
6		+Config_FACCHH_B_1(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_TCHH_ACCHH_1, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5_B, TCV_sysinfo6_B)			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	SDCCH8_A_1(bstring:BITSTRING; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as SDCCH8 channel for instance 1 of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+SDCCH8_A_1_nociph(bstring, acttypeT, slotT, tscT, chmod, Freqg, Freqd, ta, babr, cch_con, bpm)			
2		(TCV_Null := OM_CphMd(TCV_ch, CphMod_01, TCV_CphKey))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	SDCCH8_A_1_nociph(bstring:B_3; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as SDCCH8 channel for instance 1 of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_ch := OC_SubchOfSdcch8(bstring, C_CellA, 1), TCV_sacch8 := OC_SubchOfSacch8(bstring, C_CellA, 1))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_SDCCH8_A_1(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_SDCCH8_SACCHC8_1, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacch8, TCV_sysinfo5, TCV_sysinfo6)			
5		[TSPC_DCS]			
6		+Config_SDCCH8_A_1(63, 19, chmod, acttypeT, slotT, tscT, Freqd, C_SDCCH8_SACCHC8_1, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacch8, TCV_sysinfo5, TCV_sysinfo6)			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: SDCCH8_A_2_nociph(bstring:B_3; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_L3_MS_v4220/management/

Objective: To set one physical channel used as hopping SDCCH8 channel for instance 2 of cell A, for TC_26_6_13_1.

Default: OtherEvents

Comments: hopping parameters are from PIXIT, used as before time channel.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_chTch := OC_SubchOfSdcch8(bstring, C_CellA, 2), TCV_sacchTch := OC_SubchOfSacch8(bstring, C_CellA, 2))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_SDCCH8_A_2(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_SDCCH8_SACCHC8_2, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
5		[TSPC_DCS]			
6		+Config_SDCCH8_A_2(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_SDCCH8_SACCHC8_2, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SDCCH8_B_1(bstring:B_3; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_L3_MS_v4220/management/

Objective: To set one physical channel used as hopping SDCCH8 channel for instance 1 of cell B.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+SDCCH8_B_1_nociph(bstring, acttypeT, slotT, tscT, chmod, Freqg, Freqd, ta, babr, cch_con, bpm)			
2		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			

Detailed Comments:

Test Step Dynamic Behaviour					
Test Step Name:	SDCCH8_B_1_nociph(bstring:B_3; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_L3_MS_v4220/management/				
Objective:	To set one physical channel used as hopping SDCCH8 channel for instance 1 of cell B For TC_26_6_13_5.				
Default:	OtherEvents				
Comments:	hopping parameters are from PIXIT, used by handover command after time channel.				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_chTch := OC_SubchOfSdcch8(bstring, C_CellB, 1), TCV_sacchTch := OC_SubchOfSacch8(bstring, C_CellB, 1))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_SDCCH8_B_1(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_SDCCH8_SACCHC8_1, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5_B, TCV_sysinfo6_B)			
5		[TSPC_DCS]			
6		+Config_SDCCH8_B_1(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_SDCCH8_SACCHC8_1, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5_B, TCV_sysinfo6_B)			
Detailed Comments:					

Test Step Group Miscellaneous

Test Step Dynamic Behaviour					
Test Step Name:	Adjust_gsmanddcs_powerlvl(powerlevel1,powerlevel2:INTEGER; pdu_ass: ASS_CMD_PDU)				
Group:	GSM_L3_MS_v4220/Miscellaneous/				
Objective:					
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			
2		[powerlevel1 <> 0]			
3		(pdu_ass.pcmd := Pcmd_19(INT_TO_BIT(powerlevel1,5)))			
4		[powerlevel1 = 0]			
5		[TSPC_DCS]			
6		(pdu_ass.pcmd := Pcmd_19(INT_TO_BIT(powerlevel2,5)))			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: AOC_CHK_FAC(ti:TI)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: Reception of FACILITY and callinh check_AOCCackTime
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(ti))		
2		(TCV_Fn := OM_ReturnFn(TCV_chTch))			1.
3	L_0836	L?DL_DatInFac (TCV_Fn1 := DL_DatInFac.fn) CANCEL T_dly1	FacilityRcv(FacilityPdu_26(ti, facilityIErcv(FwdChAdvRslt_01)))	(P)	2.
4		+Check_Time(C_T_Wait_FAC)			
5	L_0837	?TIMEOUT T_dly1		(F)	
6		+PostLinkRelEnd(TCV_chTch)			
7	L_0838	L?DL_DatInFac (TCV_Fn1 := DL_DatInFac.fn)	FacilityRcv(FacilityPdu_26(ti, facilityIErcv(FwdChAdvRslt_01)))	(P)	2.
8		(TCV_Fn := OM_ReturnFn(TCV_chTch))			1.
9		L?DL_DatInConnAck CANCEL T_dly1	ConnAckRcv(ConnectAck_02(ti))		
10		+Check_Time(C_T_Wait_FAC)			
11	L_0839	?TIMEOUT T_dly1		(F)	
12		+PostLinkRelEnd(TCV_chTch)			
13	L_0840	?TIMEOUT T_dly1		(F)	
14		+PostLinkRelEnd(TCV_chTch)			

Detailed Comments:
1. Get the first frame number for transmitting of CONNECT message containing AOC information.
2. TCV_Fn1 contains the first frame number of the received FACILITY message with AOC acknowledge from mobile.

Test Step Dynamic Behaviour

Test Step Name: AssCh_complete(oldch,newch:LOGICCH; pdu_ass: ASS_CMD_PDU)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To provide a generic test step to Assign a traffic channel. Assign complete expected and verdict is PASS.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_DatRqAssCmd START T_dlyAss	AssCmd(oldch,pdu_ass)		
2	L_0841	?TIMEOUT T_dlyAss		(F)	
3		+PostLinkRelEnd(oldch)			
4		L?DL_EstIn (TCV_FnAss := DL_EstIn.fn) CANCEL T_dlyAss	DLEstInd(newch)		
5	L_0842	L?DL_DatInAssCom	AssCmp(newch, AsgnCmp_02)	(P)	
6		L!MDL_RelRq	MDLRelReq(oldch)		
7		L?DL_EstIn CANCEL T_dlyAss	DLEstInd(oldch)		
8	L_0843	L?DL_DatInAssfl	AssFl_any_cau(oldch)	(F)	
9		+PostLinkRelEnd(oldch)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: AssCh_failure(ch:LOGICCH; pdu_ass: ASS_CMD_PDU; any:BOOLEAN)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective: To provide a generic test step to Assign a traffic channel. Assign failure expected and verdict is PASS.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_DatRqAssCmd START T_dlyAss(4000)	AssCmd(ch,pdu_ass)		
2		[any]			
3	L_0844	?TIMEOUT T_dlyAss		(F)	
4		+PostLinkRelEnd(ch)			
5		L?DL_EstIn (TCV_FnAss := DL_EstIn.fn) CANCEL T_dlyAss	DLEstInd(ch)		
6	L_0845	L?DL_DatInAssfl	AssFl_any_cau(ch)	(P)	
7		[NOT any]			
8	L_0846	?TIMEOUT T_dlyAss		(F)	
9		+PostLinkRelEnd(ch)			
10		L?DL_EstIn (TCV_FnAss := DL_EstIn.fn) CANCEL T_dlyAss	DLEstInd(ch)		
11	L_0847	L?DL_DatInAssfl	AssFl_02(ch)	(P)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: AssCmdGen(cell:CellID; rate:IA5String; slot:SN; tsc:TSC; sch:B_1)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To generate the ASSIGNMENT COMMAND message.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[rate <> C_Half]			Full rate
2		(TCV_Tchtype := '00001'B)			
3		+ltree_Asgn			
4		[rate = C_Half]			Half rate
5		[sch = '0'B]			
6		(TCV_Tchtype := '00010'B)			
7		+ltree_Asgn			
8		[sch = '1'B]			
9		(TCV_Tchtype := '00011'B)			
10		+ltree_Asgn			
		ltree_Asgn			
11		[cell = C_CellA]			
12		+ltree_AsgnTchA			
13		[cell = C_CellB]			
14		+ltree_AsgnTchB			
		ltree_AsgnTchA			
15		[TSPC_PGSM OR TSPC_EGSM]			
16		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, slot, tsc, 7, C_arfcn_tchA, CellChDes_Omit, TCV_ChMod, StartingTm_omit, CphMod_omit))			
17		[TSPC_DCS]			
18		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, slot, tsc, 3, C_arfcn_tchAd, CellChDes_Omit, TCV_ChMod, StartingTm_omit, CphMod_omit))			
		ltree_AsgnTchB			
19		[TSPC_PGSM OR TSPC_EGSM]			
20		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, slot, tsc, 7, C_arfcn_tchB, CellChDes_Omit, TCV_ChMod, StartingTm_omit, CphMod_omit))			
21		[TSPC_DCS]			
22		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, slot, tsc, 3, C_arfcn_tchBd, CellChDes_Omit, TCV_ChMod, StartingTm_omit, CphMod_omit))			

Detailed Comments: The test step makes the assumption that the channel rate is known when the test step is called.

Test Step Dynamic Behaviour

Test Step Name: AssCmdGen_fh(rate:IA5String; slot:SN; tsc:TSC; sch:B_1; par_pwlvlg, par_pwlvld:INTEGER; maio:MAIO; hsn:HSN; frql_g, frql_d:FRQL; cchd_g, cchd_d:CCHD; chmod:CHMOD; ma_g, ma_d:MA; Cphms:CPHMS)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective: To generate a ASSIGNMENT COMMAND message in case of hopping without starting time.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[rate <> C_Half]			Full rate
2		(TCV_Tchtype := '00001'B)			
3		+ltree_Asgn			
4		[rate = C_Half]			Half rate
5		[sch = '0'B]			
6		(TCV_Tchtype := '00010'B)			
7		+ltree_Asgn			
8		[sch = '1'B]			
9		(TCV_Tchtype := '00011'B)			
10		+ltree_Asgn			
		ltree_Asgn			
11		[TSPC_PGSM OR TSPC_EGSM]			
12		(TCV_AssCmd := AsgnCmd_fh(slot, tsc, TCV_Tchtype, par_pwlvlg, maio, hsn, frql_g, cchd_g, chmod, ma_g, Cphms))			
13		[TSPC_DCS]			
14		(TCV_AssCmd := AsgnCmd_fh(slot, tsc, TCV_Tchtype, par_pwlvld, maio, hsn, frql_d, cchd_d, chmod, ma_d, Cphms))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Authentication(ch: LOGICCH; cksn: BITSTRING; rand:RAND)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective:

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_DatRqAuthRq	AuthReq(ch, AuthRequest(cksn, rand))		
2		L?DL_DatInAuthRes (TCV_Sres := DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
3		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, rand))			
4	L_0848	[NOT TCV_Res]		(F)	1)
5	L_0849	[TCV_Res]		(P)	

Detailed Comments: Authentication fails.

Test Step Dynamic Behaviour

Test Step Name: CallProcGen(setup:SETUP_MO_PDU; srv:SERVICES)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective: To generate a CALL PROCEEDING message which is compatible with input value 'setup'..

Default: OtherEvents

Comments: The generated CALL PROCEEDING is in TCV_CallProc. This test step can handle the following dual mode services: the first BCAP is speech or the second BCAP is speech

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_CallProc := CallProced_03, TCV_CallProc.ti := TCV_TI)			
2		[(srv = C_AltSpchData) OR (srv = C_SpchData) OR (srv = C_AltSpchFax)]			dual service
3		[setup.bcap1.itc = C_Speech]			1st BCAP is speech
4		(TCV_ce3 := setup.bcap2.ce)			
5		[(TCV_ce3 <> C_BothT) OR(TCV_ce3 <> C_BothNT)]			1.
6		[(TCV_ce3 = C_BothT) OR (TCV_ce3 = C_BothNT)]			2.
7		(TCV_CallProc.bcap1 := setup.bcap1, TCV_CallProc.bcap1.rchr := '00'B, TCV_CallProc.bcri := setup.bcri, TCV_CallProc.bcap2 := setup.bcap2, TCV_CallProc.bcap2.ce := INT_TO_BIT((BIT_TO_INT(setup.bcap2.ce) -2), 2), TCV_CallProc.bcap2.nirr := C_nirr_nomeaning)			
8		[NOT (setup.bcap1.itc = C_Speech)]			
9		(TCV_ce1 := setup.bcap1.ce)			
10		[(TCV_ce1 <> C_BothT) AND(TCV_ce1 <> C_BothNT)]			3.
11		[(TCV_ce1 = C_BothT) OR (TCV_ce1 = C_BothNT)]			4.
12		(TCV_CallProc.bcap1 := setup.bcap1, TCV_CallProc.bcap1.ce := INT_TO_BIT((BIT_TO_INT(setup.bcap1.ce) -2), 2), TCV_CallProc.bcap1.nirr := C_nirr_nomeaning, TCV_CallProc.bcri := setup.bcri, TCV_CallProc.bcap2 := setup.bcap2, TCV_CallProc.bcap2.rchr := '11'B)			
13		[(srv <> C_AltSpchData) AND(srv <> C_SpchData) AND(srv <> C_AltSpchFax)]			
14		[srv = C_EmgCall]			8.
15		[srv <> C_EmgCall]			
16		[setup.bcap1.itc = C_Speech]			5.
17		[setup.bcap1.itc <> C_Speech]			
18		(TCV_ce1 := setup.bcap1.ce)			
19		[(TCV_ce1 = C_BothT) OR (TCV_ce1 = C_BothNT)]			6.
20		(TCV_CallProc.bcap1 := setup.bcap1, TCV_CallProc.bcap1.ce := INT_TO_BIT((BIT_TO_INT(setup.bcap1.ce) -2), 2), TCV_CallProc.bcap1.nirr := C_nirr_nomeaning)			
21		[(TCV_ce1 <> C_BothT) OR(TCV_ce1 <> C_BothNT)]			7.

Detailed Comments:

1. BCAP1 is speech and BCAP2 does not reqire negotiation, the CallProc will contain no BCAP's
2. BCAP1 is speech and BCAP2 requires negotiation
3. BCAP2 is speech and BCAP1 does not reqire negotiation, the CallProc will contain no BCAP's
4. BCAP2 is speech and BCAP1 requires negotiation
5. BCAP2 does not exist and BCAP1 is speech, the CallProc will contain no BCAP's
6. BCAP2 does not exist and BCAP1 requires negotiation
7. BCAP2 does not exist and BCAP1 does not require negotiation
8. Emergency call, the CallProc will contain no BCAP

Test Step Dynamic Behaviour

Test Step Name: CCAssignTCH(slot:SN; tsc:TSC)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective: Send assign command TCH/H or TCH/F depending on TCV_ChRate and wait for establishment indication. This is used in CC testing.

Default: OtherEvents

Comments: The calling tree prepare three variables for the step: TCV_ChRate for the type of the channel, TCV_chMod with th channel mode IE for the channel, TCV_chTch contains the identifier fot the traffic channel and TCV_chtype the channel type.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Null := OM_ChMdModi(TCV_chTch, TCV_ChMod), TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			
2		+ltree_Asgn			1.
3		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			2. 3.
		ltree_Asgn			
4		[TSPC_PGSM OR TSPC_EGSM]			
5		(TCV_AssCmd := AsgnCmd_nfh(TCV_chtype, slot, tsc, 7, C_arfcn_tchA, CellChDes_Omit, TCV_ChMod, StartingTm_omit, CphMod_omit))			
6		[TSPC_DCS]			
7		(TCV_AssCmd := AsgnCmd_nfh(TCV_chtype, slot, tsc, 3, C_arfcn_tchAd, CellChDes_Omit, TCV_ChMod, StartingTm_omit, CphMod_omit))			

Detailed Comments:

1. To assign the suitable traffic channel to the MS.
2. ASSIGN COMMAND with channel mode, channel description and power command.
   The channel mode and type (TCH/H or TCH/F) and power command are supplied as parameter.
3. Different power command for DCS.

Test Step Dynamic Behaviour

Test Step Name: CCCH_group_Paging_group(ccd: CCD; imsi: HEXSTRING)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective: To calculate the CCCH group and the Paging group from the IMSI and the Control Channel Descriptor according to GSM 05.02, subclause 6.5.2.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[ccd.ccch_con = '000'B]			
2		(TCV_bs_cc_chans := 1)			
3		+localtree1			
4		[ccd.ccch_con = '001'B]			
5		(TCV_bs_cc_chans := 1)			
6		+localtree1			
7		[ccd.ccch_con = '010'B]			
8		(TCV_bs_cc_chans := 2)			
9		+localtree1			
10		[ccd.ccch_con = '100'B]			
11		(TCV_bs_cc_chans := 3)			
12		+localtree1			
13		[ccd.ccch_con = '110'B]			
14		(TCV_bs_cc_chans := 4)			
15		+localtree1			
		localtree1			
16		(TCV_bs_ag_blks_res := BIT_TO_INT(ccd.babr), TCV_bs_pa_mfrms := (2 + BIT_TO_INT(ccd.bpm)))			
17		[ccd.ccch_con = '001'B]			i.e. combined
18	L_0850	[(TCV_bs_ag_blks_res < 0) OR(TCV_bs_ag_blks_res > 2)]		I	Stop! A tester error.
19		[(TCV_bs_ag_blks_res >= 0) AND (TCV_bs_ag_blks_res <= 2)]			
20		(TCV_tmp := 3 - TCV_bs_ag_blks_res)			
21		+localtree2			
22		[ccd.ccch_con <> '001'B]			i.e. not combined
23	L_0851	[(TCV_bs_ag_blks_res < 0) OR(TCV_bs_ag_blks_res > 7)]		I	Stop! A tester error.
24		[(TCV_bs_ag_blks_res >= 0) AND (TCV_bs_ag_blks_res <= 7)]			
25		(TCV_tmp := 9 - TCV_bs_ag_blks_res)			
26		+localtree2			ini. TCV Pgg
		localtree2			
27		(TCV_tmp := TCV_tmp * TCV_bs_pa_mfrms)			
28		(TCV_Pgg := INT_TO_BIT((((OC_BCDtoInt(imsi, 3)) MOD(TCV_bs_cc_chans * TCV_tmp)) MOD TCV_tmp), 8), TCV_Ccchg := (((OC_BCDtoInt(imsi, 3)) MOD(TCV_bs_cc_chans * TCV_tmp)) / TCV_tmp))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: CCEstablishMO_SDCCH4(sub:B_2; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: Establish a MO SDCCH/4 connection. This is used in CC testing
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_19)		
2		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, sub, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		1.

Detailed Comments: 1. To assign a SDCCH/4.

Test Step Dynamic Behaviour

Test Step Name: CCEstablishMO_TCH(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: Establish a MO TCH/F or TCH/H connection. This is used in CC testing
Default: OtherEvents
Comments: The calling tree shall prepare one variable for the step: TCV_ChRate for the type of the channel. TCV_chTch contains the identifier for the traffic channeland TCV_cht the channel type.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_19)		
2		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
3		L!DL_UdatRqImmass	ImmAss(C_AGCH_A_1, ImmAsgn_nfh(TCV_Rr, TCV_Fn, TCV_chtype, slot, tsc, ta, TCV_tch_arfcn, C_normal_paging))		

Detailed Comments: 1. To assign TCH/F or TCH/H.

Test Step Dynamic Behaviour

Test Step Name: CCEstablishMT_SDCCH4(ta:TA; sub:B_2; cksn:B_3)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: Establish a MT SDCCH/4 connection. This is used in CC testing
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+CCPage			
2		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
3		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, sub, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		1.
4		L?DL_EstInPgRes	PagingRes(PagingRes_03(cksn))		
5		ACTIVATE(OtherEvents)			Restore Normal default

Detailed Comments: 1. To assign a SDCCH/4.

Test Step Dynamic Behaviour

Test Step Name: CCEstablishMT_TCH(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: Establish a MT TCH/F or TCH/H connection. This is used in CC testing
Default: OtherEvents
Comments: The calling tree shall prepare one variable for the step: TCV_ChRate for the type of the channel. TCV_chTch contains the identifier for the traffic channeland TCV_cht the channel type.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+CCPage			
2		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
3		L!DL_UdatRqImmass	ImmAss(C_AGCH_A_1, ImmAsgn_nfh(TCV_Rr, TCV_Fn, TCV_chtype, slot, tsc, ta, TCV_tch_arfcn, C_normal_paging))		
4		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
5		ACTIVATE(OtherEvents)			Restore Normal default

Detailed Comments: 1. To assign TCH/F or TCH/H.

Test Step Dynamic Behaviour

Test Step Name: CC_Est_MT_Call
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To perform the CC message exchange to establish a mobile station terminating call (late assignment).
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Setup_mt.sig := Signal_01)			
2		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		
3		L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))		1)
4		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
5		L?DL_DatInAlert	AlertRcv(AlertingInd_01(TI_01))		
6		(TCV_Null := OO_HookOff())			
7		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		

Detailed Comments: 1. If the MS supports the bearer capabilities, which are given in Setup message, it has to accept them. SETUP message for MT speech in the TTCN includes a bcap with speech as information transfer mode in octet 3. Any other octets above octet 3 in bcap are not transmitted (omitted). Because the MS has been correctly configured for telephony before testing an MT SETUP message with speech as information transfer mode in bcap should not disturb any MS supporting telephony. This preamble can be regarded as an alternative implementation for the MT generic setup procedure for speech. However, it may still send bearer capabilites in the Call Confirm message for either EFR or Half Rate

Test Step Dynamic Behaviour

Test Step Name: CC_EstMsTermCall(rate:IA5String; sub:B_1)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To perform the CC message exchange to establish a mobile station terminating call with non hopping channel (speech or data call). (Similar to EstMsTermFullRateCallNonFH but without initial RRmtcallprepare)
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+CC_Est_MT_Call			
2		+ltree_Asgn			
3		+Adjust_gsmanddcs_powerlvl(0, 15, TCV_AssCmd)			
4		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
5		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
		ltree_Asgn			
6		[rate <> C_Half]			Full rate
7		(TCV_Tchtype := '00001'B)			
8		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, TCV_asscmd_ts, TSPX_TscDef, 9, TCV_tch_arfcn, CellChDes_Omit, TCV_ChMod, StartingTm_omit, CphMod_omit))			
9		[rate = C_Half]			Half rate
10		(TCV_Tchtype := INT_TO_BIT((2 + BIT_TO_INT(sub)), 5))			
11		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, TCV_asscmd_ts, TSPX_TscDef, 9, TCV_tch_arfcn, CellChDes_Omit, TCV_ChMod, StartingTm_omit, CphMod_omit))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: CCModifyTCH(slot :SN; tsc:TSC)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: Send channel mode modify command and wait for successful completion. This is used for CC testing.
Default: OtherEvents
Comments: The calling tree shall prepare two variables for the step: TCV_chtype for the type of the channel, TCV_ChMod with the channel mode IE for the channel.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Null := OM_ChMdModi(TCV_chTch, TCV_ChMod))			
2		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_07(TCV_chtype, TCV_ChMod.mode, slot, tsc, TCV_tch_arfcn))		
3		L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_08(TCV_chtype, TCV_ChMod.mode, slot, tsc, TCV_tch_arfcn))		

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: CCPage
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: Send paging request and wait for channel request. This is usde in CC testing.
Default: OtherEvents
Comments: TCV_Rr contains the request reference and TCV_Fn contains the frame number of the channel request.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
2		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_01)		

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: CCstatuschk_01(ch: LOGICCH; st:CCSTATE)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To check whether the MS under test is in the CC state `st` and cause = #30.
Default: OtherEventsFail
Comments: if the cause value is #30 and the CC state value is `st`, the preliminary verdict is pass. This is used in the case of transcation initiated by test system, the DCCH is `ch`.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_DatRqCcstEnq	CCStESnd(ch, CCStatusEq_01(TI_02))		
2		L?DL_DatInCcst	CCStRcv(ch, CCStatus_14(TI_01, st))		

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: CCstatuschk_02(ch: LOGICCH; st:CCSTATE; ti_orig, ti_dest: TI)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To check whether the MS under test is in the CC state `st` and cause = #30.
Default: OtherEvents
Comments: if the cause value is #30 and the CC state value is `st`, the preliminary verdict is pass.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_DatRqCcstEnq	CCStESnd(ch, CCStatusEq_01(ti_orig))		
2		L?DL_DatInCcst (TCV_Cau := DL_DatInCcst.msg.cau, TCV_CCst := DL_DatInCcst.msg.cst)	CCStRcv(ch, CCStatus_01(ti_dest))		
3		[(TCV_Cau.cau_class = '001'B) AND (TCV_Cau.cau_va = '1110'B)]			
4		[TCV_CCst.csv = INT_TO_BIT(st, 6)]			1.
5	L_0852	[TCV_CCst.csv <> INT_TO_BIT(st, 6)]		(I)	2.
6		+PostLinkRelEnd(ch)			
7	L_0853	[(TCV_Cau.cau_class <> '001'B) OR(TCV_Cau.cau_va<> '1110'B)]		(I)	3.
8		+PostLinkRelEnd(ch)			

Detailed Comments:
1. Now in CC state 'st' and cause = #30.
2. Not in CC state 'st'.
3. Cause is not #30.

Test Step Dynamic Behaviour

Test Step Name: CCstatuschk_03(st:CCSTATE; Ti:TI)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To check whether the MS under test is in the CC state `st` and cause = #30.
Default: OtherEventsFail
Comments: if the cause value is #30 and the CC state value is `st`, the preliminary verdict is pass. This is used in the case of DCCH = SDCCH4 and transcation initiated by the MS. The calling tree shall prepare the variable for the step: TCV_ch.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_TI1 := Ti, TCV_TI1.ti_f := '0'B)			
2		L!DL_DatRqCcstEnq	CCStESnd(TCV_ch, CCStatusEq_01(Ti))		
3	L_0854	L?DL_DatInCcst	CCStRcv(TCV_ch, CCStatus_14(TCV_TI1, st))	(P)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: CCstatuschk_05(st:CCSTATE; ti:TI; ch: LOGICCH)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To check whether the MS under test is in the CC state `st` and cause = #30.
Default: OtherEventsFail
Comments: if the cause value is #30 and the CC state value is `st`, the preliminary verdict is pass.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_TI1.ti_v := ti.ti_v)			
2		[ti.ti_f = '0'B]			
3		(TCV_TI1.ti_f := '1'B)			
4		+localtree			
5		[ti.ti_f = '1'B]			
6		(TCV_TI1.ti_f := '0'B)			
7		+localtree			
		localtree			
8		L!DL_DatRqCcstEnq	CCStESnd(ch, CCStatusEq_01(ti))		
9	L_0855	L?DL_DatInCcst	CCStRcv(ch, CCStatus_14(TCV_TI1, st))	(P)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Check_hoaccPwr(Hoacc:HOACC_PARA; exp_pwr:BITSTRING)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To check that the received handover access bursts' power levels are as expected
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_mindbm := OC_OctToInt(Hoacc.min_mspwr), TCV_maxdbm := OC_OctToInt(Hoacc.max_mspwr))			1.
2		+localtree_PwrTol			
3	L_1051	[(TCV_mindbm >= ((TCV_expdbm-TCV_n) MOD 256)) AND (TCV_maxdbm <= ((TCV_expdbm+TCV_n) MOD 256))]		(P)	2.
4	L_1052	[C_Yes]		(F)	
		localtree_PwrTol			
5		[TSPC_PGSM OR TSPC_EGSM]			3.
6		(TCV_expdbm := 43 -(BIT_TO_INT(exp_pwr) * 2))			
7		[TCV_expdbm <= 5]			
8		(TCV_expdbm := 5, TCV_n := 5)			
9		[(TCV_expdbm > 5) AND (TCV_expdbm <= 11)]			
10		(TCV_n := 5)			
11		[(TCV_expdbm > 11) AND (TCV_expdbm <= 37)]			
12		(TCV_n := 3)			
13		[TCV_expdbm > 37]			
14		(TCV_expdbm := 39, TCV_n := 2)			
15		[TSPC_DCS]			
16		+localtree_DCSPwr			
17		[TCV_expdbm <= 0]			
18		(TCV_expdbm := 0, TCV_n := 5)			
19		[(TCV_expdbm > 0) AND (TCV_expdbm <= 2)]			
20		(TCV_n := 5)			
21		[(TCV_expdbm > 2) AND (TCV_expdbm <= 12)]			
22		(TCV_n := 4)			
23		[(TCV_expdbm > 12) AND (TCV_expdbm <= 34)]			
24		(TCV_n := 3)			
25		[TCV_expdbm > 34]			
26		(TCV_n := 2)			
		localtree_DCSPwr			
27		[BIT_TO_INT(exp_pwr) <= 28]			
28		(TCV_expdbm := 30 -(BIT_TO_INT(exp_pwr) * 2))			
29		[BIT_TO_INT(exp_pwr) > 28]			
30		(TCV_expdbm := (BIT_TO_INT(exp_pwr) * 2) - 30)			

Detailed Comments:
1. Convert powers to integers (TCV_mindbm, TCV_maxdbm).
2. Check power is within tolerance.
3. Convert expected power to dBm (TCV_expdbm) and work out tolerance (TCV_n). This is different for DCS vs GSM (see GSM 05.05 section 4.1.1).

Test Step Dynamic Behaviour

Test Step Name: Check_hoaccTiming(Hoacc:HOACC_PARA; exp_ta:BITSTRING)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To check that the received handover access bursts' timing advance is as expected
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_expta := BIT_TO_INT(exp_ta))			1.
2	L_1050	[(BIT_TO_INT(Hoacc.min_msta) >= ((TCV_expta-2) MOD 256)) AND(BIT_TO_INT(Hoacc.max_msta) <= ((TCV_expta+2) MOD 256))]		(P)	2.
3	L_1053	[C_Yes]		(F)	

Detailed Comments:
1. Convert timing to integer (TCV_expta).
2. Check burst timing is within 2 bits.

Test Step Dynamic Behaviour

Test Step Name: Check_Time(maxtime: INTEGER)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		+FnArith(TCV_Fn, TCV_Fn1)			
2	L_0856	[TCV_Time <= maxtime]		(P)	
3	L_0857	[TCV_Time > maxtime]		(F)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: CheckMTDif(td:OCTETSTRING; y, k:INTEGER)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To check that the mobile time difference is within +/-2 half bits of the expected value
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_temp1 := OC_OctToInt(td)/8, TCV_temp2 := (k*2 + y) MOD 2097152)			
2	L_1048	[(TCV_temp1 >= TCV_temp2 - 2) AND (TCV_temp1 <= TCV_temp2 + 2)]		(P)	td must be within +/- 2 half bits
3	L_1049	[C_Yes]		(F)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: CheckTIsInStateU0(mo: BOOLEAN; ch: LOGICCH)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: Check that all MO or MT CC entities in the MS are in state U0. This is used in CC testing.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[mo]			
2		(TCV_TI1.ti_f := '1'B, TCV_TI2.ti_f := '0'B)			
3		+localtree1			
4		[NOT mo]			
5		(TCV_TI1.ti_f := '0'B, TCV_TI2.ti_f := '1'B)			
6		+localtree1			
		localtree1			
7		(TCV_Cnt := 0)			
8		REPEAT localtree2 UNTIL [TCV_Cnt = 7]			
		localtree2			
9		(TCV_TI1.ti_v := INT_TO_BIT(TCV_Cnt, 3), TCV_TI2.ti_v := INT_TO_BIT(TCV_Cnt, 3))			
10		L!DL_DatRqCcstEnq	CCStESnd(ch, CCStatusEq_01(TCV_TI1))		
11	L_0858	L?DL_DatInRelCmp	RelComRcv(ReleaseCmpRcv(TCV_TI2, Cause_07))	(P)	1.
12		(TCV_Cnt := TCV_Cnt + 1)			

Detailed Comments: 1. Cause #81 (invalid TI value)

Test Step Dynamic Behaviour

Test Step Name: ChkMsrmntRpt(index1, index2:INTEGER)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: Check whether the mesurment report is correct and assign the verdict accordingly.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			
2		(TCV_Res := OC_MsrReptChk(TCV_MsrRes, index1))			
3	L_0859	[NOT TCV_Res]		(F)	
4	L_0860	[TCV_Res]		(P)	
5		[TSPC_DCS]			
6		(TCV_Res := OC_MsrReptChk(TCV_MsrRes, index2))			
7	L_0861	[NOT TCV_Res]		(F)	
8	L_0862	[TCV_Res]		(P)	

Detailed Comments:

Test Step Dynamic Behaviour					
Test Step Name:	Ciphering_off(ch: LOGICCH)				
Group:	GSM_L3_MS_v4220/Miscellaneous/				
Objective:					
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_CphMdChg(TCV_ch, CphMod_02, TCV_CphKey))			
2		L!DL_DatRqCphmCmd	CphCmd(ch, CphModeCmd_02)		
3	L_0863	L?DL_DatInCphmCom	CphCom(CphModeCmp_01)	(P)	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Ciphering_on(ch: LOGICCH)				
Group:	GSM_L3_MS_v4220/Miscellaneous/				
Objective:	Send cipher command and wait for successful completion				
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_CphMdChg(ch, CphMod_01, TCV_CphKey))			
2		L!DL_DatRqCphmCmd	CphCmd(ch, CphModeCmd_01)		
3	L_0864	L?DL_DatInCphmCom	CphCom(CphModeCmp_01)	(P)	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	CMsrvcRq				
Group:	GSM_L3_MS_v4220/Miscellaneous/				
Objective:					
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L?DL_DatInCmsRq	CMSerDatReq(CMServiceReq_01)		1)
2		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
3	L_0865	L?DL_DatInSetup (TCV_Mt := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)	(P)	2)
4	L_0866	L?DL_DatInRegister	Register_03(RegisterPdu_01)	(P)	2)
5	L_0867	L?DL_DatInCpData	DatInCpData(TCV_ch, CpDataPdu_any)	(P)	2)
Detailed Comments:		1) MS shall send a CM Service Request. 2) Any initial CM message is to expect. This test step is called in TC_26_7_4_3_4 & TC_26_7_5_8_2			

Test Step Dynamic Behaviour

Test Step Name: DTMFSignalling(n: INTEGER; ti_ms: TI; ti_ss: TI; ch: LOGICCH)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Cnt:=0)			
2		REPEAT localtree UNTIL [TCV_Cnt=n]			
		localtree			
3		L?DL_DatInStartDtmf(TCV_Char := DL_DatInStartDtmf.msg.kpf.kpf_info)	StartDTMFRcv(StartDtmf_02(ti_ms))		
4		L!DL_DatRqStartDtmfAck	StartDTMFAckSnd(ch, StartDtmfAck_01(ti_ss, TCV_Char))		
5		L?DL_DatInStopDtmf	StopDTMFRcv(StopDtmf_01(ti_ms))		
6		L!DL_DatRqStopDtmfAck	StopDTMFAckSnd(ch, StopDtmfAck_01(ti_ss))		
7		(TCV_Cnt:=TCV_Cnt+1)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: EstablishFacMO(t:INTEGER; comp:Component_T; sdcch:B_2; slot:SN; tsc:TSC; tch:B_1)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		+InitCall(TCV_Service)			
2		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
3		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
4		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, sdcch, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
5		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
6		ACTIVATE(OtherEventsFail)			Restore Normal default
7		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
8		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
9		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
10		+AssCmdGen(TCV_cellid, TCV_ChRate, slot, tsc, tch)			
11		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
12		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
13		L!DL_DatRqConn START T_dly(t), START T_dly1(10000)	ConnSnd(TCV_chTch, Connect_05(TCV_TI, facilityIEtsndiei(comp)))		1.
14		+AOC_CHK_FAC(TCV_TI0)			

Detailed Comments: 1. To send Facility IE of ForwardChargeAdvice - comp.

Test Step Dynamic Behaviour

Test Step Name: FnArith(fn, fn1:FN)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail
Comments: To calculate the difference of two frame numbers 'fn1' and 'fn'. The test step uses TCV_Cnt1, TCV_Cnt2, TCV_K and TCV_Time

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Cnt1 := (51 *((BIT_TO_INT(fn.t3) - BIT_TO_INT(fn.t2) + 26) MOD 26)) + BIT_TO_INT(fn.t3) +(1326 * BIT_TO_INT(fn.t1_)), TCV_Cnt2 := (51 *((BIT_TO_INT(fn1.t3) - BIT_TO_INT(fn1.t2) + 26) MOD 26)) + BIT_TO_INT(fn1.t3) +(1326 * BIT_TO_INT(fn1.t1_)), TCV_K := ((TCV_Cnt2-TCV_Cnt1+42432) MOD 42432), TCV_Time := TCV_K * 120 / 26)			GSM 04.08 section 10.5.2.30. TCV_K stores the frame number difference. TCV_Time stores the time difference in ms.

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: IdentityRequest(par_int:INTEGER; par_mi:MI)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		+ltree_idrequest			
2	L_0868	L?DL_DatInIdRes	IDRes(IDResponse_30(par_mi))	(P)	
		ltree_idrequest			
3		[par_int=C_IMSI]			
4		L!DL_DatRqIdRq	IDReq(TCV_ch, IDRequest_01('0001'B))		IMSI.
5		[par_int=C_TMSI]			
6		L!DL_DatRqIdRq	IDReq(TCV_ch, IDRequest_01('0100'B))		TMSI.
7		[par_int=C_IMEI]			
8		L!DL_DatRqIdRq	IDReq(TCV_ch, IDRequest_01('0010'B))		IMEI.
9		[par_int=C_IMEISV]			
10		L!DL_DatRqIdRq	IDReq(TCV_ch, IDRequest_01('0011'B))		IMSESV.

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: ImsiAttach(par_mi:MI; ta:TA; mcc, mnc:OCTETSTRING; sim_rmvd:BOOLEAN)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv) AND(TSPC_DetachOnSIMRmv)]			SIM has been removed during the test.
2		(TCV_Null := OO_SIMIns())			
3		+ltree_imsiattach			
4		[TSPC_SwitchOnOff]			
5		(TCV_Null := OO_SwitchOn())			
6		+ltree_imsiattach			
7		[TSPC_DetachOnPwrDn]			
8		(TCV_Null := OO_PowerUp())			Restore power source
9		+ltree_imsiattach			
10		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv) OR(NOT TSPC_DetachOnSIMRmv)) AND(NOT TSPC_SwitchOnOff) AND(NOT TSPC_DetachOnPwrDn))]			
11		(TCV_Null := OO_PowerUp())			Restore power source
		ltree_imsiattach			
12		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_09)		
13		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
14		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
15	L_0869	L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_31(par_mi, mcc, mnc, TCV_lac, C_imsi_attach, TCV_cksn))	(P)	
16		ACTIVATE(OtherEventsFail)			Restore Normal default
17		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp(mcc, mnc, TCV_lac))		No MI
18		+ChanRel(TCV_ch)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: ImsiAttachIni(par_mi:MI; ta:TA; mcc, mnc:OCTETSTRING; sim_rmvd:BOOLEAN)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv) AND(TSPC_DetachOnSIMRmv)]			SIM has been removed during the test.
2		(TCV_Null := OO_SIMIns())			
3		+ltree_imsiattachIni			
4		[TSPC_SwitchOnOff]			
5		(TCV_Null := OO_SwitchOn())			
6		+ltree_imsiattachIni			
7		[TSPC_DetachOnPwrDn]			
8		(TCV_Null := OO_PowerUp())			Restore power source
9		+ltree_imsiattachIni			
10		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv) OR(NOT TSPC_DetachOnSIMRmv)) AND(NOT TSPC_SwitchOnOff) AND(NOT TSPC_DetachOnPwrDn))]			
11		(TCV_Null := OO_PowerUp())			Restore power source
		ltree_imsiattachIni			
12		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) START T_dly(15000)	ChReq(ChRequest_09)		
13		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
14		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
15	L_0870	L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_31(par_mi, mcc, mnc, TCV_lac, C_imsi_attach, TCV_cksn))	(P)	
16		ACTIVATE(OtherEventsFail)			Restore Normal default

Detailed Comments: The test step is calld in TC_26_7_5_8_1, TC_26_7_5_8_2, TC_26_7_5_8_3.

Test Step Dynamic Behaviour

Test Step Name: ImsiAttachNoReaction(par_int: INTEGER; sim_rmvd:BOOLEAN)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv) AND(TSPC_DetachOnSIMRmv)]			SIM has been removed during the test.
2		(TCV_Null := OO_SIMIns())			
3		+NoReaction(par_int)			
4		[TSPC_SwitchOnOff]			
5		(TCV_Null := OO_SwitchOn())			
6		+NoReaction(par_int)			
7		[TSPC_DetachOnPwrDn]			
8		(TCV_Null := OO_PowerUp())			Restore power source
9		+NoReaction(par_int)			
10		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv) OR((TSPC_SIMRmv) AND(NOT TSPC_DetachOnSIMRmv))) AND(NOT TSPC_SwitchOnOff) AND(NOT TSPC_DetachOnPwrDn))]			
11		(TCV_Null := OO_PowerUp())			Restore power source

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: ImsiDetach(par_mi:MI; ta:TA; sim_rmvd:BOOLEAN)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv) AND(TSPC_DetachOnSIMRmv)]			SIM needs to be removed.
2		(TCV_Null :=OO_SIMRmv())			
3		+ltree_imsidetach			
4		[TSPC_SwitchOnOff]			
5		(TCV_Null := OO_SwitchOff())			
6		+ltree_imsidetach			
7		[TSPC_DetachOnPwrDn]			
8		(TCV_Null := OO_PowerDown())			Remove power source
9		+ltree_imsidetach			
10		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv) OR(NOT TSPC_DetachOnSIMRmv)) AND(NOT TSPC_SwitchOnOff) AND(NOT TSPC_DetachOnPwrDn))]			
11		(TCV_Null := OO_PowerDown())			Remove power source
		ltree_imsidetach			
12		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
13		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
14		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
15	L_0871	L?DL_EstInImsidIn	ImsiDet(ImsiDetach_30(par_mi))	(P)	
16		ACTIVATE(OtherEventsFail)			Restore normal default
17		+ChanRel(TCV_ch)			

Detailed Comments: If IMSI detach procedure has been executed, the channel will be relesed. If ATT=0 no detach has been executed, the channel is still connected.

Test Step Dynamic Behaviour

Test Step Name: ImsiDetachNoReaction(par_int: INTEGER; sim_rmvd:BOOLEAN)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv) AND(TSPC_DetachOnSIMRmv)]			SIM needs to be removed.
2		(TCV_Null :=OO_SIMRmv())			
3		+NoReaction(par_int)			
4		[TSPC_SwitchOnOff]			
5		(TCV_Null := OO_SwitchOff())			
6		+NoReaction(par_int)			
7		[TSPC_DetachOnPwrDn]			
8		(TCV_Null := OO_PowerDown())			Remove power source
9		+NoReaction(par_int)			
10		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv) OR((TSPC_SIMRmv) AND(NOT TSPC_DetachOnSIMRmv))) AND(NOT TSPC_SwitchOnOff) AND(NOT TSPC_DetachOnPwrDn))]			
11		(TCV_Null := OO_PowerDown())			Remove power source

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: InCallModi1(srv:SERVICES)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: MMI action to initiate in-call modification to the basic service specified.
Default: OtherEvents
Comments: TCV_Setup_mo1 has been initialised by the calling tree.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Res := OO_InCallModi(srv))			1.
2		(TCV_B := TCV_Setup_mo1.bcap2, TCV_Bcap2 := Bcap_NoIEI(TCV_B.iel,TCV_B.extb3,TCV_B.rchr,TCV_B.cs,TCV_B.tm,TCV_B.itc,TCV_B.extb4,TCV_B.spb,TCV_B.strc,TCV_B.dplxm,TCV_B.config,TCV_B.nirr,TCV_B.est,TCV_B.extb5,TCV_B.accid,TCV_B.ra,TCV_B.sacp,TCV_B.extb6,TCV_B.l1id,TCV_B.uil1,TCV_B.sb,TCV_B.extb6a,TCV_B.nsb,TCV_B.nb,TCV_B.ndb,TCV_B.ur,TCV_B.extb6b,TCV_B.ir,TCV_B.nictx,TCV_B.nicrx,TCV_B.pi,TCV_B.extb6c,TCV_B.ce,TCV_B.modemt))			
3		(TCV_B := TCV_Setup_mo1.bcap1, TCV_Bcap1 := Bcap_NoIEI(TCV_B.iel,TCV_B.extb3,TCV_B.rchr,TCV_B.cs,TCV_B.tm,TCV_B.itc,TCV_B.extb4,TCV_B.spb,TCV_B.strc,TCV_B.dplxm,TCV_B.config,TCV_B.nirr,TCV_B.est,TCV_B.extb5,TCV_B.accid,TCV_B.ra,TCV_B.sacp,TCV_B.extb6,TCV_B.l1id,TCV_B.uil1,TCV_B.sb,TCV_B.extb6a,TCV_B.nsb,TCV_B.nb,TCV_B.ndb,TCV_B.ur,TCV_B.extb6b,TCV_B.ir,TCV_B.nictx,TCV_B.nicrx,TCV_B.pi,TCV_B.extb6c,TCV_B.ce,TCV_B.modemt))			

Detailed Comments: 1. MMI action to initiate in-call modification.

Test Step Dynamic Behaviour					
Test Step Name:	MM_LUP(newmi:MI; mcc, newmnc, newlac:OCTETSTRING; locup:B_2; ta:TA)				
Group:	GSM_L3_MS_v4220/Miscellaneous/				
Objective:					
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+MM_LupInit5(locup, ta)			
2		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(newmi, mcc, newmnc, newlac))		
3		L?DL_DatInTmsireCom	TmsiReallocCmp(TCV_ch)		
4		+ChanRel(TCV_ch)			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	MM_LUP2(newmi:MI; lup_mi:MI; mcc, mnc, oldlac, newlac:OCTETSTRING; locup:B_2; cksn:BITSTRING; ta:TA)				
Group:	GSM_L3_MS_v4220/Miscellaneous/				
Objective:					
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+MM_LupInit(lup_mi, mcc, mnc, oldlac, locup, cksn, ta)			
2		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(newmi, mcc, mnc, newlac))		
3	L_0873	L?DL_DatInTmsireCom	TmsiReallocCmp(TCV_ch)	(P)	
4		+ChanRel(TCV_ch)			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	MM_LUP3(mcc, mnc, lac: OCTETSTRING; locup:B_2; ta:TA)				
Group:	GSM_L3_MS_v4220/Miscellaneous/				
Objective:					
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+MM_LupInit2(locup, ta)			
2	L_0874	L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp(mcc, mnc, lac))	(P)	without MI
3		+ChanRel(TCV_ch)			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: MM_LupAndStop(par_mi:MI; mcc, mnc, lac:OCTETSTRING; locup:B_2; cksn:BITSTRING; ta:TA; ch_sacch:LOGICCH; time:INTEGER)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective:

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		+MM_LupInit(par_mi, mcc, mnc, lac, locup, cksn, ta)			
2		+Stopmaindcch(TCV_ch, ch_sacch)			
3		[time <> 0]			
4		+NoReaction(time)			
5		L!MDL_RelRq	MDLRelReq(TCV_ch)		
6		(TCV_Null := OM_Reactivate(TCV_ch, ch_sacch))			
7		[time = 0]			
8		L!MDL_RelRq	MDLRelReq(TCV_ch)		
9		(TCV_Null := OM_Reactivate(TCV_ch, ch_sacch))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LupAuthRpt(par_mi:MI; mcc, mnc, lac:OCTETSTRING; locup:B_2; cksn:B_3; rand:RAND; ta:TA)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective: To initialise Location Update and repeated Authentication procedures till T3210 times out.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		+MM_LupInit(par_mi, mcc, mnc, lac, locup, cksn, ta)			
2		START T_dly(C_T_T3210min)			
3		(TCV_Cntstart := FALSE)			
4		REPEAT ltree_auth UNTIL [TCV_Cntstart]			
		ltree_auth			
5		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNDef, rand))		
6		?TIMEOUT T_dly			
7		L?DL_DatInAuthRes	AuthRes(AuthResponse)		
8		+ltree_rcvRel			2.
9		+ltree_rcvRel			2.
10		L?DL_DatInAuthRes	AuthRes(AuthResponse)		
		ltree_rcvRel			
11		L?DL_RelIn	DLRelInd_01		2.
12		(TCV_Cntstart := TRUE)			Exit

Detailed Comments:
1. The test step is used in TC_26_7_4_3_2 and TC_26_7_4_3_3.
2. The T3210 has expired. The MS aborts the RR connection on the main signalling link.

Test Step Dynamic Behaviour					
Test Step Name:	MM_LUPauth1(newtmsi: MI; mcc, mnc, lac:OCTETSTRING; locup:B_2; cksn: B_3; rand:RAND; ta:TA)				
Group:	GSM_L3_MS_v4220/Miscellaneous/				
Objective:					
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+MM_LupInit2(locup, ta)			
2		+Authentication(TCV_ch, cksn, rand)			
3		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(newtmsi, mcc, mnc, lac))		
4	L_0877	L?DL_DatInTmsireCom	TmsiReallocCmp(TCV_ch)	(P)	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	MM_LUPauth2(newtmsi: MI; lup_mi:MI; mcc, mnc, old_lac, new_lac:OCTETSTRING; locup:B_2; old_cksn, new_cksn: B_3; rand:RAND; ta:TA)				
Group:	GSM_L3_MS_v4220/Miscellaneous/				
Objective:					
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+MM_LupInit3(lup_mi, mcc, mnc, old_lac, locup, old_cksn, ta)			1.
2		+Authentication(TCV_ch, new_cksn, rand)			
3		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(newtmsi, mcc, mnc, new_lac))		
4	L_0878	L?DL_DatInTmsireCom	TmsiReallocCmp(TCV_ch)	(P)	
Detailed Comments:	1. The test step MM_LupInit3 cancels the timer T_dly when receiving CHANNEL REQUEST from the MS for location updating. Caution for its use.				

Test Step Dynamic Behaviour

Test Step Name: MM_LUP_imsi(duration: INTEGER; newmi, lup_mi:MI; mcc, mnc, lac, newlac:OCTETSTRING; locup:B_2; ta:TA)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective:

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Time := (duration +((20 * duration) /100)))			
2		START T_dly(TCV_Time)			
3		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) READTIMER T_dly(TCV_Time), CANCEL T_dly	ChReq(ChRequest_09)		
4		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
5		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
6		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_31(lup_mi, mcc, mnc, lac, locup, TCV_cksn))		
7		ACTIVATE(OtherEventsFail)			Restore Normal default
8		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(newmi, mcc, mnc, newlac))		
9		+ChanRel(TCV_ch)			
10	L_0879	[TCV_Time <= duration]		(P)	
11	L_0880	[TCV_Time > duration]		(F)	
12	L_0881	?TIMEOUT T_dly		(F)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LupInit(par_mi:MI; mcc, mnc, lac:OCTETSTRING; locup:B_2; cksn:BITSTRING; ta:TA)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective:

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_09)		
2		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
3		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
4	L_0882	L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_31(par_mi, mcc, mnc, lac, locup, cksn))	(P)	
5		ACTIVATE(OtherEventsFail)			Restore Normal default

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LupInit2(locup:B_2; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_09)		
2		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
3		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
4		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(locup))		
5		ACTIVATE(OtherEventsFail)			Restore Normal default

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LupInit3(par_mi:MI; mcc, mnc, lac:OCTETSTRING; locup:B_2; cksn:BITSTRING; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_09)		
2		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
3		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
4	L_0883	L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_31(par_mi, mcc, mnc, lac, locup, cksn))	(P)	
5		ACTIVATE(OtherEventsFail)			Restore Normal default

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LupInit4(locup:B_2; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_09)		
2		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
3		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
4		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(locup))		
5		ACTIVATE(OtherEventsFail)			Restore Normal default

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LupInit5(locup:B_2; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
2		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
3		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
4	L_0884	L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(locup))	(P)	
5		ACTIVATE(OtherEventsFail)			Restore Normal default

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LupInit6(locup:B_2; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_02)		
2		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
3		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
4	L_0885	L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(locup))	(P)	
5		ACTIVATE(OtherEventsFail)			Restore Normal default

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LUP_tmsirealloc(newtmsi:MI; expectedlup_mi:MI; mcc, mnc, lup_lac, lac: OCTETSTRING; locup:B_2; lup_cksn: BITSTRING; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		+MM_LupInit(expectedlup_mi, mcc, mnc, lup_lac, locup, lup_cksn, ta)			
2		+TmsiReallocation(newtmsi, mcc, mnc, lac)			
3		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiMsi_omit, mcc, mnc, lac))		

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LUPper(mcc, mnc, lac: OCTETSTRING; locup:B_2; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_dly(C_T_T3212min)			
2	L_0886	?TIMEOUT T_dly		(P)	
3		START T_dly(C_T_T3212dif*2)			
4		+MM_LupInit4(locup, ta)			
5	L_0887	L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp(mcc, mnc, lac))	(P)	no MI
6	L_0888	?TIMEOUT T_dly		(F)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LUPper2(par_timetol:INTEGER; locup:B_2; mcc, mnc, lac:OCTETSTRING; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_dly(par_timetol)			
2		+MM_LupInit4(locup, ta)			
3	L_0889	L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp(mcc, mnc, lac))	(P)	no MI, fop;
4	L_0890	?TIMEOUT T_dly		(F)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LUPperauth(oldmi, newmi: MI; mcc, mnc, oldlac, newlac:OCTETSTRING; locup:B_2; cksn: B_3; rand:RAND; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_dly(C_T_T3212min)			T3212 - 15 s
2	L_0891	?TIMEOUT T_dly		(P)	
3		START T_dly(C_T_T3212dif * 2)			2 * 15 s tolerance
4		+MM_LupInit3(oldmi, mcc, mnc, oldlac, locup, cksn, ta)			
5		+Authentication(TCV_ch, TCV_cksn, rand)			
6		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(newmi, mcc, mnc, newlac))		
7		L?DL_DatInTmsireCom	TmsiReallocCmp(TCV_ch)		
8	L_0892	?TIMEOUT T_dly		(F)	T3212 + 15 s

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LUPperrej(par_rej:REJCAU; locup:B_2; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_dly(C_T_T3212min)			
2	L_0893	?TIMEOUT T_dly		(P)	
3		START T_dly(C_T_T3212dif*2)			
4		+MM_LupInit4(locup, ta)			
5	L_0894	L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(par_rej))	(P)	
6		+ChanRel(TCV_ch)			
7	L_0895	?TIMEOUT T_dly		(F)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LUPperrej2(par_rej:REJCAU; par_mi:MI; par_toleranz: INTEGER; mcc, mnc, lac: OCTETSTRING; locup:B_2; cksn: BITSTRING; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_dly(par_toleranz)			
2		+MM_LupInit3(par_mi, mcc, mnc, lac, locup, cksn, ta)			
3	L_0896	L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(par_rej))	(P)	
4		+ChanRel(TCV_ch)			
5	L_0897	?TIMEOUT T_dly		(F)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LUPperrej3(par_mi:MI;par_toleranz:INTEGER; mcc, mnc, lac:OCTETSTRING; locup:B_2; cksn:BITSTRING; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_dly(par_toleranz)			
2		+MM_LupInit3(par_mi, mcc, mnc, lac, locup, cksn, ta)			
3		+ChanRel(TCV_ch)			
4	L_0898	?TIMEOUT T_dly		(F)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LupRej(par_rej: REJCAU; locup:B_2; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		+MM_LupInit2(locup, ta)			
2	L_0899	L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(par_rej))	(P)	
3		+ChanRel(TCV_ch)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LupRej2(par_rej: REJCAU; par_mi:MI; mcc, mnc, lac:OCTETSTRING; locup:B_2; cksn:BITSTRING; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		+MM_LupInit(par_mi, mcc, mnc, lac, locup, cksn, ta)			
2	L_0900	L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(par_rej))	(P)	
3		+ChanRel(TCV_ch)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_PwrOrSimOff(sim_rmvd:BOOLEAN)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv)]			SIM needs to be removed.
2		(TCV_Null :=OO_SIMRmv())			
3		[TSPC_SwitchOnOff]			
4		(TCV_Null := OO_SwitchOff())			
5		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv)) AND(NOT TSPC_SwitchOnOff))]			
6		(TCV_Null := OO_PowerDown())			Remove power source

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_PwrOrSimOn(sim_rmvd:BOOLEAN)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv)]			SIM has been removed
2		(TCV_Null := OO_SIMIns())			
3		[TSPC_SwitchOnOff]			
4		(TCV_Null := OO_SwitchOn())			
5		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv)) AND(NOT TSPC_SwitchOnOff))]			
6		(TCV_Null := OO_PowerUp())			Restore power source

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_check_ecall1(ta:TA; mi:MI)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_Serv_TS11]			
2		+BasicServiceMO(C_EmgCall, TSPX_EmgCallRate)			
3		+InitCall(TCV_Service)			
4		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_16)		
5		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
6		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
7		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_05(mi))		
8		ACTIVATE(OtherEventsFail)			Restore Normal default
9		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
10		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
11		+ltree_check_ti_flag			
12	L_0901	L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_04(TCV_TI))	(P)	
13		+ChanRel(TCV_ch)			
14		[NOT TSPC_Serv_TS11]			
		ltree_check_ti_flag			
15	L_0902	[TCV_TI0.ti_f = '1'B]		(F)	
16		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
17		L?DL_RelIn	DLRelInd_01		
18	L_0903	[TCV_TI0.ti_f = '0'B]		(P)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_check_ecall2(parexpected_mi: MI; parexpected_cksn: BITSTRING; ta:TA)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_Serv_TS11]			
2		+BasicServiceMO(C_EmgCall, TSPX_EmgCallRate)			
3		+InitCall(TCV_Service)			
4		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_16)		
5		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
6		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
7		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_31(parexpected_mi, parexpected_cksn))		
8		ACTIVATE(OtherEventsFail)			Restore Normal default
9		L!DL_DatRqCmsAcp	CMSerAcp(TCV_ch, CMServiceAcp_01)		
10		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
11		+ltree_check_ti_flag			
12	L_0904	L!DL_DatRqRelCmp	RelComSnd(TCV_ch, ReleaseCmp_04(TCV_TI))	(P)	
13		+ChanRel(TCV_ch)			
14	L_0905	[NOT TSPC_Serv_TS11]		(P)	
		ltree_check_ti_flag			
15	L_0906	[TCV_TI0.ti_f = '1'B]		(F)	
16		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
17		L?DL_RelIn	DLRelInd_01		
18	L_0907	[TCV_TI0.ti_f = '0'B]		(P)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_no_cmservices(par_int:INTEGER)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		+InitCall(TCV_Service)			
2		START T_dly(par_int)			
3	L_0908	?TIMEOUT T_dly		(P)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_no_paging(par_mi: MI; par_checktime:INTEGER; ccd: CCD; activ_cell:CellID)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		+CCCH_group_Paging_group(ccd, TSPX_IMSI)			
2		+SelectPagingCh(activ_cell)			
3		L!DL_UdatRqPg1Rq START T_dly(par_checktime)	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_30(par_mi))		
4	L_0909	?TIMEOUT T_dly		(P)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: NoReaction(par_int:INTEGER)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_dly(par_int)			
2	L_0910	?TIMEOUT T_dly		(P)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: RR_hocomp1(time_fn_fn1:INTEGER; oldch:LOGICCH)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To finish the HO-procedure. Timing advance = 20 bits period
Default: RcvHdOvAcc, OtherEvents
Comments: used var's: TCV_ch, TCV_Horf, TCV_Cntref

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RacInHoacc (TCV_FnBurst := DL_RacInHoacc.fn)	HndOvAccRcv(TCV_ch, HandOverAcc_02(TCV_Horf))		1.
2		L?DL_EstIn	DLEstInd(TCV_ch)		
3		(TCV_HoaccPara := OM_GetHoaccPara(TCV_ch))			
4	L_1043	[(OC_OctToInt(TCV_HoaccPara.num_bursts) >= TCV_Cntref) AND(OC_OctToInt(TCV_HoaccPara.num_bursts) <= (TCV_Cntref+20))]		(P)	2.
5		+localtree_continue			
6	L_1044	[C_Yes]		(F)	
7		+localtree_continue			
		localtree_continue			
8		(TCV_Fn1:= OM_ReturnFn(TCV_ch))			
9		L?DL_DatInHoCom(TCV_Fn := DL_DatInHoCom.fn)	HndOvCmpRcv(TCV_ch, HandOverCmp_20)		
10		L!MDL_RelRq	MDLRelReq(oldch)		Release the original channel.
11		+FnArith(TCV_Fn1, TCV_Fn)			3.
12	L_0911	[TCV_Time > time_fn_fn1]		(F)	
13	L_0912	[TCV_Time <= time_fn_fn1]		(P)	

Detailed Comments:

1. Match the 1st access burst. The access bursts except the 1st one are matched by the default test step RcvHdOvAcc.
2. Check the number of access bursts received with tolerance +20 (10 for FACCH interleave & 10 for MS processing).
3. Check HO-time.

Test Step Dynamic Behaviour

Test Step Name: RR_hocomp3(time_fn1_fn:INTEGER)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To finish the HO-procedure.
Default: OtherEvents
Comments: used var's:
TCV_ch, TCV_Horf, TCV_L1Head

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Cnt:=0, TCV_L1Head := OM_GetL1Hd(TCV_sacch))			1.
2		REPEAT localtree_hoacc UNTIL [TCV_Cnt = 4]			
3		L?DL_EstIn	DLEstInd(TCV_ch)		
4		L?DL_DatInHoCom(TCV_Fn := DL_DatInHoCom.fn)	HndOvCmpRcv(TCV_ch, HandOverCmp_20)		
5		+localtree_hotime			2.
		localtree_hoacc			
6		L?DL_RacInHoacc	HndOvAccRcv(TCV_ch, HandOverAcc_02(TCV_Horf))		
7		(TCV_Cnt := TCV_Cnt + 1)			
		localtree_hotime			
8		+FnArith(TCV_Fn1, TCV_Fn)			
9	L_0913	[TCV_Time > time_fn1_fn]		(F)	
10	L_0914	[TCV_Time <= time_fn1_fn]		(P)	

Detailed Comments: 1) Store first SACCH L1 header from MS to check TA and Power
2) Check of HO-time

Test Step Dynamic Behaviour

Test Step Name: RRmtcallprepare(ta:TA; rand:RAND)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To prepare a mobile terminating call establishment.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEstRRConn(TCV_slot, TCV_tsc, ta)			
2		+Authentication(TCV_ch, TCV_cksn, rand)			
3		+Ciphering_on(TCV_ch)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: RcvSetupOrEsetup(Setup: SETUP_MO_PDU; Esetup:ESETUP_PDU; Ecall:BOOLEAN)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To manage Setup Mobile Originated.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[NOT Ecall]			
2		L?DL_DatInSetup (TCV_Setup_mo1 := DL_DatInSetup.msg, TCV_TI := TCV_Setup_mo1.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B)	SetupRcv(Setup)		
3		+CallProcGen(TCV_Setup_mo1, TCV_Service)			
4		[Ecall]			
5		L?DL_DatInESetup (TCV_Esetup1 := DL_DatInESetup.msg, TCV_Setup_mo1.ti := TCV_Esetup1.ti, TCV_Setup_mo1.bcap1 := TCV_Esetup1.bcap, TCV_Setup_mo1.ccpd := TCV_Esetup1.ccpd, TCV_Setup_mo1.mt := TCV_Esetup1.mt, TCV_TI := TCV_Setup_mo1.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B)	ESetupRcv(Esetup)		
6		+CallProcGen(TCV_Setup_mo1, TCV_Service)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SendSeqNo_chk
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To check whether the sending sequence number is the same
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1	L_0915	[TCV_Mt1 = TCV_Mt]		(P)	same N(SD)
2	L_0916	[TCV_Mt1 <> TCV_Mt]		(F)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SelectPagingCh(cell:CellID)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective: To assign a paging channel and an access grant channel to the variable TCV_PgCh and TCV_agch respectively, depending on the parameter 'cell' (cell ID)

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[cell = C_CellA]			
2		[TCV_Ccchg=0]			
3		(TCV_PgCh := C_PCH_A_1, TCV_agch := C_AGCH_A_1, TCV_RaCh:= C_RACH_A_1)			
4		[TCV_Ccchg=1]			
5		(TCV_PgCh := C_PCH_A_2, TCV_agch := C_AGCH_A_2, TCV_RaCh := C_RACH_A_2)			
6		[TCV_Ccchg=2]			
7		(TCV_PgCh := C_PCH_A_3, TCV_agch := C_AGCH_A_3, TCV_RaCh := C_RACH_A_3)			
8		[TCV_Ccchg=3]			
9		(TCV_PgCh := C_PCH_A_4, TCV_agch := C_AGCH_A_4, TCV_RaCh := C_RACH_A_4)			
10		[cell = C_CellB]			
11		[TCV_Ccchg=0]			
12		(TCV_PgCh := C_PCH_B_1, TCV_agch := C_AGCH_B_1, TCV_RaCh := C_RACH_B_1)			
13		[TCV_Ccchg=1]			
14		(TCV_PgCh := C_PCH_B_2, TCV_agch := C_AGCH_B_2, TCV_RaCh := C_RACH_B_2)			
15		[TCV_Ccchg=2]			
16		(TCV_PgCh := C_PCH_B_3, TCV_agch := C_AGCH_B_3, TCV_RaCh := C_RACH_B_3)			
17		[TCV_Ccchg=3]			
18		(TCV_PgCh := C_PCH_B_4, TCV_agch := C_AGCH_B_4, TCV_RaCh := C_RACH_B_4)			
19		[cell = C_CellC]			
20		[TCV_Ccchg=0]			
21		(TCV_PgCh := C_PCH_C_1, TCV_agch := C_AGCH_C_1, TCV_RaCh := C_RACH_C_1)			
22		[TCV_Ccchg=1]			
23		(TCV_PgCh := C_PCH_C_2, TCV_agch := C_AGCH_C_2, TCV_RaCh := C_RACH_C_2)			
24		[TCV_Ccchg=2]			
25		(TCV_PgCh := C_PCH_C_3, TCV_agch := C_AGCH_C_3, TCV_RaCh := C_RACH_C_3)			
26		[TCV_Ccchg=3]			
27		(TCV_PgCh := C_PCH_C_4, TCV_agch := C_AGCH_C_4, TCV_RaCh := C_RACH_C_4)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Set_CellChDescr(cell:CellID)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To set the cell channel description
Default:
Comments: used in TC_26_6_13

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			
2		[cell = C_CellA]			
3		(TCV_cchd1 := CellChDes_20_A)			
4		[cell = C_CellB]			
5		(TCV_cchd1 := CellChDes_20_B)			
6		[TSPC_DCS]			
7		[cell = C_CellA]			
8		(TCV_cchd1 := CellChDes_203_Adiei)			
9		[cell = C_CellB]			
10		(TCV_cchd1 := CellChDes_201_Bdiei)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SetupRcvMo2(pdu_setup: SETUP_MO_PDU)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To manage Setup Mobile Originated.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_DatInSetup (TCV_Setup_mo := DL_DatInSetup.msg)	SetupRcv(pdu_setup)		
2		+CallProcGen(TCV_Setup_mo, TCV_Service)			
3		(TCV_TI1 := TCV_Setup_mo.ti, TCV_TI2 := TCV_TI1, TCV_TI2.ti_f := '1'B, TCV_CallProc.ti := TCV_TI2)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Switchcell(old_cell, activ_cell:CellID; lac:OCTETSTRING; subch, cksn, rand:BITSTRING; arfcngsm, arfcndcs:INTEGER; ccd: CCD; imsi:HEXSTRING)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To force the MS to reselect cell A
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+Varinit_fix(activ_cell, lac, subch, cksn, rand, arfcngsm, arfcndcs, ccd, imsi)			
2		(TCV_Null := OM_ChangeRFOf2Cells(activ_cell, C_E_default, old_cell, C_E_notsuitable))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SvcSupportedChk(svc:IA5String)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To check wether the Basic Service svc is supported against PICS answers.
Default: OtherEvents
Comments: In PICS, if the svc is declared supported, the test case variable TCV_supported is set to TRUE, otherwisw FALSE.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_supported := C_No)			
2		[(svc = C_Telephony) AND TSPC_Serv_TS11]			TS11
3		(TCV_supported := C_Yes)			
4		[(svc = C_EmgCall) AND TSPC_Serv_TS12]			TS12
5		(TCV_supported := C_Yes)			
6		[(svc = C_AltSpchFax) AND TSPC_Serv_TS61]			TS61
7		(TCV_supported := C_Yes)			
8		[(svc = C_AutoFax) AND TSPC_Serv_TS62]			TS62
9		(TCV_supported := C_Yes)			
10		[(svc = C_Async300) AND TSPC_Serv_BS21]			BS21
11		(TCV_supported := C_Yes)			
12		[(svc = C_Async1200) AND TSPC_Serv_BS22]			BS22
13		(TCV_supported := C_Yes)			
14		[(svc = C_Async120075) AND TSPC_Serv_BS23]			BS23
15		(TCV_supported := C_Yes)			
16		[(svc = C_Async2400) AND TSPC_Serv_BS24]			BS24
17		(TCV_supported := C_Yes)			
18		[(svc = C_Async4800) AND TSPC_Serv_BS25]			BS25
19		(TCV_supported := C_Yes)			
20		[(svc = C_Async9600) AND TSPC_Serv_BS26]			BS26
21		(TCV_supported := C_Yes)			
22		[(svc = C_Sync1200) AND TSPC_Serv_BS31]			BS31
23		(TCV_supported := C_Yes)			
24		[(svc = C_Sync2400) AND TSPC_Serv_BS32]			BS32
25		(TCV_supported := C_Yes)			
26		[(svc = C_Sync4800) AND TSPC_Serv_BS33]			BS33
27		(TCV_supported := C_Yes)			
28		[(svc = C_Sync9600) AND TSPC_Serv_BS34]			BS34
29		(TCV_supported := C_Yes)			
30		[(svc = C_PAD300) AND TSPC_Serv_BS41]			BS41
31		(TCV_supported := C_Yes)			
32		[(svc = C_PAD1200) AND TSPC_Serv_BS42]			BS42
33		(TCV_supported := C_Yes)			
34		[(svc = C_PAD120075) AND TSPC_Serv_BS43]			BS43
35		(TCV_supported := C_Yes)			
36		[(svc = C_PAD2400) AND TSPC_Serv_BS44]			BS44
37		(TCV_supported := C_Yes)			
38		[(svc = C_PAD4800) AND TSPC_Serv_BS45]			BS45
39		(TCV_supported := C_Yes)			
40		[(svc = C_PAD9600) AND TSPC_Serv_BS46]			BS46
41		(TCV_supported := C_Yes)			
42		[(svc = C_Packet2400) AND TSPC_Serv_BS51]			BS51
43		(TCV_supported := C_Yes)			
44		[(svc = C_Packet4800) AND TSPC_Serv_BS52]			BS52
45		(TCV_supported := C_Yes)			
46		[(svc = C_Packet9600) AND TSPC_Serv_BS53]			BS53
47		(TCV_supported := C_Yes)			
48		[(svc = C_AltSpchData) AND TSPC_Serv_BS61]			BS61
49		(TCV_supported := C_Yes)			
50		[(svc = C_SpchData) AND TSPC_Serv_BS81]			BS81
51		(TCV_supported := C_Yes)			
52		[C_Yes]			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Timadv_Pwrlvl_chk(k, y:INTEGER; pwrlvl:B_5)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To check whether the power level and the timing advance in L1 head is correct.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_temp1 := BIT_TO_INT(TCV_L1Head.ta), TCV_temp2 := (k*2 + y) MOD 256)			
2	L_1045	[(TCV_temp2 >= 60) OR (TCV_temp2 = 0) OR(TCV_temp1 > 63)]		(I)	
3	L_0917	[(TCV_temp2 >= TCV_temp1 - 2) AND (TCV_temp2 <= TCV_temp1 + 2) AND (TCV_L1Head.mspwrlvl = pwrlvl)]		(P)	TA must be within +/- 2 bits
4	L_0918	[C_Yes]		(F)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: TmsiReallocation(par_mi:MI; mcc, mnc, lac: OCTETSTRING)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_DatRqTmsireCmd	TmsiReallocSnd(par_mi, TCV_ch, mcc, mnc, lac)		
2	L_0919	L?DL_DatInTmsireCom	TmsiReallocCmp(TCV_ch)	(P)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Varinit_tch(arfcngsm, arfcndcs:INTEGER)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To assign an initial arfcn value to TCV_tch_arfcn
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			
2		(TCV_tch_arfcn:= arfcngsm)			
3		[TSPC_DCS]			
4		(TCV_tch_arfcn:= arfcndcs)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Varinit_fix(activ_cell:CellID; lac:OCTETSTRING; subch:B_2; cksn:B_3; rand:RAND; arfcngsm, arfcndcs:INTEGER; ccd: CCD; imsi:HEXSTRING)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_cellid := activ_cell, TCV_ch := OC_SubchOfSdcch4(subch, activ_cell), TCV_sacch := OC_SubchOfSacch4(subch, activ_cell), TCV_ia_ts := '000'B, TCV_lac := lac, TCV_cksn :=cksn, TCV_CphKey := OC_CphKeyGen(TSPX_Ki, rand))			
2		+CCCH_group_Paging_group(ccd, imsi)			
3		+SelectPagingCh(activ_cell)			
4		[TSPC_PGSM OR TSPC_EGSM]			
5		(TCV_chdescr_arfcn:= arfcngsm)			
6		[TSPC_DCS]			
7		(TCV_chdescr_arfcn:= arfcndcs)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Wait(par_int:INTEGER)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective:
Default: OtherEventsFail
Comments: To check whether MS doesn't initiate any RR connections

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_dly(par_int)			
2		?TIMEOUT T_dly			

Detailed Comments: This test step can be used, e.g.for 1s. delay of a PAGING REQUEST message following immediately a CHANNEL RELEASE.

Test Step Dynamic Behaviour

Test Step Name: WaitForInService
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To wait until the MS enters the Idle and updated state.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Null:=OM_PgFill(TCV_cellid, PgReqTp1Reorg))			
2		START T_dly(5000)			
3		?TIMEOUT T_dly			
4		(TCV_Null:=OM_PgFill(TCV_cellid, PgReqTp1Norm))			

Detailed Comments:

Test Step Dynamic Behaviour					
Test Step Name:	ChgLAC_A(t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; mcc, mnc, lac:OCTETSTRING)				
Group:	GSM_L3_MS_v4220/Miscellaneous/				
Objective:	To change the LAC of cell A and of System Information Messages according to used testcase.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+SysInfoSending_fh(C_SCH_A, C_BCCH_A_1, t, retr, neci, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lac, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC_1, C_NCCP_2)			
2		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
3		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	ChgLAC_B(t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; mcc, mnc, lac:OCTETSTRING)				
Group:	GSM_L3_MS_v4220/Miscellaneous/				
Objective:	To change the LAC of cell B and of System Information Messages according to used testcase.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+SysInfoSending_fh(C_SCH_B, C_BCCH_B_1, t, retr, neci, att, babr, cch_con, bpm, t3212, C_ci_cellB, mcc, mnc, lac, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC_1, C_NCCP_2)			
2		(TCV_Ccd0B := TCV_Ccd0H, TCV_sysinfo5_B := TCV_sysinf5, TCV_sysinfo6_B := TCV_sysinf6)			
3		+SysInfo_SacchSending(TCV_sacch_B, TCV_sysinfo5_B, TCV_sysinfo6_B)			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: ChgLAI_C(att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, par_mnc, lac:OCTETSTRING)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To change the LAI of cell C to HPLMN.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := '0010'B, TCV_Max := '00'B, TCV_Neci := '0'B, TCV_Ccd0C := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		L!DL_UdatRqSysinfo4	SysInfo4(C_BCCH_C_1, mcc, par_mnc, lac, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_noRestablishment)		
4		L!DL_UdatRqSysinfo6	SysInfo6(TCV_sacch_C, SysInf6(C_ci_cellC, mcc, par_mnc, lac, CellOpt_01, C_NCCP_2))		
5		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_C_1, C_ci_cellC, mcc, par_mnc, lac, TCV_Ccd0C, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_noRestablishment)		
6		[TSPC_DCS]			
7		L!DL_UdatRqSysinfo4	SysInfo4(C_BCCH_C_1, mcc, par_mnc, lac, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_noRestablishment)		
8		L!DL_UdatRqSysinfo6	SysInfo6(TCV_sacch_C, SysInf6(C_ci_cellC, mcc, par_mnc, lac, CellOpt_01, C_NCCP_2))		
9		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_C_1, C_ci_cellC, mcc, par_mnc, lac, TCV_Ccd0C, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_noRestablishment)		

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_cbms(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcchfldcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP; Sub:B_2)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective: To send system information messages for the CBMS tests.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212), TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(Sub)), 5))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+ltree_sysinfo			
		ltree_sysinfo			
5		[TSPC_PGSM OR TSPC_EGSM]			
6		L!DL_UdatRqSysinfo4	SysInfo4_CBMS(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re, ChDescrp_nfhiei(TCV_Tchtype, C_S0, C_BCC, C_arfcnA))		
7		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
8		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
9		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
10		(TCV_sysinf5 := SysInf5(bcchflgsm))			
11		[TSPC_DCS]			
12		L!DL_UdatRqSysinfo4	SysInfo4_CBMS(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re, ChDescrp_nfhiei(TCV_Tchtype, C_S0, C_BCC, C_arfcnAd))		
13		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		
14		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
15		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
16		(TCV_sysinf5 := SysInf5(bcchfldcs))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_fh(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm, cchddcs:CCHD; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcchfldcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective: To send system information messages for the L3 tests. The following parameters specified by input parameters:
- bcch: BCCH
- sch: SCH
- bcchfl: BCCH frequency list
- ci: Cell identity
- lai: Location area identification
- ccd: Control channel description
- csp: Cell selection parameters
- cchd: Cell channel description
- rachcpar: RACH controll paramenters
- nccp: NCC permitted.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+ltree_sysinfo			
		ltree_sysinfo			
5		[TSPC_PGSM OR TSPC_EGSM]			
6		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchdgsm, TCV_Max, TCV_Tx, Re)		
7		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
8		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
9		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
10		(TCV_sysinf5 := SysInf5(bcchflgsm))			
11		[TSPC_DCS]			
12		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchddcs, TCV_Max, TCV_Tx, Re)		
13		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		
14		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
15		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
16		(TCV_sysinf5 := SysInf5(bcchfldcs), TCV_sysinf6.co.pwrc := '1'B)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_nfh(sch, bcch:LOGICCH; t, retr, neci, att: INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective: To send system information messages with default parameters defined for the L3 tests for which no special parameters indicated.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+gsmOrDcs			
		gsmOrDcs			
5		[TSPC_PGSM OR TSPC_EGSM]			1.
6		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
7		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
8		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
9		[NOT TSPX_AltNb]			3.
10		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
11		(TCV_sysinf5 := SysInf5(bcchflgsm))			
12		[TSPX_AltNb]			4.
13		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcflaltgsm, TCV_Max, TCV_Tx, Re, nccp)		
14		(TCV_sysinf5 := SysInf5(bcflaltgsm))			
15		[TSPC_DCS]			2.
16		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
17		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
18		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
19		[NOT TSPX_AltNb]			3.
20		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		

21		(TCV_sysinf5 := SysInf5(bcchfldcs))			
22		[TSPX_AltNb]			4.
23		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcflaltdcs, TCV_Max, TCV_Tx, Re, nccp)		
24		(TCV_sysinf5 := SysInf5(bcflaltdcs))			

Detailed Comments:

1. For GSM900 mobile station testing.
2. For DCS1800 mobile station testing.
3. To use alternative neighbour cells description.
4. To use default neighbour cells description.

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_nfhCb(sch, bcch:LOGICCH; t, retr, neci, att: INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective: To send system information messages with default parameters defined for the L3 tests for which no special parameters indicated.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+gsmOrDcs			
		gsmOrDcs			
5		[TSPC_PGSM OR TSPC_EGSM]			1.
6		L!DL_UdatRqSysinfo4 (DL_UdatRqSysinfo4.msg.rachcp.cba := '1'B)	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
7		L!DL_UdatRqSysinfo1_nh (DL_UdatRqSysinfo1_nh.msg.rachcp.cba := '1'B)	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
8		L!DL_UdatRqSysinfo3 (DL_UdatRqSysinfo3.msg.rachcp.cba := '1'B)	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
9		[NOT TSPX_AltNb]			3.
10		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.rachcp.cba := '1'B)	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
11		(TCV_sysinf5 := SysInf5(bcchflgsm))			
12		[TSPX_AltNb]			4.
13		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.rachcp.cba := '1'B)	SysInfo2(bcch, bcflaltgsm, TCV_Max, TCV_Tx, Re, nccp)		
14		(TCV_sysinf5 := SysInf5(bcflaltgsm))			
15		[TSPC_DCS]			2.
16		L!DL_UdatRqSysinfo4 (DL_UdatRqSysinfo4.msg.rachcp.cba := '1'B)	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
17		L!DL_UdatRqSysinfo1_nh (DL_UdatRqSysinfo1_nh.msg.rachcp.cba := '1'B)	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
18		L!DL_UdatRqSysinfo3 (DL_UdatRqSysinfo3.msg.rachcp.cba := '1'B)	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
19		[NOT TSPX_AltNb]			3.
20		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.rachcp.cba := '1'B)	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		

21		(TCV_sysinf5 := SysInf5(bcchfldcs))			
22		[TSPX_AltNb]			4.
23		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.rachcp.cba := '1'B)	SysInfo2(bcch, bcflaltdcs, TCV_Max, TCV_Tx, Re, nccp)		
24		(TCV_sysinf5 := SysInf5(bcflaltdcs))			

Detailed Comments:
1. For GSM900 mobile station testing.
2. For DCS1800 mobile station testing.
3. To use alternative neighbour cells description.
4. To use default neighbour cells description.

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_e(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm:CCHD; crh, mtmcgsm:INTEGER; bcchflgsm, bcchflegsm:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP)

Group: GSM_L3_MS_v4220/Miscellaneous/

Objective: To send SYNCHRONIZATION CHANNEL INFORMATION and SYSTEM INFORMATION messages with default parameters except CCCH_CONF, Tx-integer, Max-Retrans and ATT which are specified by input parameters.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		1.
3		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
4		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchdgsm, TCV_Max, TCV_Tx, Re)		2.
5		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.bcchfl.extind := '1'B)	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
6		L!DL_UdatRqSysinfo2bis	SysInfo2bis(bcch, bcchflegsm)		
7		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
8		(TCV_sysinf5 := SysInf5(bcchflgsm), TCV_sysinfo5.bcchfl.extind := '1'B)			
9		(TCV_sysinfo5bis := SysInf5bis(bcchflegsm))			
10		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		

Detailed Comments:
1. To send SYNCHRONIZATION INFORMATION message with default parameter for cell A.
2. System Information Type 1 with Cell Channel Description for HO-testin in cell A (GSM900).

Test Step Dynamic Behaviour					
Test Step Name:	SysInfo_SacchSending(ch:LOGICCH; sysinfo5_pdu:SYSINFO5_PDU; sysinfo6_pdu:SYSINFO6_PDU)				
Group:	GSM_L3_MS_v4220/Miscellaneous/				
Objective:	To send SYSTEM INFORMATION 5 and 6 messages defined by parameters 'sysinfo5_pdu' and 'sysinfo6_pdu' in the parametrised 'ch' channel.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[ch <> "dummy"]			
2		L!DL_UdatRqSysinfo6	SysInfo6(ch, sysinfo6_pdu)		
3		L!DL_UdatRqSysinfo5	SysInfo5(ch, sysinfo5_pdu)		
4	L_0920	[ch = "dummy"]		I	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	SysInfo_5bisSending(ch:LOGICCH; sysinfo5bis_pdu:SYSINFO5bis_PDU)				
Group:	GSM_L3_MS_v4220/Miscellaneous/				
Objective:	To send SYSTEM INFORMATION 5bis message defined by parameters 'sysinfo5bis_pdu' in the parametrised 'ch' channel.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[ch <> "dummy"]			
2		L!DL_UdatRqSysinfo5bis	SysInfo5bis(ch, sysinfo5bis_pdu)		
3	L_0921	[ch = "dummy"]		I	
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: SetNECI(att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To set the NECI =1.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := '0010'B, TCV_Max := '11'B, TCV_Neci := '1'B, TCV_Ccd0A := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		L!DL_UdatRqSysinfo4	SysInfo4(C_BCCH_A_1, mcc, mnc, lac, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
4		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
5		[TSPC_DCS]			
6		L!DL_UdatRqSysinfo4	SysInfo4(C_BCCH_A_1, mcc, mnc, lac, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
7		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SetATT(t, retr, att, neci:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING)
Group: GSM_L3_MS_v4220/Miscellaneous/
Objective: To set the ATT flag to "MS's in the cell should apply IMSI attach and detach procedure"
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0A := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		[TSPC_PGSM OR TSPC_EGSM]			1.
3		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
4		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
5		START T_dly(5000)			
6		?TIMEOUT T_dly			3.
7		[TSPC_DCS]			2.
8		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
9		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
10		START T_dly(5000)			3.
11		?TIMEOUT T_dly			

Detailed Comments:
1. For P-GSM900.
2. For DCS1800.
3. Wait for 5 seconds to allow the MS read BCCH information.

Test Step Group OperatorOP

Test Step Dynamic Behaviour					
Test Step Name:	InitCall(srv: SERVICES)				
Group:	GSM_L3_MS_v4220/OperatorOP/				
Objective:	To initiate a call of the basic service `srv`.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OO_InitCall(srv))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	InitCM(srv: SERVICES)				
Group:	GSM_L3_MS_v4220/OperatorOP/				
Objective:	To initiate a CM service request.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_MOsvc]			
2		(TCV_Null := OO_InitCall(srv))			
3		[TSPC_NonCallSS]			
4		(TCV_Null := OO_InitNonCallSS())			
5		[TSPC_Serv_TS22]			
6		(TCV_Null := OO_SendMOShortMessage())			
7	L_1048	[NOT TSPC_MOsvc AND NOT TSPC_Serv_TS22 AND NOT TSPC_NonCallSS]		I	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	InitNonCallSupp				
Group:	GSM_L3_MS_v4220/OperatorOP/				
Objective:	To attempt a non call related supplementary service at the MS under test.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OO_InitNonCallSS())			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	AtmpShortMsg				
Group:	GSM_L3_MS_v4220/OperatorOP/				
Objective:	To attempt a short message service transaction at the MS under test.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OO_SendMOShortMessage())			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	CheckUssdStringDisplayed(strg: IA5String)				
Group:	GSM_L3_MS_v4220/OperatorOP/				
Objective:	To check whether the correct USSD String 'strg' is displayed on the MS				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Res := OO_CheckUssdStringDisplayed(strg))			
2	L_0922	[TCV_Res]		(P)	
3	L_0923	[NOT TCV_Res]		(F)	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	PLMNsCHK				
Group:	GSM_L3_MS_v4220/OperatorOP/				
Objective:	To check whether the MS presents a list of available PLMNs.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Res := OO_PLMNsCHK())			
2	L_0924	[TCV_Res]		P	
3	L_0925	[NOT TCV_Res]		F	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	RFtransCHK				
Group:	GSM_L3_MS_v4220/OperatorOP/				
Objective:	To check whether the MS transmits any radio signal.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Res := OO_RFoutputCHK())			
2	L_0926	[TCV_Res]		F	
3	L_0927	[NOT TCV_Res]		(P)	
Detailed Comments:					

Test Step Group Postambles

Test Step Dynamic Behaviour					
Test Step Name:	PostLinkRelEnd(ch: LOGICCH)				
Group:	GSM_L3_MS_v4220/Postambles/				
Objective:	To release the RR connection and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(ch, ChRelease_01)		
2		L?DL_RelIn	DLRelInd_01		
3	L_0928	CANCEL		R	1.
4		?TIMEOUT T_release			
5	L_0929	CANCEL		R	1.
Detailed Comments:		1. Cancel of all running timers and final verdict.			

Test Step Dynamic Behaviour					
Test Step Name:	PostMainLinkRel(chnl:LOGICCH)				
Group:	GSM_L3_MS_v4220/Postambles/				
Objective:	To release the main signalling link `ch`, and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(chnl, ChRelease_01)		
2		L?DL_RelIn CANCEL T_release	DLRelInd_01		
3		?TIMEOUT T_release			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	ChanRel(ch: LOGICCH)				
Group:	GSM_L3_MS_v4220/Postambles/				
Objective:	To release the RR connection on the channel TCV_ch and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(ch, ChRelease_01)		
2		L?DL_RelIn CANCEL T_release	DLRelInd_01		
3	L_0930	?TIMEOUT T_release		(F)	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	ChanRel_P(ch: LOGICCH)				
Group:	GSM_L3_MS_v4220/Postambles/				
Objective:	To release the RR connection on the channel TCV_chmaindcch and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(ch, ChRelease_01)		
2	L_0931	L?DL_RelIn	DLRelInd_01	(P)	
3		?TIMEOUT T_release			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	ChanRel_end(ch: LOGICCH)				
Group:	GSM_L3_MS_v4220/Postambles/				
Objective:	To release the RR connection and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(ch, ChRelease_01)		
2		L?DL_RelIn	DLRelInd_01		
3	L_0932	CANCEL		(P)	1.
4		?TIMEOUT T_release			
5	L_0933	CANCEL		(F)	1.
Detailed Comments:		1. Cancel of all running timers.			

Test Step Dynamic Behaviour					
Test Step Name:	RmvForbiddenPlmn(activ_cell:CellID; lac:OCTETSTRING; subch:B_2; cksn:B_3; rand:RAND; arfcngsm, arfcndcs:INTEGER; ccd: CCD; imsi:HEXSTRING)				
Group:	GSM_L3_MS_v4220/Postambles/				
Objective:	To remove a forbidden PLMN in the list on SIM.				
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+StartCellB(C_E_default, arfcngsm, arfcndcs, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, lac, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			
2		+Varinit_fix(activ_cell, lac, subch, cksn, rand, arfcngsm, arfcndcs, ccd, imsi)			
3		(TCV_Null := OM_StopCell(C_CellA))			
4		(TCV_Null := OO_PLMNselModeMan())			
5		(TCV_Null := OO_SelPLMN(C_PLMN_2))			
6		+MM_LUP3(C_MCC_1, C_PLMN_2, lac, C_norm_period_attach, TimingAdv(30))			
7		(TCV_Null := OO_PLMNselModeAuto())			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	RmvForbiddenPlmn1(activ_cell:CellID; lac:OCTETSTRING; subch:B_2; cksn:B_3; rand:RAND; arfcngsm, arfcndcs:INTEGER; ccd: CCD; imsi:HEXSTRING)				
Group:	GSM_L3_MS_v4220/Postambles/				
Objective:	To remove a forbidden PLMN in the list on SIM.				
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+StartCellB(C_E_default, arfcngsm, arfcndcs, C_Immass, TCV_slot, TCV_tsc, TimingAdv(30), C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_1, C_MCC_1, C_PLMN_2, lac, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			
2		+Varinit_fix(activ_cell, lac, subch, cksn, rand, arfcngsm, arfcndcs, ccd, imsi)			
3		(TCV_Null := OO_PLMNselModeMan())			
4		(TCV_Null := OO_SelPLMN(C_PLMN_2))			
5		+MM_LUP3(C_MCC_1, C_PLMN_2, lac, C_norm_period_attach, TimingAdv(30))			
6		(TCV_Null := OO_PLMNselModeAuto())			
Detailed Comments:					

Test Step Group Preambles

Test Step Dynamic Behaviour					
Test Step Name:	BasicServiceMO(svc:SERVICES; rate:RATE)				
Group:	GSM_L3_MS_v4220/Preambles/				
Objective:	To generate a Bcap, an MO SETUP message and an appropriate channel mode based on the service and rate, as well as on several IXIT parameters, itc, ce, ur, synchronous bit for an MO call.				
Default:	OtherEvents				
Comments:	Used for CC tests				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+local_init			Default values
2		[TCV_Service = C_EmgCall]			
3		[(NOT TSPC_EFR_Speech_v2) AND(NOT TSPC_EFR_Speech_v3)]			
4		(TCV_ChMod.mode := C_ChMod_r, TCV_Esetup := Setup_04(Bcap_Speech_MO(TCV_Rchr)), TCV_Ecall := C_Yes)			Emergency Call - Non EFR Mobile
5		[TSPC_EFR_Speech_v2 OR TSPC_EFR_Speech_v3]			
6		(TCV_ChMod.mode := C_ChMod_r, TCV_Esetup := Setup_04(Bcap_Speech_Efr_MO(TCV_Rchr)), TCV_Ecall := C_Yes)			Emergency Call - EFR Mobile
7		[TCV_Service = C_Telephony]			
8		[(NOT TSPC_EFR_Speech_v2) AND(NOT TSPC_EFR_Speech_v3)]			
9		(TCV_ChMod.mode := C_ChMod_r, TCV_Setup_mo := Setup_02(Bcap_Speech_MO(TCV_Rchr)))			Speech - Non EFR Mobile
10		[TSPC_EFR_Speech_v2 OR TSPC_EFR_Speech_v3]			
11		(TCV_ChMod.mode := C_ChMod_r, TCV_Setup_mo := Setup_02(Bcap_Speech_Efr_MO(TCV_Rchr)))			Speech - EFR Mobile
12		[TCV_Service = C_AltSpchFax]			
13		[rate = C_Full]			
14		+TS61MO(C_AltSpchFax)			Speech/fax
15	L_0934	[rate <> C_Full]		I	IXIT error
16		[TCV_Service = C_AutoFax]			
17		[rate = C_Full]			
18		+TS62MO(C_AutoFax)			Auto fax
19	L_0935	[rate <> C_Full]		I	IXIT error
20		[TCV_Service = C_Async300]			
21		+BS2xMO(C_Async300, TSPX_BS_21_sacp, TSPX_BS_21_T_NT, TSPX_BS_21_itc1, C_300bs, C_modemt_V21, TSPX_BS_21_ce)			BS21
22		[TCV_Service = C_Async1200]			
23		+BS2xMO(C_Async1200, TSPX_BS_22_sacp, TSPX_BS_22_T_NT, TSPX_BS_22_itc1, C_1200bs, C_modemt_V22, TSPX_BS_22_ce)			BS22
24		[TCV_Service = C_Async120075]			
25		+BS2xMO(C_Async120075, TSPX_BS_23_sacp, TSPX_BS_23_T_NT, TSPX_BS_23_itc, C_120075bs, C_modemt_V23, TSPX_BS_23_ce)			BS23
26		[TCV_Service = C_Async2400]			
27		(TCV_bcapListIndicator := C_modemList_ce)			
28		+BS2xMO(C_Async2400, TSPX_BS_24_sacp, TSPX_BS_24_T_NT, TSPX_BS_24_itc1, C_2400bs, TCV_MODEM, TSPX_BS_24_ce)			BS24
29		[TCV_Service = C_Async4800]			
30		+BS2xMO(C_Async4800, TSPX_BS_25_sacp, TSPX_BS_25_T_NT, TSPX_BS_25_itc1, C_4800bs, C_modemt_V32, TSPX_BS_25_ce)			BS25
31		[TCV_Service = C_Async9600]			
32		+BS2xMO(C_Async9600, TSPX_BS_26_sacp, TSPX_BS_26_T_NT, TSPX_BS_26_itc1,			BS26

		C_9600bs, C_modemt_V32, TSPX_BS_26_ce)			
33		[TCV_Service = C_Sync1200]			
34		+BS3xMO(C_Sync1200, FALSE, TSPX_BS_31_itc1, TSPX_BS_31_sacp1, C_1200bs, C_transparent, C_modemt_V22)			BS31
35		[TCV_Service = C_Sync2400]			
36		(TCV_bcapListIndicator := C_modemList_ce)			
37		+BS3xMO(C_Sync2400, TSPX_BS_32_X32_T_NT, TSPX_BS_32_itc1, TSPX_BS_32_sacp1, C_2400bs, TSPX_BS_32_X32_ce, TCV_MODEM)			BS32
38		[TCV_Service = C_Sync4800]			
39		+BS3xMO(C_Sync4800, TSPX_BS_33_X32_T_NT, TSPX_BS_33_itc1, TSPX_BS_33_sacp1, C_4800bs, TSPX_BS_33_X32_ce, C_modemt_V32)			BS33
40		[TCV_Service = C_Sync9600]			
41		+BS3xMO(C_Sync9600, TSPX_BS_34_X32_T_NT, TSPX_BS_34_itc1, TSPX_BS_34_sacp1, C_9600bs, TSPX_BS_34_X32_ce, C_modemt_V32)			BS34
42		[TCV_Service = C_PAD300]			
43		+BS4xMO(C_PAD300, C_300bs, TSPX_BS_41_ce, TSPX_BS_41_T_NT)			BS41
44		[TCV_Service = C_PAD1200]			
45		+BS4xMO(C_PAD1200, C_1200bs, TSPX_BS_42_ce, TSPX_BS_42_T_NT)			BS42
46		[TCV_Service = C_PAD120075]			
47		+BS4xMO(C_PAD120075, C_120075bs, TSPX_BS_43_ce, TSPX_BS_43_T_NT)			BS43
48		[TCV_Service = C_PAD2400]			
49		+BS4xMO(C_PAD2400, C_2400bs, TSPX_BS_44_ce, TSPX_BS_44_T_NT)			BS44
50		[TCV_Service = C_PAD4800]			
51		+BS4xMO(C_PAD4800, C_4800bs, TSPX_BS_45_ce, TSPX_BS_45_T_NT)			BS45
52		[TCV_Service = C_PAD9600]			
53		+BS4xMO(C_PAD9600, C_9600bs, TSPX_BS_46_ce, TSPX_BS_46_T_NT)			BS46
54		[TCV_Service = C_Packet2400]			
55		+BS5xMO(C_Packet2400, C_2400bs)			BS51
56		[TCV_Service = C_Packet4800]			
57		+BS5xMO(C_Packet4800, C_4800bs)			BS52
58		[TCV_Service = C_Packet9600]			
59		+BS5xMO(C_Packet9600, C_9600bs)			BS53
60		[TCV_Service = C_AltSpchData]			
61		+BS61or81MO(C_AltSpchData, TSPX_BS_61_A, TSPX_BS_61_A_ur1_T_NT, TSPX_BS_61_A_ur1, TSPX_BS_61_A_ur1_ce, TSPX_BS_61_S, TSPX_BS_61_S_ur1, C_RI_alternate)			BS61
62		[TCV_Service = C_SpchData]			
63		+BS61or81MO(C_SpchData, TSPX_BS_81_A, TSPX_BS_81_A_ur1_T_NT, TSPX_BS_81_A_ur1, TSPX_BS_81_A_ur1_ce, TSPX_BS_81_S, TSPX_BS_81_S_ur1, C_RI_follow)			BS81
64	L_0936	[C_Yes]		I	IXIT error
		local_init			
65		(TCV_Service := svc, TCV_Rchr := C_Rchr_Full, TCV_Ecall := C_No, TCV_ChRate :=rate, TCV_ChMod.iei := '01100011'B, TCV_ChMod.mode := C_ChMod_12k, TCV_ChModb.iei := '01100011'B, TCV_ChModb.mode := C_ChMod_12k, TCV_MODEM := C_modemt_none, TCV_bcapListIndicator := C_modemt_ce)			
66		[TSPC_DualRate]			
67		(TCV_Rchr := C_Rchr_DualForDualH0)			

68		[NOT TSPC_DualRate]			

Detailed Comments: This test step is called in the preamble of test cases where an MO call is established in the test body.

Test Step Dynamic Behaviour

Test Step Name: BasicServiceMOorTelephony(svc:SERVICES; rate:RATE)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To get a MO SETUP message with right BC IE.
Default: OtherEvents
Comments: Used for CC tests

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_Serv_TS11]			
2		+BasicServiceMO(C_Telephony,TSPX_Telephony_Rate)			
3		[NOT TSPC_Serv_TS11]			
4		+BasicServiceMO(svc,rate)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: BasicServiceMT(svc:SERVICES; rate:RATE)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To generate a Bcap, an MT SETUP message and an appropriate channel mode based on the service and rate, as well as on several IXIT parameters, itc, ce, ur, synchronous bit for an MT call.
To generate a second SETUP message for TC_11_1_1.

Default: OtherEvents

Comments: Used for CC tests

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Service := svc, TCV_ChRate :=rate, TCV_ChMod.iei := '01100011'B, TCV_ChMod.mode := C_ChMod_12k, TCV_ChModb.iei := '01100011'B, TCV_ChModb.mode := C_ChMod_12k, TCV_2ndtest := C_No)			Default values
2		+SvcSupportedChk(svc)			
3		+local_main **local_main**			
4		[TCV_Service = C_Telephony]			
5		(TCV_ChMod.mode := C_ChMod_r, TCV_Setup_mt := Setup_01(TI_02))			Speech
6		+local_IFsetup_Telephony			Config speech
7		[TCV_Service = C_AltSpchFax]			
8		+TS61MT(C_AltSpchFax)			Speech/fax
9		[TCV_Service = C_AutoFax]			
10		+TS62MT(C_AutoFax)			Auto fax
11		[TCV_Service = C_Async300]			
12		+BS2xMT(C_Async300, TSPX_BS_21_more_itc, TSPX_BS_21_T_NT, TSPX_BS_21_itc1, TSPX_BS_21_itc2, C_300bs, C_modemt_V21, TSPX_BS_21_ce)			BS21
13		[TCV_Service = C_Async1200]			
14		+BS2xMT(C_Async1200, TSPX_BS_22_more_itc, TSPX_BS_22_T_NT, TSPX_BS_22_itc1, TSPX_BS_22_itc2, C_1200bs, C_modemt_V22, TSPX_BS_22_ce)			BS22
15		[TCV_Service = C_Async2400]			
16		+BS2xMT(C_Async2400, TSPX_BS_24_more_itc, TSPX_BS_24_T_NT, TSPX_BS_24_itc1, TSPX_BS_24_itc2, C_2400bs, C_modemt_V22bis, TSPX_BS_24_ce)			BS24
17		[TCV_Service = C_Async4800]			
18		+BS2xMT(C_Async4800, TSPX_BS_25_more_itc, TSPX_BS_25_T_NT, TSPX_BS_25_itc1, TSPX_BS_25_itc2, C_4800bs, C_modemt_V32, TSPX_BS_25_ce)			BS25
19		[TCV_Service = C_Async9600]			
20		+BS2xMT(C_Async9600, TSPX_BS_26_more_itc, TSPX_BS_26_T_NT, TSPX_BS_26_itc1, TSPX_BS_26_itc2, C_9600bs, C_modemt_V32, TSPX_BS_26_ce)			BS26
21		[TCV_Service = C_Sync1200]			
22		+BS3xMT(C_Sync1200, TSPX_BS_31_more_itc, C_No, TSPX_BS_31_more_sacp, TSPX_BS_31_itc1, TSPX_BS_31_itc2, TSPX_BS_31_sacp1, TSPX_BS_31_sacp2, C_1200bs, C_transparent)			BS31
23		[TCV_Service = C_Sync2400]			
24		+BS3xMT(C_Sync2400, TSPX_BS_32_more_itc, TSPX_BS_32_X32_T_NT, TSPX_BS_32_more_sacp, TSPX_BS_32_itc1, TSPX_BS_32_itc2, TSPX_BS_32_sacp1, TSPX_BS_32_sacp2, C_2400bs, TSPX_BS_32_X32_ce)			BS32
25		[TCV_Service = C_Sync4800]			
26		+BS3xMT(C_Sync4800, TSPX_BS_33_more_itc, TSPX_BS_33_X32_T_NT, TSPX_BS_33_more_sacp, TSPX_BS_33_itc1, TSPX_BS_33_itc2, TSPX_BS_33_sacp1, TSPX_BS_33_sacp2, C_4800bs,			BS33

		TSPX_BS_33_X32_ce)			
27		[TCV_Service = C_Sync9600]			
28		+BS3xMT(C_Sync9600, TSPX_BS_34_more_itc, TSPX_BS_34_X32_T_NT, TSPX_BS_34_more_sacp, TSPX_BS_34_itc1, TSPX_BS_34_itc2, TSPX_BS_34_sacp1, TSPX_BS_34_sacp2, C_9600bs, TSPX_BS_34_X32_ce)			BS34
29		[TCV_Service = C_AltSpchData]			
30		+BS61or81MT(C_AltSpchData, TSPX_BS_61_A, TSPX_BS_61_A_ur1_T_NT, TSPX_BS_61_A_ur2_T_NT, TSPX_BS_61_A_more_ur, TSPX_BS_61_S, TSPX_BS_61_S_more_ur, TSPX_BS_61_A_ur1, TSPX_BS_61_A_ur2, TSPX_BS_61_S_ur1, TSPX_BS_61_S_ur2, C_RI_alternate, TSPX_BS_61_A_ur1_ce)			BS61
31		[TCV_Service = C_SpchData]			
32		+BS61or81MT(C_SpchData, TSPX_BS_81_A, TSPX_BS_81_A_ur1_T_NT, TSPX_BS_81_A_ur2_T_NT, TSPX_BS_81_A_more_ur, TSPX_BS_81_S, TSPX_BS_81_S_more_ur, TSPX_BS_81_A_ur1, TSPX_BS_81_A_ur2, TSPX_BS_81_S_ur1, TSPX_BS_81_S_ur2, C_RI_follow, TSPX_BS_81_A_ur1_ce)			BS81
33	L_0937	[(TCV_Service <> C_Telephony) AND(TCV_Service <> C_AltSpchFax) AND(TCV_Service <> C_AutoFax) AND(TCV_Service <> C_Async300) AND(TCV_Service <> C_Async1200) AND(TCV_Service <> C_Async2400) AND(TCV_Service <> C_Async4800) AND(TCV_Service <> C_Async9600) AND(TCV_Service <> C_Sync1200) AND(TCV_Service <> C_Sync2400) AND(TCV_Service <> C_Sync4800) AND(TCV_Service <> C_Sync9600) AND(TCV_Service <> C_AltSpchData) AND(TCV_Service <> C_SpchData)]		I	
		local_IFsetup_Telephony			
34		[TCV_supported]			
35		(TCV_Null := OO_IFsetup_Telephony())			
36		[NOT TCV_supported]			
37		[TCV_SpecialCase]			
38		START T_dly(C_T_Wait)			
39		?TIMEOUT T_dly			
40	L_0938	[NOT TCV_SpecialCase]		I	parameter error

Detailed Comments: This test step is called in the preamble of test cases where an MT call is established in the test body.

Test Step Dynamic Behaviour

Test Step Name: BasicServiceMTorTelephony(svc:SERVICES; rate:RATE)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To get a MT SETUP message with right BC IE.
Default: OtherEvents
Comments: Used for CC tests

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_Serv_TS11]			
2		+BasicServiceMT(C_Telephony,TSPX_Telephony_Rate)			
3		[NOT TSPC_Serv_TS11]			
4		+BasicServiceMT(svc,rate)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: BS2xMO(srv:SERVICES; sacp: B_3; both:BOOLEAN; itc1:B_3; ur:B_4; modem:B_5; ce:B_2)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To generate an MO SETUP message with appropriate Bcap and LLC for BS2x service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_srv := srv, TCV_ur := ur, TCV_CE := C_BothT, TCV_ir := C_ir_16kbs)			
2		+local_pixitvaluechk			validate IXIT values
3		+local_selectCe			select a connection element
4		+local_bcap			generate BC
5		+local_setupmsg			generate SETUP
6		+Chmod(TCV_ChRate, TCV_ce, ur, srv)			initiate channel mode variables
7		(TCV_Null := OO_IFsetup_BS2xorBS3x_MO(TCV_srv, itc1, sacp, TCV_ce, TCV_ChRate))			ask test operator to configure the MS
		local_bcap			
8		[NOT both]			
9		[TCV_ce = C_notransparent]			
10		[itc1 = C_UDI]			
11		+local_BS2x_UDI_NT(C_notransparent, sacp)			
12		[itc1 = C_3100Hz]			
13		+local_BS2x_3100_NT(C_notransparent, sacp)			
14		[TCV_ce = C_transparent]			CE:T only MS
15		[itc1 = C_UDI]			
16		+local_BS2x_UDI_T(sacp)			
17		[itc1 = C_3100Hz]			
18		+local_BS2x_3100_T(sacp)			
19		[both]			
20		[TCV_ce = C_transparent]			
21		(TCV_bcapListIndicator := TCV_bcapListIndicator + 1)			
22		[itc1 = C_UDI]			
23		+local_BS2x_UDI_Tboth(TCV_CE,sacp)			
24		[itc1 = C_3100Hz]			3.1kHz
25		+local_BS2x_3100_Tboth(TCV_CE, sacp)			
26		[(TCV_ce = C_notransparent)]			
27		(TCV_bcapListIndicator := TCV_bcapListIndicator + 2)			
28		[itc1 = C_UDI]			
29		+local_BS2x_UDI_NT(TCV_CE,sacp)			
30		[itc1 = C_3100Hz]			3.1kHz
31		+local_BS2x_3100_NT(TCV_CE, sacp)			
		local_BS2x_3100_NT(cel: B_2; sacp:B_3)			
32		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, C_SDUintegrity, C_nirr_dontcare, C_no_rate_adaption, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, cel, modem, C_uil2p_dontcare), TCV_bcapListIndicator))			11.8.2.1.1.2
		local_BS2x_UDI_NT(cel: B_2; sacp:B_3)			
33		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_UDI, C_SDUintegrity, C_nirr_dontcare, C_rate_adaption_V110, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, cel, C_modemt_none, C_uil2p_dontcare), TCV_bcapListIndicator))			11.8.2.1.1.4
		local_BS2x_3100_T(sacp:B_3)			

34		[ur = C_9600bs]			
35		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, C_transparent, modem))			11.8.2.1.1.1
36		[(ur = C_300bs) OR (ur = C_1200bs) OR (ur = C_120075bs) OR (ur = C_2400bs) OR (ur = C_4800bs)]			
37		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent, modem), TCV_ir := C_ir_8kbs)			11.8.2.1.1.1
		local_BS2x_UDI_T(sacp:B_3)			
38		[ur = C_9600bs]			
39		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, C_transparent, C_modemt_none))			11.8.2.1.1.3
40		[(ur = C_300bs) OR (ur = C_1200bs) OR (ur = C_120075bs) OR (ur = C_2400bs) OR (ur = C_4800bs)]			
41		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent, C_modemt_none), TCV_ir := C_ir_8kbs)			11.8.2.1.1.3
		local_BS2x_3100_Tboth(cel: B_2; sacp:B_3)			
42		[ur <> C_9600bs]			
43		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, C_struc_dontchk, C_nirr_dontcare, C_no_rate_adaption, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8or16kbs, C_parity_dontcare, cel, modem, C_uil2p_dontcare), TCV_bcapListIndicator))			11.8.2.1.1.1 conbinrd with 11.8.2.1.1.2 , CE and structure without exact match
44		[ur = C_9600bs]			
45		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, C_struc_dontchk, C_nirr_nomeaning, C_no_rate_adaption, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8or16kbs, C_parity_dontcare, cel, modem, C_uil2p_dontcare), TCV_bcapListIndicator))			11.8.2.1.1.1 conbinrd with 11.8.2.1.1.2 , CE and structure without exact match
		local_BS2x_UDI_Tboth(cel: B_2; sacp:B_3)			
46		[ur = C_9600bs]			
47		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_UDI, C_struc_dontchk, C_nirr_nomeaning, C_rate_adaption_V110, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8or16kbs, C_parity_dontcare, cel, C_modemt_none, C_uil2p_dontcare), TCV_bcapListIndicator))			11.8.2.1.1.3 conbined with 11.8.2.1.1.4, CE and structure without exact match
48		[ur <> C_9600bs]			
49		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_UDI, C_struc_dontchk, C_nirr_dontcare, C_rate_adaption_V110, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8or16kbs, C_parity_dontcare, cel, C_modemt_none, C_uil2p_dontcare), TCV_bcapListIndicator))			11.8.2.1.1.3 conbined with 11.8.2.1.1.4, CE and structure without exact match

		local_setupmsg			
50		[itc1 = C_UDI]			
51		(TCV_Setup_mo := Setup_10_UDI(TCV_BcapMO1, Llcmp_BS2xOrBS4x_UDI, Hlcmp_AnyOrOmit))			
52		[itc1 = C_3100Hz]			
53		(TCV_Setup_mo := Setup_10_3100(TCV_BcapMO1, Llcmp_BS2xOrBS61aOrBS81a_3100, Hlcmp_AnyOrOmit))			
		local_selectCe			
54		[both]			
55		[(ur = C_300bs) OR (ur = C_120075bs) OR (ur = C_4800bs)]			
56		(TCV_ce := C_notransparent)			
57		[(ur = C_1200bs) OR (ur = C_2400bs) OR (ur = C_9600bs)]			
58		(TCV_ce := C_transparent)			
59		[NOT both]			
60		(TCV_ce := ce)			
		local_pixitvaluechk			
61	L_0939	[(ce <> C_transparent) AND(ce <> C_notransparent) AND(NOT both)]		I	IXIT error
62	L_0940	[(itc1 <> C_UDI) AND(itc1 <> C_3100Hz)]		I	IXIT error
63	L_0941	[(sacp <> C_I440_450) AND(sacp <> C_X28_nond)]		I	IXIT error
64		[C_Yes]			IXIT values OK

Detailed Comments:

The BS2x Bearer Capabilities and SETUP messages for MO is constructed according to section 11.8.2.1.1.1, section 11.8.2.1.1.2, section 11.8.2.1.1.3 and section 11.8.2.1.1.4 of GSM 11.10 and section B.1.2.1, section B.1.2.2, section B.1.2.1 and section B.2.2.2 of GSM 07.01.

The algorithm for selecting parameters:

```
IF (TSPX_BS_2x_T_NT)
        TCV_ce := C_notransparent
ELSE
        TCV_ce := TSPX_BS_2x_ce
```

use (TCV_ce, TSPX_BS_2x_itc1, TSPX_BS_2x_sacp1, svc) for OO_IFsetup_BS2xorBS3x_MO

Test Step Dynamic Behaviour

Test Step Name: BS2xMT(srv:SERVICES; more, both:BOOLEAN; itc1, itc2:B_3; ur:B_4; modem:B_5; ce:B_2)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To generate an MT SETUP message with appropriate Bcap for BS2x service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+local_main			
2		+local_IFsetup			
		local_main			
3		(TCV_srv := srv, TCV_itc := itc1)			
4		+local_bcap1			
5		(TCV_Setup_mt := Setup_20(TCV_BcapMT1, TI_02))			1st SETUP
6		+Chmod(TCV_ChRate, TCV_ce, ur, srv)			Ini ch mode
7		+local_bcap2			
8		(TCV_Setup_mt1:= Setup_20(TCV_BcapMT2, TI_02), TCV_2ndtest := C_Yes)			2nd SETUP
		local_bcap1			
9		[itc1 = C_3100Hz]			3.1kHz
10		[both OR (ce = C_notransparent)]			
11		+local_BS2x_3100_NT(C_notransparent)			
12		[NOT both AND (ce = C_transparent)]			
13		+local_BS2x_3100_T			
14		[itc1 = C_UDI]			UDI
15		[both OR (ce = C_notransparent)]			
16		+local_BS2x_UDI_NT(C_notransparent)			
17		[NOT both AND (ce = C_transparent)]			
18		+local_BS2x_UDI_T			
19	L_0942	[(itc1 <> C_3100Hz) AND(itc1 <> C_UDI)]		I	
		local_bcap2			
20		+local_itc			
21		+local_bcap			
22		(TCV_BcapMT2 := TCV_BcapMT1)			
		local_itc			
23		(TCV_itc2 := itc1)			
24		[more]			3.1 kHz
25		(TCV_itc2 := itc2)			
26		[NOT more]			
		local_bcap			
27		[both AND (TCV_itc2 = C_3100Hz)]			
28		+local_BS2x_3100_T			
29		[both AND (TCV_itc2 = C_UDI)]			UDI
30		+local_BS2x_UDI_T			
31		[NOT both AND (TCV_itc2 = C_3100Hz)]			
32		+local_BS2x_3100_NT(C_BothT)			
33		[NOT both AND (TCV_itc2 = C_UDI)]			UDI
34		+local_BS2x_UDI_NT(C_BothT)			
35	L_0943	[(TCV_itc2 <> C_3100Hz) AND(TCV_itc2 <> C_UDI)]		I	
		local_BS2x_3100_NT(cel: B_2)			
36		[TSPX_TE_FLCT = C_Inband]			
37		(TCV_BcapMT1 := BcapX_MT(C_3100Hz, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, C_ir_16kbs, TSPX_TE_parity, cel, modem, C_ISO6429), TCV_ce := cel)			11.8.1.1.1.2

38		[TSPX_TE_FLCT = C_Nocontrol]			
39		(TCV_BcapMT1 := BcapX_MT(C_3100Hz, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, C_ir_16kbs, TSPX_TE_parity, cel, modem, C_COPnoFLCT), TCV_ce := cel)			11.8.1.1.1.2
40	L_0944	[(TSPX_TE_FLCT = C_Outband) OR((TSPX_TE_FLCT <> C_Inband) AND(TSPX_TE_FLCT <> C_Nocontrol))]		I	
		local_BS2x_UDI_NT(cel: B_2)			
41		[TSPX_TE_FLCT = C_Outband]			
42		(TCV_BcapMT1 := Bcap_MT(C_UDI, C_SDUintegrity, C_nirr_nomeaning, C_rate_adaption_V110, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, C_ir_16kbs, TSPX_TE_parity, cel, C_modemt_none), TCV_ce := cel)			11.8.1.1.1.4
43		[TSPX_TE_FLCT = C_Inband]			
44		(TCV_BcapMT1 := BcapX_MT(C_UDI, C_SDUintegrity, C_nirr_nomeaning, C_rate_adaption_V110, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, C_ir_16kbs, TSPX_TE_parity, cel, C_modemt_none, C_ISO6429), TCV_ce := cel)			11.8.1.1.1.4
45		[TSPX_TE_FLCT = C_Nocontrol]			
46		(TCV_BcapMT1 := BcapX_MT(C_UDI, C_SDUintegrity, C_nirr_nomeaning, C_rate_adaption_V110, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, C_ir_16kbs, TSPX_TE_parity, cel, C_modemt_none, C_COPnoFLCT), TCV_ce := cel)			11.8.1.1.1.4
47	L_0945	[(TSPX_TE_FLCT <> C_Outband) AND(TSPX_TE_FLCT <> C_Inband) AND(TSPX_TE_FLCT <> C_Nocontrol)]		I	
		local_BS2x_3100_T			
48		[ur = C_9600bs]			
49		(TCV_BcapMT1 := Bcap_MT(C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, C_ir_16kbs, TSPX_TE_parity, C_transparent, modem), TCV_itc2 := C_3100Hz, TCV_ce := C_transparent)			11.8.1.1.1.1
50		[ur <> C_9600bs]			
51		(TCV_BcapMT1 := Bcap_MT(C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, C_ir_8kbs, TSPX_TE_parity, C_transparent, modem), TCV_itc2 := C_3100Hz, TCV_ce := C_transparent)			11.8.1.1.1.1
		local_BS2x_UDI_T			
52		[ur = C_9600bs]			
53		(TCV_BcapMT1 := Bcap_MT(C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, C_ir_16kbs, TSPX_TE_parity, C_transparent, C_modemt_none), TCV_itc2 := C_UDI, TCV_ce := C_transparent)			11.8.1.1.1.3
54		[ur <> C_9600bs]			
55		(TCV_BcapMT1 := Bcap_MT(C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, C_I440_450,			11.8.1.1.1.3

		C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, C_ir_8kbs, TSPX_TE_parity, C_transparent, C_modemt_none), TCV_itc2 := C_UDI, TCV_ce := C_transparent)			
		local_IFsetup			
56		[TCV_supported]			
57		(TCV_Null := OO_IFsetup_BS2x_MT(TCV_srv, TCV_itc))			
58		[NOT TCV_supported]			
59		[TCV_SpecialCase]			
60		START T_dly(C_T_Wait)			
61		?TIMEOUT T_dly			
62	L_0946	[NOT TCV_SpecialCase]		I	Parameter error

Detailed Comments: The algorithm for derivation of BS2x Bearer Capabilities :

The Bearer Capability for 1st SETUP (used for an MT call):
itc= TSPX_BS_2x_itc1
IF itc=3.1kHz
 IF TSPX_BS_2x_T_MT OR (NOT TSPX_BS_2x_T_MT AND (TSPX_BS_2x_ce = NT))
  use coding of sec. 11.8.1.1.1.2 with (ce=NT)
 ELSE IF NOT TSPX_BS_2x_T_MT AND (TSPX_BS_2x_ce = T)
  use coding of sec. 11.8.1.1.1.1 with (ce=T)
ELSE IF itc=udi
 IF TSPX_BS_2x_T_MT OR (NOT TSPX_BS_2x_T_MT AND (TSPX_BS_2x_ce = NT))
  use coding of sec. 11.8.1.1.1.4 with (ce=NT)
 ELSE IF NOT TSPX_BS_2x_T_MT AND (TSPX_BS_2x_ce = T)
  use coding of sec. 11.8.1.1.1.3 with (ce=T)

The derived 2nd Bearer Capability for 2nd SETUP: (used for TC_11_1_1)
 IF TSPX_BS_2x_more_itc
  itc=TSPX_BS_2x_itc2
 ELSE
  itc= TSPX_BS_2x_itc1
  IF TSPX_BS_2x_T_NT
   IF itc=3.1kHz use coding of sec. 11.8.1.1.1.1 with (ce=T)
   IF itc=udi use coding of sec. 11.8.1.1.1.3 with (ce=T)
  ELSE
   IF itc=3.1Khz use coding of sec. 11.8.1.1.1.2 with (ce=both/T)
   IF itc=udi use coding of sec. 11.8.1.1.1.4 with (ce=both/T)

Test Step Dynamic Behaviour

Test Step Name: BS3xMO(srv:SERVICES; both:BOOLEAN; itc1, sacp1:B_3; ur:B_4; ce:B_2; modem:B_5)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To generate an MO SETUP message with appropriate Bcap and LLC for BS3x service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_srv := srv, TCV_ur := ur, TCV_CE := C_BothT, TCV_ir := C_ir_16kbs, TCV_uil1p := C_no_rate_adaption)			
2		+local_pixitvaluechk			2.
3		+local_selectCe			select a CE
4		+local_bcap			generate BC
5		+local_setupmsg			3.
6		+Chmod(TCV_ChRate, TCV_ce, ur, srv)			initiate channel mode variables
7		(TCV_Null := OO_IFsetup_BS2xorBS3x_MO(TCV_srv, itc1, sacp1, TCV_ce, TCV_ChRate))			ask test operator to configure the MS
		local_bcap			
8		[NOT both]			
9		[TCV_ce = C_notransparent]			
10		[itc1 = C_UDI]			
11		[sacp1 = C_X32]			
12		+local_BS3x_UDI_X32_NT			
13	L_0947	[sacp1 <> C_X32]		I	5.
14		[itc1 = C_3100Hz]			
15	L_0948	[sacp1 <> C_X32]		I	6.
16		[sacp1 = C_X32]			
17		+local_BS3x_3100_X32_NT(C_notransparent)			
18		[TCV_ce = C_transparent]			CE:T only MS
19		[itc1 = C_UDI]			
20	L_0949	[sacp1 = C_X32]		I	7.
21		[sacp1<> C_X32]			
22		+local_BS3x_UDI_T(sacp1)			
23		[itc1 = C_3100Hz]			
24	L_0950	[sacp1 = C_X21]		I	8.
25		[sacp1 = C_I440_450]			
26		+local_BS3x_3100_T			
27		[sacp1 = C_X32]			
28		+local_BS3x_3100_X32_T			
29		[both]			
30		[TCV_ce = C_transparent]			
31		[itc1 = C_UDI]			
32	L_0951	[sacp1 = C_X32]		I	7.
33		[sacp1<> C_X32]			
34		+local_BS3x_UDI_T(sacp1)			
35		[itc1 = C_3100Hz]			3.1kHz
36	L_0952	[sacp1 = C_X21]		I	8.
37		[sacp1 = C_I440_450]			
38		+local_BS3x_3100_T			
39		[sacp1 = C_X32]			
40		(TCV_bcapListIndicator := TCV_bcapListIndicator + 1)			
41		+local_BS3x_3100_X32_NT(TCV_CE)			
42		[(TCV_ce = C_notransparent)]			
43		[itc1 = C_UDI]			
44		[sacp1 = C_X32]			
45		+local_BS3x_UDI_X32_NT			

46	L_0953	[sacp1 <> C_X32]		I	5.
47		[itc1 = C_3100Hz]			3.1kHz
48	L_0954	[sacp1 <> C_X32]		I	6.
49		[sacp1 = C_X32]			
50		(TCV_bcapListIndicator := TCV_bcapListIndicator + 2)			
51		+local_BS3x_3100_X32_Tboth(TCV_CE)			
		local_setupmsg			
52		[itc1 = C_UDI]			
53		(TCV_Setup_mo := Setup_10_UDI(TCV_BcapMO1, Llcmp_BS3x_UDI, Hlcmp_AnyOrOmit))			
54		[itc1 = C_3100Hz]			
55		(TCV_Setup_mo := Setup_10_3100(TCV_BcapMO1, Llcmp_BS3xOrBS61sOrBS81s_3100, Hlcmp_AnyOrOmit))			
		local_BS3x_3100_T			
56		[ur = C_9600bs]			
57		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, C_transparent, modem))			11.8.2.2.1.1
58		[ur <> C_9600bs]			
59		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_8kbs, C_parity_NA, C_transparent, modem))			11.8.2.2.1.1
		local_BS3x_UDI_T(sacp:B_3)			
60		[ur = C_9600bs]			
61		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, sacp, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, C_transparent, C_modemt_none), TCV_uil1p := C_rate_adaption_V110)			11.8.2.2.1.2
62		[NOT (ur = C_9600bs)]			
63		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, sacp, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_8kbs, C_parity_NA, C_transparent, C_modemt_none), TCV_ir := C_ir_8kbs, TCV_uil1p := C_rate_adaption_V110)			11.8.2.2.1.2
		local_BS3x_3100_X32_T			
64		[ur = C_9600bs]			
65		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, C_transparent, modem))			11.8.2.2.1.3
66		[ur <> C_9600bs]			
67		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_8kbs, C_parity_NA, C_transparent, modem))			11.8.2.2.1.3
		local_BS3x_3100_X32_NT(ce2:B_2)			
68		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, C_SDUintegrity, C_nirr_dontcare, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur,			11.8.2.2.1.4

		C_ir_16kbs, C_parity_NA, ce2, modem, C_X25_flct), TCV_bcapListIndicator))			
		local_BS3x_UDI_X32_NT			
69		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_UDI, C_SDUintegrity, C_nirr_dontcare, C_rate_adaption_X31, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, C_notransparent, C_modemt_none, C_X25_flct), TCV_bcapListIndicator), TCV_uil1p := C_rate_adaption_X31)			11.8.2.2.1.5
		local_BS3x_3100_X32_Tboth(ce2:B_2)			
70		[ur = C_9600bs]			
71		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, C_struc_dontchk, C_nirr_nomeaning, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, ce2, modem, C_X25_flct), TCV_bcapListIndicator))			11.8.2.2.1.3 combined with 11.8.2.2.1.4, CE and structure without exact match
72		[ur <> C_9600bs]			
73		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, C_struc_dontchk, C_nirr_dontcare, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_8or16kbs, C_parity_NA, ce2, modem, C_X25_flct), TCV_bcapListIndicator))			11.8.2.2.1.3 combined with 11.8.2.2.1.4, CE, IR and structure without exact match
		local_selectCe			
74		[both]			
75		[(ur = C_300bs) OR (ur = C_120075bs) OR (ur = C_4800bs)]			
76		(TCV_ce := C_transparent)			
77		[(ur = C_1200bs) OR (ur = C_2400bs) OR (ur = C_9600bs)]			
78		(TCV_ce := C_notransparent)			
79		[NOT both]			
80		(TCV_ce := ce)			
		local_pixitvaluechk			
81	L_0955	[(ce <> C_transparent) AND(ce <> C_notransparent) AND(NOT both)]		I	IXIT error
82	L_0956	[(itc1 <> C_UDI) AND(itc1 <> C_3100Hz)]		I	IXIT error
83	L_0957	[(sacp1 <> C_I440_450) AND(sacp1 <> C_X21) AND(sacp1 <>C_X32)]		I	IXIT error
84	L_0958	[(ur = C_1200bs) AND (sacp1 = C_X32)]		I	4.
85		[C_Yes]			IXIT values OK, but combination not checked yet

Detailed Comments:

1. The BS3x Bearer Capabilities and SETUP messages for MO is constructed according to section
   11.8.2.2.1.1, section 11.8.2.2.1.2, section 11.8.2.2.1.3, section 11.8.2.2.1.4 and section 11.8.2.2.1.5 of GSM 11.10 and section B.1.3.1.1, section B.1.3.1.2, section B.1.3.2.1, section B.1.3.2.2, section B.2.3.1 and section B.2.3.2 of GSM 07.01.
2. validate IXIT values.
3. generate SETUP.
4. IXIT error, BS31 does not support X32.
5. IXIT error, UDI non-X32 case has no non transparent mode.
6. IXIT error, 3.1kHz non-X32 case has no non transparent mode.
7. IXIT error, UDI X32 case has no transparent mode.
8. IXIT error, 3.1kHz case has no X21 signalling access protocol.

The algorithm for selecting parameters:

```
IF (TSPX_BS_3x_T_NT)
        TCV_ce := C_notransparent
ELSE
```

```
TCV_ce := TSPX_BS_3x_ce

use (TCV_ce, TSPX_BS_3x_itc1, TSPX_BS_3x_sacp1, svc) for OO_IFsetup_BS2xorBS3x_MO
```

Test Step Dynamic Behaviour					
Test Step Name:		BS3xMT(srv:SERVICES; more_itc, both, more_sacp:BOOLEAN; itc1, itc2, sacp1, sacp2:B_3; ur:B_4; ce:B_2)			
Group:		GSM_L3_MS_v4220/Preambles/			
Objective:		To generate an MT SETUP message with appropriate Bcap for BS3x service.			
Default:		OtherEvents			
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+local_main			
2		+local_IFsetup			
		local_main			
3		(TCV_srv := srv, TCV_ur := ur, TCV_ur2 := ur, TCV_ce := C_transparent)			
4		+local_bcap1			
5		(TCV_BcapMT1 := TCV_BcapMT2)			
6		[more_itc OR more_sacp OR (itc1 = C_3100Hz) AND (sacp1 = C_X32)]			
7		[more_itc]			
8		+local_bcap2			
9		(TCV_Setup_mt1 := Setup_20(TCV_BcapMT2, TI_02), TCV_2ndtest := C_Yes)			2nd SETUP
10		[NOT more_itc]			
11		[itc1 = C_3100Hz]			
12		(TCV_itc2 := C_3100Hz)			
13		[more_sacp]			
14		+local_3100_1(sacp2)			
15		(TCV_Setup_mt1 := Setup_20(TCV_BcapMT2, TI_02), TCV_2ndtest := C_Yes, TCV_sacp2 := sacp2)			2nd SETUP
16		[NOT more_sacp]			
17		[sacp1 = C_I440_450]			no second BC
18		[sacp1 = C_X32]			
19		+local_3100_1(sacp1)			
20		(TCV_Setup_mt1 := Setup_20(TCV_BcapMT2, TI_02), TCV_2ndtest := C_Yes, TCV_sacp2 := sacp1)			2nd SETUP
21	L_0959	[(sacp1 <> C_I440_450) AND(sacp1 <> C_X32)]		I	
22		[itc1 = C_UDI]			
23		[more_sacp]			
24		+local_UDI(sacp2)			
25		(TCV_Setup_mt1 := Setup_20(TCV_BcapMT2, TI_02), TCV_2ndtest := C_Yes, TCV_sacp2 := sacp2, TCV_itc2 := C_UDI)			2nd SETUP
26		[NOT more_sacp]			no second BC
27		[(NOT more_itc) AND(NOT more_sacp) AND ((NOT(itc1=C_3100Hz)) OR (NOT(sacp1=C_X32)))]			
		local_bcap1			
28		[itc1 = C_3100Hz]			
29		+local_3100(sacp1)			
30		(TCV_itc := C_3100Hz, TCV_sacp := sacp1)			
31		(TCV_Setup_mt := Setup_20(TCV_BcapMT2, TI_02), TCV_sacp := sacp1)			1st SETUP
32		+Chmod(TCV_ChRate, TCV_ce, ur, srv)			
33		[itc1 = C_UDI]			
34		+local_UDI(sacp1)			
35		(TCV_Setup_mt := Setup_20(TCV_BcapMT2, TI_02), TCV_sacp := sacp1, TCV_itc := C_UDI)			
36		[sacp1 = C_X32]			
37		+Chmod(TCV_ChRate, C_notransparent, ur, srv)			

38		[NOT (sacp1 = C_X32)]			
39		+Chmod(TCV_ChRate, TCV_ce, ur, srv)			
40	L_0960	[(itc1 <> C_3100Hz) AND(itc1 <> C_UDI)]		I	
		local_bcap2			
41		[itc2 = C_3100Hz]			
42		[more_sacp]			
43		+local_3100_1(sacp2)			
44		(TCV_itc2 := C_3100Hz, TCV_sacp2 := sacp2)			
45		[NOT more_sacp AND(sacp1 <> C_X21)]			
46		+local_3100_1(sacp1)			
47		(TCV_itc2 := C_3100Hz, TCV_sacp2 := sacp1)			
48		[NOT more_sacp AND (sacp1 = C_X21)]			no second BC
49		[itc2 = C_UDI]			
50		[more_sacp]			
51		+local_UDI(sacp2)			
52		(TCV_itc2 := C_UDI, TCV_sacp2 := sacp2)			
53		[NOT more_sacp]			
54		+local_UDI(sacp1)			
55		(TCV_itc2 := C_UDI, TCV_sacp2 := sacp1)			
56	L_0961	[(itc2 <> C_3100Hz) AND(itc2 <> C_UDI)]		I	
		local_3100(sacp:B_3)			
57		[sacp = C_I440_450]			
58		+local_BS3x_3100			
59		[sacp = C_X32]			
60		[NOT (ur= C_1200bs)]			
61		[both OR (ce = C_notransparent)]			
62		+local_BS3x_3100_X32_NT(C_notransparent)			
63		(TCV_ce := C_notransparent)			
64		[NOT both AND (ce = C_transparent)]			
65		+local_BS3x_3100_X32_T			
66	L_0962	[ur= C_1200bs]		I	BS31 not support X32
67	L_0963	[(sacp <> C_I440_450) AND(sacp <> C_X32)]		I	
		local_3100_1(sacp:B_3)			
68		[sacp = C_I440_450]			
69		+local_BS3x_3100			
70		[sacp = C_X32]			
71		+local_BS3x_3100_X32_NT(C_BothT)			
72	L_0964	[(sacp <> C_I440_450) AND(sacp <> C_X32)]		I	
		local_UDI(sacp:B_3)			
73		[sacp = C_I440_450]			
74		[ur = C_9600bs]			
75		+ltree_BS3x_UDI_I440_T(C_ir_16kbs)			
76		[ur <> C_9600bs]			
77		+ltree_BS3x_UDI_I440_T(C_ir_8kbs)			
78		[sacp = C_X21]			
79		[ur = C_9600bs]			
80		+ltree_BS3x_UDI_X21_T(C_ir_16kbs)			
81		[ur <> C_9600bs]			
82		+ltree_BS3x_UDI_X21_T(C_ir_8kbs)			
83		[sacp = C_X32]			
84		[ur <> C_1200bs]			
85		+ltree_BS3x_UDI_X32_NT			
86	L_0965	[ur = C_1200bs]		I	
87	L_0966	[(sacp <> C_I440_450) AND(sacp <> C_X21) AND(sacp <> C_X32)]		I	

		ltree_BS3x_3100_T(ir:B_2; modemt:B_5)			
88		(TCV_BcapMT2 := Bcap_MT(C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, ir, C_parity_NA, C_transparent, modemt))			11.8.1.2.1.1
		ltree_BS3x_UDI_I440_T(ir:B_2)			
89		(TCV_BcapMT2 := Bcap_MT(C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, C_I440_450, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, ir, C_parity_NA, C_transparent, C_modemt_none))			11.8.1.2.1.2
		ltree_BS3x_UDI_X21_T(ir:B_2)			
90		(TCV_BcapMT2 := Bcap_MT(C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, C_X21, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, ir, C_parity_NA, C_transparent, C_modemt_none))			11.8.1.2.1.2
		ltree_BS3x_3100_X32_T(ir:B_2; modemt:B_5)			
91		(TCV_BcapMT2 := Bcap_MT(C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, ir, C_parity_NA, C_transparent, modemt))			11.8.1.2.1.3
		ltree_BS3x_3100_X32_NT(modemt:B_5; ce1:B_2)			
92		(TCV_BcapMT2 := BcapX_MT(C_3100Hz, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, ce1, modemt, C_X25_flct))			11.8.1.2.1.4
		ltree_BS3x_UDI_X32_NT			
93		(TCV_BcapMT2 := BcapX_MT(C_UDI, C_SDUintegrity, C_nirr_nomeaning, C_rate_adaption_X31, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, C_notransparent, C_modemt_none, C_X25_flct))			11.8.1.2.1.5
		local_BS3x_3100			
94		[ur = C_1200bs]			
95		+ltree_BS3x_3100_T(C_ir_8kbs, C_modemt_V22)			
96		[ur = C_2400bs]			
97		+ltree_BS3x_3100_T(C_ir_8kbs, C_modemt_V22bis)			
98		[ur = C_4800bs]			
99		+ltree_BS3x_3100_T(C_ir_8kbs, C_modemt_V32)			
100		[ur = C_9600bs]			
101		+ltree_BS3x_3100_T(C_ir_16kbs, C_modemt_V32)			
102	L_0967	[(ur <> C_1200bs) AND(ur <> C_2400bs) AND(ur <> C_4800bs) AND(ur <> C_9600bs)]		I	
		local_BS3x_3100_X32_NT(ce2:B_2)			
103		[ur = C_2400bs]			
104		+ltree_BS3x_3100_X32_NT(C_modemt_V22bis, ce2)			
105		[(ur = C_4800bs) OR (ur = C_9600bs)]			
106		+ltree_BS3x_3100_X32_NT(C_modemt_V32, ce2)			
107	L_0968	[(ur <> C_2400bs) AND(ur <> C_4800bs) AND(ur <> C_9600bs)]		I	
		local_BS3x_3100_X32_T			
108		[ur = C_2400bs]			
109		+ltree_BS3x_3100_X32_T(C_ir_8kbs, C_modemt_V22bis)			

110		[ur = C_4800bs]			
111		+ltree_BS3x_3100_X32_T(C_ir_8kbs, C_modemt_V32)			
112		[ur = C_9600bs]			
113		+ltree_BS3x_3100_X32_T(C_ir_16kbs, C_modemt_V32)			
114	L_0969	[(ur <> C_2400bs) AND(ur <> C_4800bs) AND(ur <> C_9600bs)]		I	
		local_IFsetup			
115		[TCV_supported]			
116		(TCV_Null := OO_IFsetup_BS3x_MT(TCV_srv, TCV_itc, TCV_sacp))			
117		[NOT TCV_supported]			
118		[TCV_SpecialCase]			
119		START T_dly(C_T_Wait)			
120		?TIMEOUT T_dly			
121	L_0970	[NOT TCV_SpecialCase]		I	Parameter error

Detailed Comments: The algorithm for derivation of BS3x Bearer Capabilities :

The Bearer Capability for 1st SETUP (used for an MT call):
itc= TSPX_BS_3x_itc1
sacp= TSPX_BS_3x_sacp1
IF itc= 3.1Khz
IF sacp= I440 use coding of sec. 11.8.1.2.1.1 with (ce=T)
IF sacp= X32
IF TSPX_BS_3x_T_NT OR (TSPX_BS_3x_ce = NT)
use coding of sec. 11.8.1.2.1.4 with (ce= NT)
IF NOT TSPX_BS_3x_T_NT AND (TSPX_BS_3x_ce = T)
use coding of sec. 11.8.1.2.1.3 with (ce= T)
IF itc=udi
IF sacp= I440 use coding of sec. 11.8.1.2.1.2 with (ce=T)
IF sacp= X21 use coding of sec. 11.8.1.2.1.2 with (ce=T)
IF sacp= X32 use coding of sec. 11.8.1.2.1.5 with (ce=NT)

The derived 2nd Bearer Capability for 2nd SETUP (used for TC_11_1_ only):
Note: the 2nd BC is derived only IF
TSPX_BS_3x_more_itc OR TSPX_BS_3x_more_sacp OR
((TSPX_BS_3x_itc1 = 3.1kHZ) AND (TSPX_BS_3x_scap1 = X32))
IF TSPX_BS_3x_more_itc
itc= TSPX_BS_3x_itc2
IF itc= 3.1kHz
IF TSP_BS_3x_more_sacp
sacp= TSPX_BS_3x_sacp2
ELSE
sacp = TSPX_BS_3x_sacp1
IF sacp= I440 use coding of sec. 11.8.1.2.1.1 with (ce=T)
IF sacp= X32 use coding of sec. 11.8.1.2.1.4 with (ce=both/T)
IF itc= udi
IF TSP_BS_3x_more_sacp
sacp= TSPX_BS_3x_sacp2
IF sacp= I440 use coding of sec. 11.8.1.2.1.2 with (ce=T)
IF sacp= X21 use coding of sec. 11.8.1.2.1.2 with (ce=T)
IF sacp= X32 use coding of sec. 11.8.1.2.1.5 with (ce=NT)
ELSE
sacp= TSPX_BS_3x_sacp1
IF sacp=I440 use coding of sec. 11.8.1.2.1.2 with (ce=T)
IF sacp= X32 use coding of sec. 11.8.1.2.1.5 with (ce=NT)
ELSE
itc= TSPX_BS_3x_itc1
IF itc= 3.1kHz
IF TSPX_BS_3x_more_sacp
sacp= TSPX_BS_3x_sacp2
IF sacp= I440

```
                    use coding of sec. 11.8.1.2.1.1 with (ce=T)
                IF sacp= X32
                    use coding of sec. 11.8.1.2.1.4 with (ce=both/T)
              ELSE
                sacp= TSPX_BS_3x_sacp1
              IF sacp= I440 no second Bearer Capability
              IF sacp= X32 use coding of sec. 11.8.1.2.1.4 with (ce=both/T)
          IF itc= udi
            IF TSP_BS_3x_more_sacp
              sacp= TSPX_BS_3x_sacp2
                IF sacp= I440 use coding of sec. 11.8.1.2.1.2 with (ce=T)
                IF sacp= X21 use coding of sec. 11.8.1.2.1.2 with (ce=T)
                IF sacp= X32 use coding of sec. 11.8.1.2.1.5 with (ce=NT)
              ELSE
                sacp= TSPX_BS_3x_sacp1
                    IF sacp=I440 no second Bearer Capability
                    IF sacp=X21  no second Bearer Capability
                    IF sacp= X32 no second Bearer Capability
```

Test Step Dynamic Behaviour

Test Step Name: BS4xMO(srv:SERVICES; ur:B_4; ce:B_2; both:BOOLEAN)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To generate an MO SETUP message with appropriate Bcap and LLC for BS4x service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_srv := srv, TCV_ur := ur, TCV_CE := C_BothT, TCV_ir := C_ir_16kbs)			
2		+local_pixitvaluechk			validate IXIT values
3		+local_selectCe			select a CE
4		+local_bcap			generate BC
5		(TCV_Setup_mo := Setup_10_UDI(TCV_BcapMO1, Llcmp_BS2xOrBS4x_UDI, Hlcmp_AnyOrOmit))			generate SETUP
6		+Chmod(TCV_ChRate, TCV_ce, ur, srv)			initiate channel mode variables
7		(TCV_Null := OO_IFsetup_BS4x_MO(TCV_srv, TCV_ce, TCV_ChRate))			ask test operator to set up the MS
		local_bcap			
8		[NOT both]			
9		[TCV_ce = C_notransparent]			
10		+local_BS4x_UDI_NT(C_notransparent)			
11		[TCV_ce = C_transparent]			CE:T only MS
12		[ur = C_9600bs]			
13		+local_BS4x_UDI_T(C_ir_16kbs)			
14		[(ur = C_300bs) OR (ur = C_1200bs) OR (ur = C_120075bs) OR (ur = C_2400bs) OR (ur = C_4800bs)]			
15		+local_BS4x_UDI_T(C_ir_8kbs)			
16		[both]			
17		[TCV_ce = C_transparent]			
18		(TCV_bcapListIndicator := TCV_bcapListIndicator + 1)			
19		+local_BS4x_UDI_Tboth(TCV_CE)			
20		[(TCV_ce = C_notransparent)]			
21		(TCV_bcapListIndicator := TCV_bcapListIndicator + 2)			
22		+local_BS4x_UDI_NT(TCV_CE)			
		local_BS4x_UDI_T(ir:B_2)			
23		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, C_X28_unui, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, ir, C_parity_dontcare, C_transparent, C_modemt_none), TCV_ir := ir)			11.8.2.3.1.1
		local_BS4x_UDI_NT(ce1:B_2)			
24		[(ur = C_300bs) OR (ur = C_1200bs) OR (ur = C_120075bs) OR (ur = C_2400bs) OR (ur = C_4800bs)]			
25		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_UDI, C_SDUintegrity, C_nirr_dontcare, C_rate_adaption_V110, C_X28_unui, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, ce1, C_modemt_none, C_uil2p_dontcare), TCV_bcapListIndicator))			11.8.2.3.1.2
26		[ur = C_9600bs]			
27		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_UDI, C_SDUintegrity, C_nirr_nomeaning, C_rate_adaption_V110, C_X28_unui, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, ce1, C_modemt_none,			11.8.2.3.1.2

		C_uil2p_dontcare), TCV_bcapListIndicator))			
		local_BS4x_UDI_Tboth(ce1:B_2)			
28		[(ur = C_300bs) OR (ur = C_1200bs) OR (ur = C_120075bs) OR (ur = C_2400bs) OR (ur = C_4800bs)]			
29		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_UDI, C_struc_dontchk, C_nirr_dontcare, C_rate_adaption_V110, C_X28_unui, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8or16kbs, C_parity_dontcare, ce1, C_modemt_none, C_uil2p_dontcare), TCV_bcapListIndicator))			11.8.2.3.1.2 conbined with 11.8.2.3.1.1, CE, IR and structure without exact match
30		[ur = C_9600bs]			
31		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_UDI, C_struc_dontchk, C_nirr_nomeaning, C_rate_adaption_V110, C_X28_unui, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8or16kbs, C_parity_dontcare, ce1, C_modemt_none, C_uil2p_dontcare), TCV_bcapListIndicator))			11.8.2.3.1.2 conbined with 11.8.2.3.1.1, CE, IR and structure without exact match
		local_pixitvaluechk			
32	L_0971	[(ce <> C_transparent) AND(ce <> C_notransparent) AND(NOT both)]		I	IXIT error
33		[C_Yes]			IXIT values OK
		local_selectCe			
34		[both]			
35		[(ur = C_300bs) OR (ur = C_120075bs) OR (ur = C_4800bs)]			
36		(TCV_ce := C_notransparent)			
37		[(ur = C_1200bs) OR (ur = C_2400bs) OR (ur = C_9600bs)]			
38		(TCV_ce := C_transparent)			
39		[NOT both]			
40		(TCV_ce := ce)			

Detailed Comments:

The BS4x Bearer Capabilities and SETUP messages for MO is constructed according to section 11.8.2.3.1.1 and section 11.8.2.3.1.2 of GSM 11.10 and section B.1.4 and section B.2.2.1 of GSM 07.01.

The algorithm for selecting parameters:

```
IF (TSPX_BS_4x_T_NT)
        IF (user rate is one of the C_300bs, C_120075bs, C_4800bs)
                TCV_ce := C_notransparent
        ELSE   /* user rate is one of C_1200bs, C_2400bs, C_9600bs \
```

Test Step Dynamic Behaviour

Test Step Name: BS5xMO(srv:SERVICES; ur:B_4)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To generate an MO SETUP message with appropriate Bcap for BS5x service.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_srv := srv, TCV_ur := ur)			
2		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_UDI, C_SDUintegrity, C_nirr_nomeaning, C_rate_adaption_X31, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, C_notransparent, C_modemt_none, C_X25_flct))			11.8.2.4.1
3		(TCV_Setup_mo := Setup_10_UDI(TCV_BcapMO1, Llcmp_BS5x, Hlcmp_AnyOrOmit))			
4		+Chmod(TCV_ChRate, C_notransparent, ur, srv)			channel mode
5		(TCV_Null := OO_IFsetup_BS5x_MO(TCV_srv, TCV_ur, TCV_ChRate))			

Detailed Comments:

1. Bearer Caoability is coded according to section 11.8.2.4.1 of GSM 11.10 ver 4.16
2. low layer compatibility is coded according to section B.2.5 of GSM 07.01 ver 4.90

Test Step Dynamic Behaviour

Test Step Name: BS61or81MO(srv:SERVICES; async, both: BOOLEAN; a_ur1: B_4; ce:B_2; sync:BOOLEAN; s_ur1: B_4; ri:B_8)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To get a MO SETUP message with right BC IE and LLC for BS61 or BS81 service.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_srv := srv, TCV_CE := C_BothT, TCV_ir := C_ir_8kbs, TCV_MODEM := C_modemt_none)			
2		+local_pixitvaluechk			1.
3		+local_selectSaAndCe			2.
4		+local_bcap			generate BC
5		+local_setupmsg			generate SETUP
6		+Chmod(TCV_ChRate, TCV_ce, TCV_ur, srv)			
7		(TCV_Null := OO_IFsetup_BS61orBS81_MO(TCV_srv, TCV_ur, TCV_sa, TCV_ce, TCV_ChRate))			
		local_bcap			
8		[TCV_sa = C_Synchronous]			
9		[TCV_ur = C_1200bs]			
10		+local_BS61or81_S_T(C_ir_8kbs, C_1200bs, C_modemt_V22)			
11		[TCV_ur = C_2400bs]			
12		(TCV_bcapListIndicator := C_modemList_ce)			
13		+local_BS61or81_S_T(C_ir_8kbs, C_2400bs, TCV_MODEM)			
14		[TCV_ur = C_4800bs]			
15		+local_BS61or81_S_T(C_ir_8kbs, C_4800bs, C_modemt_V32)			
16		[TCV_ur = C_9600bs]			
17		+local_BS61or81_S_T(C_ir_16kbs, C_9600bs, C_modemt_V32)			
18		(TCV_ir := C_ir_16kbs)			
19		[TCV_sa = C_Asynchronous]			
20		[(TCV_ce = C_transparent) AND(NOT both)]			CE:T only MS
21		+local_BS61or81_A_T(TCV_ur)			
22		[(TCV_ce = C_notransparent) AND(NOT both)]			CE:NT only MS
23		+local_BS61or81_A_NT(TCV_ur, C_notransparent, C_SDUintegrity, C_nirr_dontcare, C_ir_16kbs)			
24		[(TCV_ce = C_notransparent) AND both]			
25		(TCV_bcapListIndicator := TCV_bcapListIndicator + 2)			
26		+local_BS61or81_A_NT(TCV_ur, TCV_CE, C_SDUintegrity, C_nirr_dontcare, C_ir_16kbs)			
27		[(TCV_ce = C_transparent) AND both]			
28		(TCV_bcapListIndicator := TCV_bcapListIndicator + 1)			
29		+local_BS61or81_A_NT(TCV_ur, TCV_CE, C_struc_dontchk, C_nirr_dontcare, C_ir_8or16kbs)			
		local_BS61or81_A_T(ur:B_4)			
30		[ur = C_300bs]			
31		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent, C_modemt_V21), TCV_modemt := C_modemt_V21)			11.8.2.5.1
32		[ur = C_1200bs]			
33		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning,			11.8.2.5.1

		C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent, C_modemt_V22), TCV_modemt := C_modemt_V22)			
34		[ur = C_120075bs]			
35		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent, C_modemt_V23), TCV_modemt := C_modemt_V23)			11.8.2.5.1
36		[ur = C_2400bs]			
37		(TCV_bcapListIndicator := C_modemList_ce, TCV_BcapMO1 := OC_ModifyBcap(Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent, TCV_MODEM), TCV_bcapListIndicator), TCV_modemt := TCV_MODEM)			11.8.2.5.1
38		[ur = C_4800bs]			
39		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent, C_modemt_V32), TCV_modemt := C_modemt_V32)			11.8.2.5.1
40		[ur = C_9600bs]			
41		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, C_transparent, C_modemt_V32), TCV_modemt := C_modemt_V32)			11.8.2.5.1
42		(TCV_ir := C_ir_16kbs)			
		local_BS61or81_A_NT(ur:B_4; ce:B_2; struc:B_2; nirr :B_1; ir:B_2)			
43		(TCV_ir := C_ir_16kbs)			
44		[ur = C_300bs]			
45		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, struc, nirr, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, ir, C_parity_dontcare, ce, C_modemt_V21, C_uil2p_dontcare), TCV_bcapListIndicator), TCV_modemt := C_modemt_V21)			11.8.2.5.2
46		[ur = C_1200bs]			
47		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, struc, nirr, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, ir, C_parity_dontcare, ce, C_modemt_V22, C_uil2p_dontcare), TCV_bcapListIndicator), TCV_modemt := C_modemt_V22)			11.8.2.5.2
48		[ur = C_120075bs]			
49		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, struc, nirr, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, ir, C_parity_dontcare, ce, C_modemt_V23, C_uil2p_dontcare), TCV_bcapListIndicator),			11.8.2.5.2

		TCV_modemt := C_modemt_V23)			
50		[ur = C_2400bs]			
51		+local_BumpListIndicator			
52		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, struc, nirr, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, ir, C_parity_dontcare, ce, TCV_MODEM, C_uil2p_dontcare), TCV_bcapListIndicator), TCV_modemt := TCV_MODEM)			11.8.2.5.2
53		[ur = C_4800bs]			
54		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, struc, nirr, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, ir, C_parity_dontcare, ce, C_modemt_V32, C_uil2p_dontcare), TCV_bcapListIndicator), TCV_modemt := C_modemt_V32)			11.8.2.5.2
55		[ur = C_9600bs]			
56		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, struc, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, ce, C_modemt_V32, C_uil2p_dontcare), TCV_bcapListIndicator), TCV_modemt := C_modemt_V32)			11.8.2.5.2
		local_BumpListIndicator			
57		[TCV_bcapListIndicator < C_modemList_ce]			
58		(TCV_bcapListIndicator := TCV_bcapListIndicator + 3)			
59		[TCV_bcapListIndicator >= C_modemList_ce]			
		local_BS61or81_S_T(ir:B_2; ur: B_4; modemt:B_5)			
60		(TCV_BcapMO1 := OC_ModifyBcap(Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, ir, C_parity_NA, C_transparent, modemt), TCV_bcapListIndicator), TCV_modemt := modemt)			11.8.2.5.3
		local_pixitvaluechk			
61	L_0972	[(ce <> C_transparent) AND(ce <> C_notransparent) AND(NOT both) AND async]		I	IXIT error
62	L_0973	[(a_ur1 <> C_300bs) AND(a_ur1 <> C_1200bs) AND(a_ur1 <> C_2400bs) AND(a_ur1 <> C_4800bs) AND(a_ur1 <> C_9600bs) AND(a_ur1 <> C_120075bs) AND async]		I	IXIT error
63	L_0974	[(s_ur1 <> C_1200bs) AND(s_ur1 <> C_2400bs) AND(s_ur1 <> C_4800bs) AND(s_ur1 <> C_9600bs) AND sync]		I	IXIT error
64		[C_Yes]			IXIT values OK
		local_selectSaAndCe			
65		[async]			3.
66		(TCV_sa := C_Asynchronous, TCV_ur := a_ur1)			
67		[both]			
68		[(a_ur1 = C_300bs) OR (a_ur1 = C_120075bs) OR (a_ur1 = C_4800bs)]			
69		(TCV_ce := C_notransparent)			
70		[(a_ur1 = C_1200bs) OR (a_ur1 = C_2400bs) OR (a_ur1 = C_9600bs)]			
71		(TCV_ce := C_transparent)			
72		[NOT both]			
73		(TCV_ce := ce)			

74		[sync]			
75		(TCV_sa := C_Synchronous, TCV_ce := C_transparent, TCV_ur := s_ur1)			
		local_setupmsg			
76		[TCV_sa = C_Asynchronous]			
77		[(NOT TSPC_EFR_Speech_v2) AND(NOT TSPC_EFR_Speech_v3)]			
78		(TCV_Setup_mo := Setup_11_3100(ri, Bcap_Speech_MO(C_Rchr_dontcare), TCV_BcapMO1, Llcmp_NotApplicable, Llcmp_BS2xOrBS61aOrBS81a_3100, ri, ri, Hlcmp_NotApplicable, Hlcmp_AnyOrOmit))			Non EFR Mobile
79		[TSPC_EFR_Speech_v2 OR TSPC_EFR_Speech_v3]			
80		(TCV_Setup_mo := Setup_11_3100(ri, Bcap_Speech_Efr_MO(C_Rchr_dontcare), TCV_BcapMO1, Llcmp_NotApplicable, Llcmp_BS2xOrBS61aOrBS81a_3100, ri, ri, Hlcmp_NotApplicable, Hlcmp_AnyOrOmit))			EFR Mobile
81		[TCV_sa = C_Synchronous]			
82		[(NOT TSPC_EFR_Speech_v2) AND(NOT TSPC_EFR_Speech_v3)]			
83		(TCV_Setup_mo := Setup_11_3100(ri, Bcap_Speech_MO(C_Rchr_dontcare), TCV_BcapMO1, Llcmp_NotApplicable, Llcmp_BS3xOrBS61sOrBS81s_3100, ri, ri, Hlcmp_NotApplicable, Hlcmp_AnyOrOmit))			Non EFR Mobile
84		[TSPC_EFR_Speech_v2 OR TSPC_EFR_Speech_v3]			
85		(TCV_Setup_mo := Setup_11_3100(ri, Bcap_Speech_Efr_MO(C_Rchr_dontcare), TCV_BcapMO1, Llcmp_NotApplicable, Llcmp_BS3xOrBS61sOrBS81s_3100, ri, ri, Hlcmp_NotApplicable, Hlcmp_AnyOrOmit))			EFR Mobile

Detailed Comments:

1. validate IXIT values
2. select sync or async mode and corresponding CE
3. If the MS supports both synchronous and asynchronous mode, selete the asynchronous mode
4. The BS61 or BS81 Bearer Capabilities and SETUP messages for MO is constructed according to section 11.8.2.5.1, section 11.8.2.5.2 and section 11.8.2.5.3 of GSM 11.10 and section B.1.6.1, section B.1.6.2.1, section B.1.6.2.2, section B.2.2.2 and section B.2.3.2 of GSM 07.01.

The algorithm for selecting parameters for BS61 basice service:

```
IF (TSPX_BS_61_A)
        TCV_sa := C_Asynchronous
        IF (TSPX_BS_61_A_ur1_T_NT)
                TCV_ce := C_notransparent
        ELSE
                TCV_ce := TSPX_BS_61_A_ur1_ce
        use (TCV_sa, TCV_ce, TSPX_BS_61_A_ur1, svc) for OO_IFsetup_BS61orBS81_MO
ELSE
        TCV_sa := C_Synchronous
        use (TCV_sa, C_transparent, TSPX_BS_61_S_ur1, svc) for
OO_IFsetup_BS61orBS81_MO
```

The algorithm for selecting parameters for BS61 basice service is the same as above except using corresponding IXIT parameters.

Test Step Dynamic Behaviour

Test Step Name: BS61or81MT(srv:SERVICES; async, t_nt_ur1, t_nt_ur2, a_more_ur, sync, s_more_ur:BOOLEAN; a_ur1, a_ur2, s_ur1, s_ur2: B_4; ri:B_8; ce:B_2)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To get a MT SETUP message with right BC IE for BS61 or BS81 service.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+local_main			
2		+local_IFsetup			
		local_main			
3		(TCV_srv := srv)			
4		(TCV_BcapMT1 := Bcap_Speech_MT)			
5		[async]			If asynchro
6		+local_bcapa			
7		(TCV_Setup_mt := Setup_21(ri, TCV_BcapMT1, TCV_BcapMT2, TI_02), TCV_ur := a_ur1, TCV_sa := C_Asynchronous)			1st SETUP
8		+Chmod(TCV_ChRate, TCV_ce, a_ur1, srv)			2nd ch mode
9		(TCV_ChMod.mode := C_ChMod_r)			Speech mode
10		[sync]			
11		+local_bcap2S(s_ur1)			
12		(TCV_Setup_mt1 := Setup_21(ri, TCV_BcapMT1, TCV_BcapMT2, TI_02), TCV_2ndtest :=C_Yes, TCV_ur2 := s_ur1, TCV_sa2 := C_Synchronous)			2nd SETUP
13		[a_more_ur]			
14		[t_nt_ur2]			
15		+local_bcap2A(a_ur2, C_transparent)			
16		(TCV_Setup_mt1 := Setup_21(ri, TCV_BcapMT1, TCV_BcapMT2, TI_02), TCV_2ndtest :=C_Yes, TCV_ur2 := a_ur2, TCV_sa2 := C_Asynchronous)			2nd SETUP
17		[NOT t_nt_ur2]			
18		+local_bcap2A(a_ur2, C_BothT)			
19		(TCV_Setup_mt1 := Setup_21(ri, TCV_BcapMT1, TCV_BcapMT2, TI_02), TCV_2ndtest :=C_Yes, TCV_ur2 := a_ur2, TCV_sa2 := C_Asynchronous)			2nd SETUP
20		[NOT a_more_ur]			
21		[t_nt_ur1]			
22		+local_bcap2A(a_ur1, C_transparent)			
23		(TCV_Setup_mt1 := Setup_21(ri, TCV_BcapMT1, TCV_BcapMT2, TI_02), TCV_2ndtest := C_Yes, TCV_ur2 := a_ur1, TCV_sa2 := C_Asynchronous)			2nd SETUP
24		[NOT t_nt_ur1]			
25		+local_bcap2A(a_ur1, C_BothT)			
26		(TCV_Setup_mt1 := Setup_21(ri, TCV_BcapMT1, TCV_BcapMT2, TI_02), TCV_2ndtest := C_Yes, TCV_ur2 := a_ur1, TCV_sa2 := C_Asynchronous)			2nd SETUP
27		[sync]			If synchro
28		+local_bcap2S(s_ur1)			
29		(TCV_Setup_mt := Setup_21(ri, TCV_BcapMT1, TCV_BcapMT2, TI_02), TCV_ur := s_ur1, TCV_sa := C_Synchronous)			1st SETUP
30		+Chmod(TCV_ChRate, C_transparent, s_ur1, srv)			2nd ch mode
31		(TCV_ChMod.mode := C_ChMod_r)			Speech mode
32		[s_more_ur]			

33		+local_bcap2S(s_ur2)			
34		(TCV_Setup_mt1 := Setup_21(ri, TCV_BcapMT1, TCV_BcapMT2, TI_02), TCV_2ndtest :=C_Yes, TCV_ur2 := s_ur1, TCV_sa2 := C_Synchronous)			2nd SETUP
35		[NOT s_more_ur]			no second BC
		local_bcap2A(ur:B_4; ce2: B_2)			
36		[ce2 = C_transparent]			
37		[ur = C_300bs]			
38		+local_BS61or81_A_T(C_ir_8kbs, ur, C_modemt_V21)			
39		[ur = C_1200bs]			
40		+local_BS61or81_A_T(C_ir_8kbs, ur, C_modemt_V22)			
41		[ur = C_2400bs]			
42		+local_BS61or81_A_T(C_ir_8kbs, ur, C_modemt_V22bis)			
43		[ur = C_4800bs]			
44		+local_BS61or81_A_T(C_ir_8kbs, ur, C_modemt_V32)			
45		[ur = C_9600bs]			
46		+local_BS61or81_A_T(C_ir_16kbs, ur, C_modemt_V32)			
47	L_0975	[(ur <> C_300bs) AND(ur <> C_1200bs) AND(ur <> C_2400bs) AND(ur <> C_4800bs) AND(ur <> C_9600bs)]		I	
48		[ce2 = C_BothNT]			
49		[ur = C_300bs]			
50		+local_BS61or81_A_NT(ur, C_modemt_V21, C_BothNT)			
51		[ur = C_1200bs]			
52		+local_BS61or81_A_NT(ur, C_modemt_V22, C_BothNT)			
53		[ur = C_2400bs]			
54		+local_BS61or81_A_NT(ur, C_modemt_V22bis, C_BothNT)			
55		[ur = C_4800bs]			
56		+local_BS61or81_A_NT(ur, C_modemt_V32, C_BothNT)			
57		[ur = C_9600bs]			
58		+local_BS61or81_A_NT(ur, C_modemt_V32, C_BothNT)			
59	L_0976	[(ur <> C_300bs) AND(ur <> C_1200bs) AND(ur <> C_2400bs) AND(ur <> C_4800bs) AND(ur <> C_9600bs)]		I	
60		[ce2 = C_notransparent]			
61		[ur = C_300bs]			
62		+local_BS61or81_A_NT(ur, C_modemt_V21, C_notransparent)			
63		[ur = C_1200bs]			
64		+local_BS61or81_A_NT(ur, C_modemt_V22, C_notransparent)			
65		[ur = C_2400bs]			
66		+local_BS61or81_A_NT(ur, C_modemt_V22bis, C_notransparent)			
67		[ur = C_4800bs]			
68		+local_BS61or81_A_NT(ur, C_modemt_V32, C_notransparent)			
69		[ur = C_9600bs]			
70		+local_BS61or81_A_NT(ur, C_modemt_V32, C_notransparent)			
71	L_0977	[(ur <> C_300bs) AND(ur <> C_1200bs) AND(ur <> C_2400bs) AND(ur <> C_4800bs) AND(ur <> C_9600bs)]		I	
72	L_0978	[(ce2 <> C_notransparent) AND(ce2 <> C_BothNT) AND (ce2 = C_transparent)]		I	

		local_bcap2S(ur:B_4)			
73		[ur = C_1200bs]			
74		+local_BS61or81_S_T(C_ir_8kbs, ur, C_modemt_V22)			
75		[ur = C_2400bs]			
76		+local_BS61or81_S_T(C_ir_8kbs, ur, C_modemt_V22bis)			
77		[ur = C_4800bs]			
78		+local_BS61or81_S_T(C_ir_8kbs, ur, C_modemt_V32)			
79		[ur = C_9600bs]			
80		+local_BS61or81_S_T(C_ir_16kbs, ur, C_modemt_V32)			
81	L_0979	[(ur <> C_1200bs) AND(ur <> C_2400bs) AND(ur <> C_4800bs) AND(ur <> C_9600bs)]		I	
		local_BS61or81_A_T(ir:B_2; ur:B_4; modemt:B_5)			
82		[TSPX_TE_FLCT = C_Outband]			
83		(TCV_BcapMT2 := Bcap_MT(C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, ir, TSPX_TE_parity, C_transparent, modemt))			11.8.1.3.1
84		[TSPX_TE_FLCT = C_Inband]			
85		(TCV_BcapMT2 := BcapX_MT(C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, ir, TSPX_TE_parity, C_transparent, modemt, C_ISO6429))			11.8.1.3.1
86		[TSPX_TE_FLCT = C_Nocontrol]			
87		(TCV_BcapMT2 := BcapX_MT(C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, ir, TSPX_TE_parity, C_transparent, modemt, C_COPnoFLCT))			11.8.1.3.1
88	L_0980	[(TSPX_TE_FLCT <> C_Outband) AND(TSPX_TE_FLCT <> C_Inband) AND(TSPX_TE_FLCT <> C_Nocontrol)]		I	
		local_BS61or81_A_NT(ur:B_4; modemt:B_5; cel:B_2)			
89		[TSPX_TE_FLCT = C_Outband]			
90		(TCV_BcapMT2 := Bcap_MT(C_3100Hz, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, C_ir_16kbs, TSPX_TE_parity, cel, modemt))			11.8.1.3.2
91		[TSPX_TE_FLCT = C_Inband]			
92		(TCV_BcapMT2 := BcapX_MT(C_3100Hz, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, C_ir_16kbs, TSPX_TE_parity, cel, modemt, C_ISO6429))			11.8.1.3.2
93		[TSPX_TE_FLCT = C_Nocontrol]			
94		(TCV_BcapMT2 := BcapX_MT(C_3100Hz, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, TSPX_TE_stopbit, TSPX_TE_databit, ur, C_ir_16kbs, TSPX_TE_parity, cel, modemt, C_COPnoFLCT))			11.8.1.3.2
95	L_0981	[(TSPX_TE_FLCT <> C_Outband) AND(TSPX_TE_FLCT <> C_Inband) AND(TSPX_TE_FLCT <> C_Nocontrol)]		I	
		local_BS61or81_S_T(ir:B_2; ur: B_4; modemt:B_5)			

96		(TCV_BcapMT2 := Bcap_MT(C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, ir, C_parity_NA, C_transparent, modemt))			11.8.1.3.3
		local_bcapa			
97		[t_nt_ur1 OR (ce = C_notransparent)]			
98		+local_bcap2A(a_ur1, C_notransparent)			
99		(TCV_ce := C_notransparent)			
100		[NOT t_nt_ur1 AND (ce = C_transparent)]			
101		+local_bcap2A(a_ur1, C_transparent)			
102		(TCV_ce := C_transparent)			
		local_IFsetup			
103		[TCV_supported]			
104		(TCV_Null := OO_IFsetup_BS61orBS81_MT(TCV_srv, TCV_ur, TCV_sa))			
105		[NOT TCV_supported]			
106		[TCV_SpecialCase]			
107		START T_dly(C_T_Wait)			
108		?TIMEOUT T_dly			
109	L_0982	[NOT TCV_SpecialCase]		I	Parameter error

Detailed Comments:

The algorithm for derivation of BS61 Bearer Capabilities:

The data Bearer Capability for 1st SETUP (used for an MT call):

```
IF TSPX_BS_61_A
    ur= TSPX_BS_61_A_ur1
    IF TSPX_BS_61_A_ur1_T_NT OR (TSPX_BS_61_A_ur1_ce = NT)
        use coding of sec. 11.8.1.3.2 with (ur & ce=NT)
    ELSE
        use coding of sec. 11.8.1.3.1 with (ur & ce=T)
ELSE
   IF TSPX_BS_61_S
    ur= TSPX_BS_61_S_ur1 use coding of sec. 11.8.1.3.3 with (ur & ce=T)
```

The 2nd Bearer Capability for second SETUP (used for TC_11_1_1):

Note: the 2nd BC is derived IF (TSPX_BS_61_A OR (TSPX_BS_61_S AND TSPX_BS_61_S_more_ur))

```
IF TSPX_BS_61_A AND TSPX_BS_61_S
    ur= TSPX_BS_61_S_ur1 use coding of sec. 11.8.1.3.3 with (ur & ce= T)
ELSE
    IF TSPX_BS_61_A_more_ur  AND TSPX_BS_61_A
          ur= TSPX_BS_61_A_ur2
        IF TSPX_BS_61_A_ur2_T_NT
            use coding of sec. 11.8.1.3.1 with (ur & ce=T)
        ELSE
            use coding of sec. 11.8.1.3.2 with (ur & ce=both/T)
    ELSE
        IF TSPX_BS_61_S AND TSPX_BS_61_S_more_ur
            ur=TSPX_BS_61_S_ur2
            use coding of sec. 11.8.1.3.3 with (ur & ce= T)
        ELSE
            IF TSPX_BS_61_A
                ur= TSPX_BS_61_A_ ur1
              IF TSPX_BS_61_A_ur1_T_NT
                use coding of sec. 11.8.1.3.1 with (ur & ce=T)
              ELSE
                use coding of sec. 11.8.1.3.2 with (ur & ce=both/T)
```

The algorithm for derivation of BS81 Bearer Capabilities is same as above except all parameters are changed to corresponding BS81's parameters.

Test Step Dynamic Behaviour

Test Step Name: CallCfmGen(svc:SERVICES; rate:RATE)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To generate a CALL COMFIRMED based on the service and rate, as well as on several IXIT parameters, itc, ce, ur, synchronous bit for verifying received CALL COMFIRMED message.
Default: OtherEvents
Comments: Used only for TC_11_2

Nr	Label	Behaviour Description	CRef	V	Comments
1		+local_init			Default values
2		[TCV_Service = C_Telephony]			
3		[(NOT TSPC_EFR_Speech_v2) AND(NOT TSPC_EFR_Speech_v3)]			
4		(TCV_BcapMO1 := Bcap_Speech_MO(TCV_Rchr))			TS11 - Non EFR Mobile
5		+local_callcfm1			
6		[TSPC_EFR_Speech_v2 OR TSPC_EFR_Speech_v3]			
7		(TCV_BcapMO1 := Bcap_Speech_Efr_MO(TCV_Rchr))			TS11 - EFR Mobile
8		+local_callcfm1			
9		[TCV_Service = C_AltSpchFax]			
10		[rate = C_Full]			
11		+local_TS61(C_AltSpchFax)			TS61
12	L_0983	[rate <> C_Full]		I	IXIT error
13		[TCV_Service = C_AutoFax]			
14		[rate = C_Full]			
15		+local_TS62(C_AutoFax)			TS62
16	L_0984	[rate <> C_Full]		I	IXIT error
17		[TCV_Service = C_Async300]			
18		+local_BS2x(C_Async300, C_300bs, C_modemt_V21, TSPX_BS_21_ce)			BS21
19		[TCV_Service = C_Async1200]			
20		+local_BS2x(C_Async1200, C_1200bs, C_modemt_V22, TSPX_BS_22_ce)			BS22
21		[TCV_Service = C_Async120075]			
22		+local_BS2x(C_Async120075, C_120075bs, C_modemt_V23, TSPX_BS_23_ce)			BS23
23		[TCV_Service = C_Async2400]			
24		(TCV_bcapListIndicator := C_modemList_ce)			
25		+local_BS2x(C_Async2400, C_2400bs, TCV_MODEM, TSPX_BS_24_ce)			BS24
26		[TCV_Service = C_Async4800]			
27		+local_BS2x(C_Async4800, C_4800bs, C_modemt_V32, TSPX_BS_25_ce)			BS25
28		[TCV_Service = C_Async9600]			
29		+local_BS2x(C_Async9600, C_9600bs, C_modemt_V32, TSPX_BS_26_ce)			BS26
30		[TCV_Service = C_Sync1200]			
31		+local_BS3x(C_Sync1200, TSPX_BS_31_sacp1, C_1200bs, C_transparent, C_modemt_V22)			BS31
32		[TCV_Service = C_Sync2400]			
33		(TCV_bcapListIndicator := C_modemList_ce)			
34		+local_BS3x(C_Sync2400, TSPX_BS_32_sacp1, C_2400bs, TSPX_BS_32_X32_ce, TCV_MODEM)			BS32
35		[TCV_Service = C_Sync4800]			
36		+local_BS3x(C_Sync4800, TSPX_BS_33_sacp1, C_4800bs, TSPX_BS_33_X32_ce, C_modemt_V32)			BS33
37		[TCV_Service = C_Sync9600]			
38		+local_BS3x(C_Sync9600, TSPX_BS_34_sacp1, C_9600bs, TSPX_BS_34_X32_ce, C_modemt_V32)			BS34
39		[TCV_Service = C_AltSpchData]			
40		+local_BS61or81(C_AltSpchData, TSPX_BS_61_A, TSPX_BS_61_A_ur1,			BS61

		TSPX_BS_61_A_ur1_ce, TSPX_BS_61_S, TSPX_BS_61_S_ur1, C_RI_alternate)			
41		[TCV_Service = C_SpchData]			
42		+local_BS61or81(C_SpchData, TSPX_BS_81_A, TSPX_BS_81_A_ur1, TSPX_BS_81_A_ur1_ce, TSPX_BS_81_S, TSPX_BS_81_S_ur1, C_RI_follow)			BS81
43	L_0985	[C_Yes]		I	IXIT error
		local_init			
44		(TCV_Service := svc, TCV_Rchr := C_Rchr_Full, TCV_MODEM := C_modemt_none, TCV_bcapListIndicator := C_modemt_ce)			
45		[TSPC_DualRate]			
46		(TCV_Rchr := '1?'B)			
47		[NOT TSPC_DualRate]			
		local_BS2x(svc:SERVICES; ur:B_4; modem:B_5; ce:B_2)			
48	L_0986	[(ce <> C_transparent) AND(ce <> C_notransparent)]		I	IXIT value error
49		[C_Yes]			
50		+local_callCfm_BS2x(ce, ur, modem)			generate call comfirmed msg
51		(TCV_Null := OO_IFsetup_BS2xorBS3x_MO(svc, C_3100Hz, C_I440_450, ce, TCV_ChRate))			ask test operator to configure the MS
		local_callCfm_BS2x(ce1: B_2; ur1:B_4; modem1:B_5)			
52		[ce1 = C_transparent]			
53		[ur1 = C_9600bs]			
54		(TCV_BcapMO1 := OC_ModifyBcap(Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur1, C_ir_16kbs, C_parity_dontcare, C_transparent, modem1), TCV_bcapListIndicator))			11.8.2.1.1.1
55		+local_callcfm1			
56		[ur1 <> C_9600bs]			
57		(TCV_BcapMO1 := OC_ModifyBcap(Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur1, C_ir_8kbs, C_parity_dontcare, C_transparent, modem1), TCV_bcapListIndicator))			11.8.2.1.1.1
58		+local_callcfm1			
59		[ce1 = C_notransparent]			
60		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur1, C_ir_16kbs, C_parity_dontcare, C_notransparent, modem1, C_uil2p_dontcare), TCV_bcapListIndicator))			11.8.2.1.1.2
61		+local_callcfm1			
		local_BS3x(svc:SERVICES; sacp:B_3; ur:B_4; ce:B_2; modem:B_5)			
62	L_0987	[((ce <> C_transparent) AND(ce <> C_notransparent)) OR((sacp <> C_I440_450) AND(sacp <> C_X32))]		I	IXIT value error
63		[C_Yes]			
64		+local_callCfm_BS3x(ce, ur, sacp, modem)			generate call comfirmed msg
65		(TCV_Null := OO_IFsetup_BS2xorBS3x_MO(svc,			ask operator to configure

		C_3100Hz, sacp, ce, TCV_ChRate))			the MS
		local_callCfm_BS3x(ce1: B_2; ur1: B_4; sacp1: B_3; modem1 :B_5)			
66		[sacp1 = C_I440_450]			
67		[ur1 = C_9600bs]			
68		(TCV_BcapMO1 := OC_ModifyBcap(Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Synchronous, C_stopbit_NA, C_databit_NA, ur1, C_ir_16kbs, C_parity_NA, C_transparent, modem1), TCV_bcapListIndicator))			11.8.2.2.1.1
69		+local_callcfm1			
70		[ur1 <> C_9600bs]			
71		(TCV_BcapMO1 := OC_ModifyBcap(Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Synchronous, C_stopbit_NA, C_databit_NA, ur1, C_ir_8kbs, C_parity_NA, C_transparent, modem1), TCV_bcapListIndicator))			11.8.2.2.1.1
72		+local_callcfm1			
73		[sacp1 = C_X32]			
74		[ce1 = C_transparent]			
75		[ur1 = C_9600bs]			
76		(TCV_BcapMO1 := OC_ModifyBcap(Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur1, C_ir_16kbs, C_parity_NA, C_transparent, modem1), TCV_bcapListIndicator))			11.8.2.2.1.3
77		+local_callcfm1			
78		[ur1 <> C_9600bs]			
79		(TCV_BcapMO1 := OC_ModifyBcap(Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur1, C_ir_8kbs, C_parity_NA, C_transparent, modem1), TCV_bcapListIndicator))			11.8.2.2.1.3
80		+local_callcfm1			
81		[ce1 = C_notransparent]			
82		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_3100Hz, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur1, C_ir_16kbs, C_parity_NA, ce1, modem1, C_X25_flct), TCV_bcapListIndicator))			11.8.2.2.1.4
83		+local_callcfm1			
		local_BS61or81(svc1:SERVICES; async1:BOOLEAN; ur_a1:B_4; ce_a1:B_2; sync1:BOOLEAN; ur_s1:B_4; ri3: RPI)			
84	L_0988	[(ce_a1 <> C_transparent) AND(ce_a1 <> C_notransparent) AND async1]		I	IXIT error
85	L_0989	[(ur_a1 <> C_300bs) AND(ur_a1 <> C_1200bs) AND(ur_a1 <> C_2400bs) AND(ur_a1 <> C_4800bs) AND(ur_a1 <> C_9600bs) AND(ur_a1 <> C_120075bs) AND async1]		I	IXIT error
86	L_0990	[(ur_s1 <> C_1200bs) AND(ur_s1 <> C_2400bs) AND(ur_s1 <> C_4800bs) AND(ur_s1 <> C_9600bs) AND sync1]		I	IXIT error
87		[C_Yes]			IXIT values OK
88		+local_callCfm_BS61or81(svc1, async1, ur_a1, ce_a1, sync1, ur_s1, ri3)			
89		(TCV_Null := OO_IFsetup_BS61orBS81_MO(svc, TCV_ur, TCV_sa, TCV_ce, TCV_ChRate))			
		local_callCfm_BS61or81(svc:SERVICES; async:BOOLEAN; ur_a:B_4; ce_a:B_2;			

```
                 sync:BOOLEAN; ur_s:B_4; ri:RPI)
90               [async]
91                 [ce_a = C_transparent]
92                   (TCV_ur := ur_a, TCV_sa := C_Asynchronous,
                     TCV_ce := C_transparent)
93                     [ur_a = C_300bs]
94                       +local_Bs61or81_async_T(C_300bs,
                         C_modemt_V21, ri)
95                     [ur_a = C_1200bs]
96                       +local_Bs61or81_async_T(C_1200bs,
                         C_modemt_V22, ri)
97                     [ur_a = C_120075bs]
98                       +local_Bs61or81_async_T(C_120075bs,
                         C_modemt_V23, ri)
99                     [ur_a = C_2400bs]
100                      +local_Bs61or81_async_T(C_2400bs,
                         TCV_MODEM, ri)
101                    [ur_a = C_4800bs]
102                      +local_Bs61or81_async_T(C_4800bs,
                         C_modemt_V32, ri)
103                    [ur_a = C_9600bs]
104                      +local_Bs61or81_async_T(C_9600bs,
                         C_modemt_V32, ri)
105                [ce_a = C_notransparent]
106                  (TCV_ur := ur_a, TCV_sa := C_Asynchronous,
                     TCV_ce := C_notransparent)
107                    [ur_a = C_300bs]
108                      +local_Bs61or81_async_NT( C_300bs,
                         C_modemt_V21, ri)
109                    [ur_a = C_1200bs]
110                      +local_Bs61or81_async_NT( C_1200bs,
                         C_modemt_V22, ri)
111                    [ur_a = C_120075bs]
112                      +local_Bs61or81_async_NT( C_120075bs,
                         C_modemt_V23, ri)
113                    [ur_a = C_2400bs]
114                      +local_Bs61or81_async_NT( C_2400bs,
                         TCV_MODEM, ri)
115                    [ur_a = C_4800bs]
116                      +local_Bs61or81_async_NT( C_1200bs,
                         C_modemt_V32, ri)
117                    [ur_a = C_9600bs]
118                      +local_Bs61or81_async_NT( C_1200bs,
                         C_modemt_V32, ri)
119              [sync]
120                (TCV_ur := ur_s, TCV_sa := C_Synchronous,
                   TCV_ce := C_transparent)
121                  [ur_s = C_1200bs]
122                    +local_Bs61or81_sync_T(C_1200bs,
                       C_modemt_V22, ri)
123                  [ur_s = C_2400bs]
124                    +local_Bs61or81_sync_T(C_2400bs,
                       TCV_MODEM, ri)
125                  [ur_s = C_4800bs]
126                    +local_Bs61or81_sync_T(C_4800bs,
                       C_modemt_V32, ri)
127                  [ur_s = C_9600bs]
128                    +local_Bs61or81_sync_T(C_9600bs,
                       C_modemt_V32, ri)


                 local_Bs61or81_async_T(usrrt:B_4;
                 modemtp:B_5; rpi:RPI)
129              [usrrt <> C_9600bs]
130                (TCV_BcapMO1 := Bcap_MO( TCV_Rchr,                    11.8.2.5.1
                   C_3100Hz, C_Unstructured, C_nirr_nomeaning,
                   C_no_rate_adaption, C_I440_450,
                   C_Asynchronous, C_stopbit_dontcare,
                   C_databit_dontcare, usrrt, C_ir_8kbs,
                   C_parity_dontcare, C_transparent,   modemtp))
```

131		+local_callcfm2(rpi)			
132		[usrrt = C_9600bs]			
133		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare,usrrt, C_ir_16kbs, C_parity_dontcare, C_transparent, modemtp))			11.8.2.5.1
134		+local_callcfm2(rpi)			
		local_Bs61or81_async_NT(usrrt1:B_4; modemtp2:B_5; rpi:RPI)			
135		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_3100Hz, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, usrrt1, C_ir_16kbs, C_parity_dontcare, C_notransparent, modemtp2, C_uil2p_dontcare))			11.8.2.5.2
136		+local_callcfm2(rpi)			
		local_Bs61or81_sync_T(usrrt2:B_4; modemtp2:B_5; rpi2:RPI)			
137		[usrrt2 <> C_9600bs]			
138		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Synchronous, C_stopbit_NA, C_databit_NA, usrrt2, C_ir_8kbs, C_parity_NA, C_transparent, modemtp2))			11.8.2.5.3
139		+local_callcfm2(rpi2)			
140		[usrrt2 = C_9600bs]			
141		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Synchronous, C_stopbit_NA, C_databit_NA, usrrt2, C_ir_16kbs, C_parity_NA, C_transparent, modemtp2))			11.8.2.5.3
142		+local_callcfm2(rpi2)			
		local_TS61(svc:SERVICES)			
143		(TCV_ur := TSPX_TS_61_ur1, TCV_ce := TSPX_TS_61_ce)			
144	L_0991	[(TCV_ur <> C_4800bs) AND(TCV_ur <> C_2400bs) AND(TCV_ur <> C_9600bs)]		I	IXIT parameter error
145	L_0992	[(TCV_ce <> C_transparent) AND(TCV_ce <> C_notransparent)]		I	IXIT parameter error
146		[C_Yes]			
147		+local_Bcap_TS6x			
148		+local_callcfm2(C_RI_alternate)			
149		(TCV_Null := OO_IFsetup_TS6x_MO(svc, TCV_ur, TCV_ce, TCV_ChRate))			
		local_TS62(svc:SERVICES)			
150		(TCV_ur := TSPX_TS_62_ur1, TCV_ce := TSPX_TS_62_ce)			
151	L_0993	[(TCV_ur <> C_4800bs) AND(TCV_ur <> C_2400bs) AND(TCV_ur <> C_9600bs)]		I	IXIT parameter error
152	L_0994	[(TCV_ce <> C_transparent) AND(TCV_ce <> C_notransparent)]		I	IXIT parameter error
153		[C_Yes]			
154		+local_Bcap_TS6x			
155		+local_callcfm1			
156		(TCV_Null := OO_IFsetup_TS6x_MO(svc, TCV_ur, TCV_ce, TCV_ChRate))			
		local_Bcap_TS6x			
157		[TCV_ce = C_transparent]			
158		[TCV_ur = C_9600bs]			
159		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_FAX3, C_Unstructured, C_nirr_nomeaning,			11.8.2.7.1

		C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, TCV_ur, C_ir_16kbs, C_parity_NA, C_transparent, C_modemt_none))			
160		[(TCV_ur = C_4800bs) OR (TCV_ur = C_2400bs)]			
161		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_FAX3, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, TCV_ur, C_ir_8kbs, C_parity_NA, C_transparent, C_modemt_none))			11.8.2.7.1
162		[TCV_ce = C_notransparent]			
163		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_FAX3, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, TCV_ur, C_ir_16kbs, C_parity_NA, C_notransparent, C_modemt_none, C_X25_flct))			11.8.2.7.2
		local_callcfm1			
164		(TCV_CallCfm := CallConfirm_04(TI_01, TCV_BcapMO1))			
		local_callcfm2(rpi:RPI)			
165		[(NOT TSPC_EFR_Speech_v2) AND(NOT TSPC_EFR_Speech_v3)]			
166		(TCV_CallCfm := CallConfirm_05(TI_01, rpi, Bcap_Speech_MO(TCV_Rchr), TCV_BcapMO1))			Non EFR Mobile
167		[TSPC_EFR_Speech_v2 OR TSPC_EFR_Speech_v3]			
168		(TCV_CallCfm := CallConfirm_05(TI_01, rpi, Bcap_Speech_Efr_MO(TCV_Rchr), TCV_BcapMO1))			EFR Mobile
Detailed Comments:		The implementation of this test step is based on the SMG 7 Tdoc 266/96			

Test Step Dynamic Behaviour

Test Step Name: Chmod(rate:RATE; ce:B_2; ur:B_4; svc:SERVICES)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To assign correct values to the test case variables TCV_ChMod and TCV_ChModb.

Default: OtherEvents

Comments: The rules for assignment are in Table 10-1 of GSM 11.10 section 10.3. If the value for ce is "BothT", the rule for transparent is used. If the value for ce is "BothNT", the rule for non transparent is used. That means the preferred value will be used in test configuration.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+local_tree			
2		[(svc = C_AltSpchData) OR (svc = C_SpchData) OR (svc = C_AltSpchFax)]			dual mode call
3		(TCV_ChMod.mode := C_ChMod_r)			
4		[(svc<> C_AltSpchData) AND(svc <> C_SpchData) AND(svc <> C_AltSpchFax)]			
		local_tree			
5		[((ce = C_notransparent) OR (ce = C_BothNT)) AND (rate = C_Full)]			
6		(TCV_ChMod.mode := C_ChMod_12k, TCV_ChModb.mode := C_ChMod_12k)			
7		[((ce = C_notransparent) OR (ce = C_BothNT)) AND (rate = C_Half)]			
8		[ur <> C_9600bs]			
9		(TCV_ChMod.mode := C_ChMod_6k, TCV_ChModb.mode := C_ChMod_6k)			
10	L_0995	[ur = C_9600bs]		I	not applicable
11		[(ce = C_transparent) OR (ce = C_BothT)]			
12		[(ur = C_9600bs) AND (rate = C_Full)]			
13		(TCV_ChMod.mode := C_ChMod_12k, TCV_ChModb.mode := C_ChMod_12k)			
14		[ur = C_4800bs]			
15		(TCV_ChMod.mode := C_ChMod_6k, TCV_ChModb.mode := C_ChMod_6k)			
16		[(ur <> C_4800bs) AND(ur <> C_9600bs)]			
17		(TCV_ChMod.mode := C_ChMod_3k, TCV_ChModb.mode := C_ChMod_3k)			
18	L_0996	[(ur = C_9600bs) AND (rate = C_Half)]		I	not applicable

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Est_MO_Call(ta:TA; rate:IA5String; rand:RAND; sdcch:B_2; slot:SN; tsc:TSC; tch:B_1)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To establish a mobile station originating call using standard procedure.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+InitCall(TCV_Service)			
2		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
3		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
4		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, sdcch, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
5		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
6		ACTIVATE(OtherEvents)			Restore Normal default
7		+Authentication(TCV_ch, TCV_cksn, rand)			
8		+Ciphering_on(TCV_ch)			
9		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
10		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
11		L!DL_DatRqAlert	AlertSnd(TCV_ch, Alerting_01(TCV_TI))		
12		+AssCmdGen(TCV_cellid, rate, slot, tsc, tch)			
13		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
14		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
15		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Est_MO_Call_init(par1: OCTETSTRING; mag, mad:MA; ta:TA; tsc1:B_3; nfrqg, nfrqd:INTEGER)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To initiate a mobile originating call for the supported bearer capability. The channel in use is frequency hopping channel.

Default: OtherEvents

Comments: The test case variable TCV_Mt holds the message type of the received SETUP msg.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_cellid:=C_CellA)			
2		+InitCall(TCV_Service)			
3		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
4		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
5		+localtree_ia(par1)			
6		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
7		ACTIVATE(OtherEvents)			Restore Normal default
8		+ltree_continue			
		ltree_continue			
9		(TCV_Null := OM_NoL2Ack(2,TCV_ch))			Don't acknowledge setup message
10		+Ciphering_on(TCV_ch)			
11		L?DL_DatInSetup (TCV_Mt := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
		localtree_ia(par_chn: OCTETSTRING)			
12		[par_chn = C_CHTCHF_NonFH]			
13		(TCV_Tchtype := '00001'B)			
14		+ltree_immass_nfh(TCV_tch_arfcn, tsc1)			
15		[par_chn = C_CHSDCCH4_NonFH]			
16		(TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubDef)), 5))			
17		+ltree_immass_nfh(TCV_chdescr_arfcn, tsc1)			
18		[par_chn = C_CHSDCCH8_NonFH]			
19		(TCV_Tchtype := INT_TO_BIT((8 + BIT_TO_INT(TSPX_SDCCH8SubDef)), 5))			
20		+ltree_immass_nfh(TCV_tch_arfcn, tsc1)			
21		[par_chn = C_CHTCHF_FH]			
22		(TCV_Tchtype := '00001'B)			
23		+ltree_immass_fh(nfrqg, nfrqd)			
24		[par_chn = C_CHTCHH_FH]			
25		(TCV_Tchtype := INT_TO_BIT((2 + BIT_TO_INT(TSPX_TCHHSubDef)), 5))			
26		+ltree_immass_fh(nfrqg, nfrqd)			
27		[par_chn = C_CHTCHH_NonFH]			
28		(TCV_Tchtype := INT_TO_BIT((2 + BIT_TO_INT(TSPX_TCHHSubDef)), 5))			
29		+ltree_immass_nfh(TCV_tch_arfcn, tsc1)			
30		[par_chn = C_CHSDCCH8_FH]			
31		(TCV_Tchtype := INT_TO_BIT((8 + BIT_TO_INT(TSPX_SDCCH8SubDef)), 5))			
32		+ltree_immass_fh(nfrqg, nfrqd)			
		ltree_immass_fh(frqnog, frqnod:INTEGER)			
33		[TSPC_PGSM OR TSPC_EGSM]			
34		(TCV_maio := INT_TO_BIT((TSPX_MAIO MOD frqnog), 6), TCV_iel := mag.iel)			1.
35		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_221(TCV_Rr, TCV_Fn, TCV_ia_ts, tsc1, TCV_Tchtype, TCV_maio, INT_TO_BIT(TSPX_HSN, 6), ta, mag, TCV_iel))		

36		[TSPC_DCS]			
37		(TCV_maio := INT_TO_BIT((TSPX_MAIO MOD frqnod), 6), TCV_iel := mad.iel)			1.
38		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_221(TCV_Rr, TCV_Fn, TCV_ia_ts, tsc1, TCV_Tchtype, TCV_maio, INT_TO_BIT(TSPX_HSN, 6), ta, mad, TCV_iel))		
		ltree_immass_nfh(arfcn:INTEGER; tsc:B_3)			
39		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_nfh(TCV_Rr, TCV_Fn, TCV_Tchtype, TCV_ia_ts, tsc, ta, arfcn, C_normal_paging))		

Detailed Comments: 1. To make the MAIO is within 0 ... N-1, where N is the number of frequencies of MA. "frqnog" and "frqnod" are number of of frequencies of MA for GSM and DCS respectively.

Test Step Dynamic Behaviour

Test Step Name: Est_MT_Call_FH(ta:TA; rate:IA5String; maio:MAIO; hsn:INTEGER; rand:RAND; slot:SN; tsc:TSC; tch:B_1)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To establish a mobile station terminating call with hopping channel (speech or data call).

Default: OtherEvents

Comments: used var's:
TCV_cellid, TCV_ch

Nr	Label	Behaviour Description	CRef	V	Comments
1		+RRmtcallprepare(ta, rand)			
2		+CC_Est_MT_Call			
3		+localtree			
4		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
5		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
		localtree			
6		[rate = C_Full]			Full rate
7		[TCV_cellid = C_CellA]			
8		+AssCmdGen_fh(C_Full, slot, tsc, tch, 9, 15, maio, INT_TO_BIT(hsn, 6), Frql_20_A, Frql_20_Ad, CellChDes_Omit, CellChDes_Omit, ChMod_omit, MoblAllc_omit, MoblAllc_omit, CphMod_omit)			
9		[TCV_cellid = C_CellB]			
10		+AssCmdGen_fh(C_Full, slot, tsc, tch, 9, 15, maio, INT_TO_BIT(hsn, 6), Frql_20_B1, Frql_20_B8d, CellChDes_Omit, CellChDes_Omit, ChMod_omit, MoblAllc_omit, MoblAllc_omit, CphMod_omit)			
11		[rate = C_Half]			Half rate
12		[TCV_cellid = C_CellA]			
13		+AssCmdGen_fh(C_Half, slot, tsc, tch, 9, 15, maio, INT_TO_BIT(hsn, 6), Frql_20_A, Frql_20_Ad, CellChDes_Omit, CellChDes_Omit, ChMod_omit, MoblAllc_omit, MoblAllc_omit, CphMod_omit)			
14		[TCV_cellid = C_CellB]			
15		+AssCmdGen_fh(C_Half, slot, tsc, tch, 9, 15, maio, INT_TO_BIT(hsn, 6), Frql_20_B3, Frql_20_B3d, CellChDes_Omit, CellChDes_Omit, ChMod_omit, MoblAllc_omit, MoblAllc_omit, CphMod_omit)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Est_MT_CallNonFH(ta:TA; rate:IA5String; sub:B_1; rand:RAND)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To establish a mobile station terminating call with non hopping channel (speech or data call).

Default: OtherEvents

Comments: used var's:
TCV_cellid, TCV_chdescr_arfcn, TCV_Bcap1, TCV_chTch, TCV_asscmd_ts

Nr	Label	Behaviour Description	CRef	V	Comments
1		+RRmtcallprepare(ta, rand)			
2		+CC_EstMsTermCall(rate, sub)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: IdleUpdated(par_bspwr:INTEGER; activ_cell:CellID; sch, bcch:LOGICCH; acttype:BITSTRING; slot, slot2, slot3, slot4:SN; tsc:TSC; t, retr, neci, att:INTEGER; ta:TA; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm, cchddcs:CCHD; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; StartType:INTEGER; subch:B_2; cksn:B_3; rand:RAND; arfcngsm, arfcndcs:INTEGER; imsi:HEXSTRING; sysinfonb:INTEGER; nccp:NCCP)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To ensure that the SIM is either updated to the initial conditions and the MS with CKSN valid, TMSI valid and idle updated in cell A, B or C; or idle updated without TMSI; or totally not updated and all LAI information deleted.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_slot := slot, TCV_tsc := tsc, TCV_Ccd0H := CntrlChDscrp(att, babr, C_cch_1Comb, bpm, t3212))			
2		+MM_PwrOrSimOff(C_SIMneedRmv)			
3		+Varinit_fix(activ_cell, lac, subch, cksn, rand, arfcngsm, arfcndcs, TCV_Ccd0H, imsi)			
4		+ltree_chconf(C_cch_1Comb)			Config LT
5		+localtree1			
		localtree1			
6		+ltree_sysinfo(1, C_cch_1Comb)			ATT= '1'B
7		+MM_PwrOrSimOn(C_SIMneedRmv)			
8		+localtree_Lup_Auth			
		localtree_Lup_Auth			
9		START T_dly(C_T_Wait1stChReq)			
10		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_02)		Any channel request PDU
11		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
12		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, subch, slot, tsc, TCV_chdescr_arfcn, ta))		
13		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(C_norm_period_attach))		Any location update type
14		ACTIVATE(OtherEvents)			Restore Normal default
15		[StartType = C_Start_Rej]			
16		L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(C_rc_LAnotallowed))		Delete LAI, tmsi, cksn
17		+PostMainLinkRel(TCV_ch)			Release Ch
18		+MM_PwrOrSimOff(C_SIMneedRmv)			
19		+localtree2			
20		[StartType = C_Start_Imsi]			
21		+Authentication(TCV_ch, cksn, rand)			
22		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiImsi_01iei, mcc, mnc, TCV_lac))		Location Accept with IMSI
23		+PostMainLinkRel(TCV_ch)			Release Ch
24		+localtree2			
25		[(StartType = C_Start_Tmsi) OR (StartType = C_Start_TmsiOff)]			
26		+Authentication(TCV_ch, cksn, rand)			
27		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiTmsi_01iei, mcc, mnc, TCV_lac))		Location Accept with TMSI
28		L?DL_DatInTmsireCom	TmsiReallocCmp(TCV_ch)		
29		+PostMainLinkRel(TCV_ch)			Release Ch

30		+localtree2			
31		[StartType = C_Start_PLMNnot]			
32		L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(C_rc_plmn_not))		Delete LAI, tmsi, cksn
33		+PostMainLinkRel(TCV_ch)			Release Ch
34		+localtree2			
35	L_0997	[(StartType <> C_Start_Rej) AND(StartType <> C_Start_Tmsi) AND(StartType <> C_Start_TmsiOff) AND(StartType <> C_Start_Imsi) AND(StartType <> C_Start_PLMNnot)]		I	
36		?TIMEOUT T_dly			
37		+ltree_rmvPLMNfbdnlist			
		localtree2			
38		[(att = 1) AND(StartType <> C_Start_TmsiOff)]			ATT no change
39		[cch_con <> C_cch_1Comb]			If non combined
40		+localtree3			
41		+ltree_PgReorg			
42		[cch_con = C_cch_1Comb]			If combined
43		[(att = 1) AND (StartType = C_Start_TmsiOff)]			
44		+ImsiDetach(MiTmsi_01, ta, C_SIMneedRmv)			Switch off, detach
45		[cch_con <> C_cch_1Comb]			If non combined
46		+localtree3			
47		[cch_con = C_cch_1Comb]			If combined
48		[att = 0]			
49		[cch_con = C_cch_1Comb]			If combined
50		+ltree_sysinfo(0, C_cch_1Comb)			ATT=0
51		+ltree_PgReorg			
52		[cch_con <> C_cch_1Comb]			If non combined
53		+localtree3			
54		+ltree_PgReorg			
		localtree3			
55		(TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
56		+Varinit_fix(activ_cell, lac, subch, cksn, rand, arfcngsm, arfcndcs, TCV_Ccd0H, imsi)			
57		+ltree_chconf(cch_con)			Config LT
58		+ltree_sysinfo(att, cch_con)			
		ltree_PgReorg			
59		(TCV_Null:=OM_PgFill(activ_cell, PgReqTp1Reorg))			
60		START T_dly(5000)			
61		?TIMEOUT T_dly			
62		(TCV_Null:=OM_PgFill(activ_cell, PgReqTp1Norm))			
		ltree_sysinfo(attach:INTEGER; con_ch:B_3)			
63		[activ_cell = C_CellA]			
64		+SysInfoSending(attach, con_ch)			SI1-4, cellA
65		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
66		[con_ch = C_cch_1Comb]			If combined
67		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			SI5, 6 on SACCH/C4, cell A
68		[con_ch <> C_cch_1Comb]			
69		+ltree_sysinfo_other(attach)			More SIs, cellA
70		[activ_cell = C_CellB]			
71		+SysInfoSending(attach, con_ch)			SI1-4, cellB
72		(TCV_Ccd0B := TCV_Ccd0H, TCV_sacch_B := TCV_sacch, TCV_sysinfo5_B := TCV_sysinf5, TCV_sysinfo6_B := TCV_sysinf6)			SI5, 6 on SACCH/C4, cell B
73		[con_ch = C_cch_1Comb]			If combined
74		+SysInfo_SacchSending(TCV_sacch_B, TCV_sysinfo5_B, TCV_sysinfo6_B)			SI5, 6 on SACCH/C4, cell A

75		[con_ch <> C_cch_1Comb]			No more SIs, cell B
76		[activ_cell = C_CellC]			
77		+SysInfoSending(attach, con_ch)			SI1-4, cellC
78		(TCV_Ccd0C := TCV_Ccd0H, TCV_sacch_C := TCV_sacch, TCV_sysinfo5_C := TCV_sysinf5, TCV_sysinfo6_C := TCV_sysinf6)			
79		[con_ch = C_cch_1Comb]			If combined
80		+SysInfo_SacchSending(TCV_sacch_C, TCV_sysinfo5_C, TCV_sysinfo6_C)			SI5, 6 on SACCH/C4, cell C
81		[con_ch <> C_cch_1Comb]			No more SIs, cell C
		SysInfoSending(attachb:INTEGER; con_chs:B_3)			
82		[sysinfonb = C_Test_fh]			Hopping
83		+SysInfoSending_fh(sch, bcch, t, retr, neci, attachb, babr, con_chs, bpm, t3212, ci, mcc, mnc, lac, co, cchdgsm, cchddcs, crh, mtmcgsm, mtmcdcs, bcchflgsm, bcchfldcs, Re, bcc, ncc, nccp)			
84		[sysinfonb = C_Test_nfh]			Non hopping
85		+SysInfoSending_nfh(sch, bcch, t, retr, neci, attachb, babr, con_chs, bpm, t3212, ci, mcc, mnc, lac, co, crh, mtmcgsm, mtmcdcs, bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs, Re, bcc, ncc, nccp)			
86		[sysinfonb = C_Test_egsm]			EGSM
87		+SysInfoSending_e(sch, bcch, t, retr, neci, attachb, babr, con_chs, bpm, t3212, ci, mcc, mnc, lac, co, cchdgsm, crh, mtmcgsm, bcchflgsm, bcchfldcs, Re, bcc, ncc, nccp)			
88		[sysinfonb = C_Test_cbms]			CBMS
89		+SysInfoSending_cbms(sch, bcch, t, retr, neci, attachb, babr, con_chs, bpm, t3212, ci, mcc, mnc, lac, co, crh, mtmcgsm, mtmcdcs, bcchflgsm, bcchfldcs, Re, bcc, ncc, nccp, subch)			
		ltree_chconf(conf:B_3)			
90		[activ_cell = C_CellA]			
91		+ltree_chconf_A(conf)			Config LT
92		[activ_cell = C_CellB]			
93		+ltree_chconf_B(conf)			Config LT
94		[activ_cell = C_CellC]			
95		+ltree_chconf_C(conf)			Config LT
		ltree_chconf_A(conf_a:B_3)			
96		[conf_a = C_cch_1nonComb]			
97		+NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1 non combined CCCH, cell A
98		(TCV_Upd := C_NotCombined)			
99		[conf_a = C_cch_1Comb]			
100		[sysinfonb <> C_Test_cbms]			
101		+CombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1 combined CCCH, cell A
102		(TCV_Upd := C_Combined)			
103		[sysinfonb = C_Test_cbms]			
104		+CombinedBCCH_A_CBMS(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1 combined CCCH + CBMS, cell A
105		(TCV_Upd := C_Combined)			
106		[conf_a = C_cch_2nonComb]			2 CCCHs
107		+NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1st non combined CCCH, cell A
108		+NonCombinedBCCH_A_2(acttype, slot2, tsc, ta, babr, conf_a, bpm)			2nd non combined CCCH, cell A
109		(TCV_Upd := C_NotCombined)			
110		[conf_a = C_cch_3nonComb]			
111		+NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta,			1st non combined CCCH, cell A

		babr, conf_a, bpm)			
112		+NonCombinedBCCH_A_2(acttype, slot2, tsc, ta, babr, conf_a, bpm)			2nd non combined CCCH, cell A
113		+NonCombinedBCCH_A_3(acttype, slot3, tsc, ta, babr, conf_a, bpm)			3rd non combined CCCH, cell A
114		(TCV_Upd := C_NotCombined)			
115		[conf_a = C_cch_4nonComb]			
116		+NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1st non combined CCCH, cell A
117		+NonCombinedBCCH_A_2(acttype,slot2, tsc, ta, babr, conf_a, bpm)			2nd non combined CCCH, cell A
118		+NonCombinedBCCH_A_3(acttype, slot3, tsc, ta, babr, conf_a, bpm)			3rd non combined CCCH, cell A
119		+NonCombinedBCCH_A_4(acttype, slot4, tsc, ta, babr, conf_a, bpm)			4th non combined CCCH, cell A
120		(TCV_Upd := C_NotCombined)			
121	L_0998	[(conf_a <> C_cch_1nonComb) AND(conf_a <> C_cch_2nonComb) AND(conf_a <> C_cch_3nonComb) AND(conf_a <> C_cch_4nonComb) AND(conf_a <> C_cch_1Comb)]		I	A wrong CCCH-CONF value, cell A
		ltree_chconf_B(conf_b:B_3)			
122		[conf_b = C_cch_1nonComb]			
123		+NonCombinedBCCH_B(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype,slot,tsc, ta, babr, conf_b, bpm)			1 non combined CCCH, cell B
124		(TCV_Upd := C_NotCombined)			
125		[conf_b = C_cch_1Comb]			
126		+CombinedBCCH_B(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_b, bpm)			1 combined CCCH, cell B
127		(TCV_Upd := C_Combined)			
128	L_0999	[(conf_b <> C_cch_1nonComb) AND(conf_b <> C_cch_1Comb)]		I	A wrong CCCH-CONF value,cell B
		ltree_chconf_C(conf_c:B_3)			
129		[conf_c = C_cch_1Comb]			
130		+CombinedBCCH_C(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_c, bpm)			1 combined CCCH, cell C
131		(TCV_Upd := C_Combined)			
132	L_1000	[conf_c <> C_cch_1Comb]		I	
		ltree_sysinfo_other(attach:INTEGER)			
133		[(cch_con = C_cch_1nonComb) OR (cch_con = C_cch_1Comb)]			
134		[cch_con = C_cch_2nonComb]			
135		+SysInfoSending_fh(sch, C_BCCH_A_2, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 2nd CCCH
136		[cch_con = C_cch_3nonComb]			
137		+SysInfoSending_fh(sch, C_BCCH_A_2, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 2nd CCCH
138		+SysInfoSending_fh(sch, C_BCCH_A_3, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 3rd CCCH
139		[cch_con = C_cch_4nonComb]			
140		+SysInfoSending_fh(sch, C_BCCH_A_2, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 2nd CCCH

141		+SysInfoSending_fh(sch, C_BCCH_A_3, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 3rd CCCH
142		+SysInfoSending_fh(sch, C_BCCH_A_4, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 4th CCCH
		ltree_rmvPLMNfbdnlist			
143		(TCV_Null := OO_PLMNselModeMan())			
144		(TCV_Null := OO_SelPLMN(mnc))			
145		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_02)		Any channel request PDU
146		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
147		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, subch, slot, tsc, TCV_chdescr_arfcn, ta))		
148		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(C_norm_period_attach))		Any location update type
149		ACTIVATE(OtherEvents)			Restore Normal default
150		[StartType = C_Start_PLMNnot]			The PLMN is already in forbidden list on short SIM
151		L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(C_rc_plmn_not))		Delete LAI, tmsi, cksn, PLMN still in list
152		+PostMainLinkRel(TCV_ch)			Release Ch
153		(TCV_Null := OO_PLMNselModeAuto())			
154		[StartType <> C_Start_PLMNnot]			
155		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiTmsi_r02iei, mcc, mnc, TCV_lac))		Location Accept with no TMSI
156		+PostMainLinkRel(TCV_ch)			Release Ch
157	L_1001	(TCV_Null := OO_PLMNselModeAuto())		I	No ChReq received: Inconclusive

Detailed Comments:

Formal Parameter Type Comments & Explanations Possible values

par_bspwr INTEGER Base station power level C_E_default, C_E_suitable

activ_cell CellID Activated cell identity C_CellA, C_CellB, C_CellC

sch LOGICCH Sync. channel identity
If formal para activ_cell = C_CellA, C_SCH_A
If formal para activ_cell = C_CellB, C_SCH_B
If formal para activ_cell = C_CellC, C_SCH_C

bcch LOGICCH 1st broadcast channel id
If formal para activ_cell = C_CellA, C_BCCH_A_1
If formal para activ_cell = C_CellB, C_BCCH_B_1
If formal para activ_cell = C_CellC, C_BCCH_C_1

acttype BITSTRING Type for activation of a channel C_Immass

slot SN Slot number for the 1st slot C_S0

slot2 SN Slot number for the 2nd slot
In case of 2 or 3 or 4 non combined CCCH C_S2

Otherwise (1 combined or 1 non combined) C_SlotNU

slot3 SN Slot number for the 3rd slot
In case of 3 or 4 non combined CCCH C_S4
Otherwise (1 combined or 1, 2 non combined) C_SlotNU

slot4 SN Slot number for the 4th slot
In case of 4 non combined CCCH channels C_S6
Otherwise (1 combined,1, 2, 3 non combined) C_SlotNU

tsc TSC Training Sequence code C_BCC

t INTEGER Rach Control Parameter field, Tx-Integer
for SysInfo1, SysInfo2, SysInfo3, SysInfo4 C_TxInt_5, C_TxInt_7, C_TxInt_10

retr INTEGER Rach Control Parameter field, Max-Retrans
for SysInfo1, SysInfo2, SysInfo3, SysInfo4 C_Max_1, C_Max_2, C_Max_7

neci INTEGER Requested value f. NECI bit in test case C_NECI_0, C_NECI_1

att INTEGER Requested value f. ATTACH bit in test case C_ATT_0, C_ATT_1

ta TA Requested Timing Advance value

babr B_3 Number of blocks reserved for acces grant
If arbitrary TSPX_AGBLKSx
If fixed C_BABR_0, C_BABR_1, C_BABR_2

cch_con B_3 Type of CCCH channel.
If arbitrary TSPX_CcchConfx
1 conbined CCCH channel C_cch_1Comb
1 non conbined CCCH channel C_cch_1nonComb
2 non conbined CCCH channels C_cch_2nonComb
3 non conbined CCCH channels C_cch_3nonComb
4 non conbined CCCH channels C_cch_4nonComb

bpm B_3 Number of multiframes for transmission
of PAGING REQUEST messages to the same
paging subgroup
If arbitrary TSPX_PAMFRMSx
If fixed (2, 4, 5, 9 multiframes) C_BPM_0, C_BPM_2, C_BPM_3, C_BPM_7

t3212 OCTETSTR. Requested value for T3212 in test case (Periodic Location Update Value)
infinite C_T3212_0
6 mn / 1 deci hours C_T3212_1
12 mn / 2 deci hours C_T3212_2
30 mn / 5 deci hours C_T3212_5

ci CI Cell Identity
If formal para activ_cell = C_CellA C_ci_CellA
If formal para activ_cell = C_CellB C_ci_CellB
If formal para activ_cell = C_CellC C_ci_CellC

mcc OCTETSTR. Mobile Contry Code for the LAI C_MCC, C_MCC_x

mnc OCTETSTR. Mobile Network Code for the LAI C_PLMN_Home, C_PLMN_x

lac OCTETSTR. Local Area Code for the LAI C_LAC_x

co CO Cell Options CellOpt_xx

cchdgsm CCHD Cell Channels Description (GSM)
If formal para sysinfonb=C_Test_nfh CellChDes_omit
If formal para sysinfonb=C_Test_cbms CellChDes_omit
If formal para sysinfonb=C_Test_fh GSM CellChDes_xx

cchddcs CCHD Cell Channels Description (DCS)
If formal para sysinfonb=C_Test_nfh CellChDes_omit
If formal para sysinfonb=C_Test_cbms CellChDes_omit
If formal para sysinfonb=C_Test_fh DCS CellChDes_xx

crh INTEGER Cell Reselection Hysteresis C_CellReselectHys12, C_CellReselectHys4

mtmcgsm INTEGER Requested maximum power level for GSM MS C_MaxPwrLvlG

mtmcdcs INTEGER Requested maximum power level for DCS MS C_MaxPwrLvlD

bcchflgsm NCD Neighbour Cells Description for GSM BcchFreqLst_xx

bcflaltgsm NCD Alternative Neighbour Cells Description for GSM.
If sysinfonb = C_Test_nfh Alternat.GSM BcchFreqLst_xx
Otherwise: BcchFreqLst_Omit

bcchfldcs NCD Neighbour Cells Description for DCS BcchFreqLst_xx

bcflaltdcs NCD Alternative Neighbour Cells Description for DCS
If sysinfonb = C_Test_nfh Alternat.DCS BcchFreqLst_xx
Otherwise: BcchFreqLst_Omit

Re B_1 Rach Control Parameter field,
Re-establishment bit for SysInfo1,
SysInfo2, SysInfo3, SysInfo4 C_Restablishment, C_noRestablishmen
bcc BCC Base station Color Code C_BCC

ncc NCC Network Color Code C_NCC

StartType INTEGER Initial conditions for MS
Starting without TMSI C_Start_Imsi
Starting with Tmsi_01 C_Start_Tmsi
Starting with Tmsi_01, MS deactivated C_Start_TmsiOff
Starting with LAI deleted, MS deactivated C_Start_Rej
Starting with PLMN not allowed C_Start_PLMNnot

subch BITSTRING Selected sub-channel
If Formal para cch_con = C_cch_1Comb, TSPX_SDCCH4SubDef TSPX_SDCCH4SubA, SubB, SubC
cksn BITSTRING Ciphering Key Sequence Number used for authentication TSPX_CKSNDef, TSPX_CKSNA, B, C

rand BITSTRING Random Number for authentication TSPX_RANDDef, TSPX_RANDA, B , C

arfcngsm INTEGER BCCH channel ARFCN (GSM)
If sysinfonb = C_Test_fh C_Noarfcn
Otherwise: If activ_cell = C_CellA C_arfcnA_x If activ_cell = C_CellB C_arfcnB_x If activ_cell = C_CellC C_arfcnC_x

arfcndcs INTEGER BCCH channel ARFCN (DCS)
If sysinfonb = C_Test_fh C_Noarfcn

		Otherwise:	If activ_cell = C_CellA C_arfcnAd_x If activ_cell = C_CellB C_arfcnBd_x If activ_cell = C_CellC C_arfcnCd_x
imsi	HEXSTRING	The expected IMSI In TC_26_7_3_2	TSPX_IMSI C_shortIMSI
sysinfonb	INTEGER	Type of sysinfo sending	
		Frequency Hopping	C_Test_fh
		Non Frequency hopping	C_Test_nfh
		EGSM sequence	C_Test_egsm
		CBMS sequence	C_Test_cbms
nccp	NCCP	Network Country Code Permitted in SysInfo2	C_NCCP_2, C_NCCP_5

Test Step Dynamic Behaviour

Test Step Name:	PreEnterIdleState_r02(acttype:BITSTRING; slot:SN; tsc:TSC; tx, retrans:INTEGER; slot2:SN; slot3:SN; slot4:SN; ta:TA; att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING)
Group:	GSM_L3_MS_v4220/Preambles/
Objective:	To broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 containing default parameters in cell A with the following exceptions : - legal combination of CCCH-CONF, BS-AG-BLKS-RES, BS-PA-MFRMS are specified by parameters - max retransmission and tx-integer can be assigned, The test system is then waiting for the SUT (MS) entering the Idle updated state.
Default:	OtherEvents
Comments:	The preamble is used for the RR tests.

Nr	Label	Behaviour Description	CRef	V	Comments
1		[cch_con = C_cch_1Comb]			
2		+CombinedBCCH_A(63, C_MaxPwrLvlG, C_MaxPwrLvlD, C_arfcnA, C_arfcnAd, acttype, slot, tsc, ta, babr, cch_con, bpm)			
3		+SysInfoSending_fh(C_SCH_A, C_BCCH_A_1, tx, retrans, 0, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lac, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			
4		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
5		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			
6		[cch_con = C_cch_1nonComb]			
7		+localtree			
8		[cch_con = C_cch_2nonComb]			
9		+NonCombinedBCCH_A_2(acttype,slot2,tsc, ta, babr, cch_con, bpm)			
10		+SysInfoSending_fh(C_SCH_A, C_BCCH_A_2, tx, retrans, 0, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lac, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			
11		+localtree			
12		[cch_con = C_cch_3nonComb]			
13		+NonCombinedBCCH_A_2(acttype,slot2,tsc, ta, babr, cch_con, bpm)			
14		+NonCombinedBCCH_A_3(acttype,slot3,tsc, ta, babr, cch_con, bpm)			
15		+SysInfoSending_fh(C_SCH_A, C_BCCH_A_2, tx, retrans, 0, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lac, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			
16		+SysInfoSending_fh(C_SCH_A, C_BCCH_A_3, tx, retrans, 0, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lac, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			
17		+localtree			
18		[cch_con = C_cch_4nonComb]			
19		+NonCombinedBCCH_A_2(acttype,slot2,tsc, ta, babr, cch_con, bpm)			

20		+NonCombinedBCCH_A_3(acttype,slot3,tsc, ta, babr, cch_con, bpm)			
21		+NonCombinedBCCH_A_4(acttype,slot4,tsc, ta, babr, cch_con, bpm)			
22		+SysInfoSending_fh(C_SCH_A, C_BCCH_A_2, tx, retrans, 0, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lac, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			
23		+SysInfoSending_fh(C_SCH_A, C_BCCH_A_3, tx, retrans, 0, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lac, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			
24		+SysInfoSending_fh(C_SCH_A, C_BCCH_A_4, tx, retrans, 0, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lac, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			
25		+localtree			
26	L_1002	[(cch_con = C_cch_2Comb) OR (cch_con = C_cch_3Comb) OR (cch_con = C_cch_4Comb)]		I	Reserved
		localtree			
27		+NonCombinedBCCH_A(63, C_MaxPwrLvlG, C_MaxPwrLvlD, C_arfcnA, C_arfcnAd, acttype,slot,tsc, ta, babr, cch_con, bpm)			
28		+SysInfoSending_fh(C_SCH_A, C_BCCH_A_1, tx, retrans, 0, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lac, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			
29		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU01_21(sub:B_2; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U0.1 by procedure in table 26.8.1.2/1. This is used in CC testing.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+CCEstablishMO_SDCCH4(sub, ta)			1.
2		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
3		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_04)		
4		ACTIVATE(OtherEvents)			Restore Normal default tree

Detailed Comments: 1. To assign SDCCH4 channel.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU1_21(sub:B_2; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U1 by procedure in table 26.8.1.2/1. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body. The calling tree shall preare two variables for the step: TCV_ch for the SDCCH4 subchannel; TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU01_21(sub, ta)			
2		+Ciphering_on(TCV_ch)			
3		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
4		+CCstatuschk_02(TCV_ch, C_U1, TCV_TI, TCV_TI0)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU1_22(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U1 by procedure in table 26.8.1.2/2. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+CCEstablishMO_TCH(slot, tsc, ta)			1.
2		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
3		L?DL_EstInCmsRq (TCV_Fn :=DL_EstInCmsRq.fn)	CMSerReq(CMServiceReq_04)		
4		ACTIVATE(OtherEvents)			Restore Normal default
5		+CCModifyTCH(slot, tsc)			
6		+Ciphering_on(TCV_chTch)			
7		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
8		+CCstatuschk_02(TCV_chTch, C_U1, TCV_TI, TCV_TI0)			2.

Detailed Comments:
1. To assign TCH/F channel or TCH/H channel.
2. Check that CC is now state U1

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU1_22Timer(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U1 by procedure in table 26.8.1.2/2. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transaction identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+CCEstablishMO_TCH(slot, tsc, ta)			1.
2		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
3		L?DL_EstInCmsRq (TCV_Fn :=DL_EstInCmsRq.fn) START T_dly(45000)	CMSerReq(CMServiceReq_04)		
4		ACTIVATE(OtherEvents)			Restore Normal default
5		+CCModifyTCH(slot, tsc)			
6		+Ciphering_on(TCV_chTch)			
7		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
8		+CCstatuschk_02(TCV_chTch, C_U1, TCV_TI, TCV_TI0)			2.

Detailed Comments:
1. To assign TCH/F channel or TCH/H channel.
2. Check that CC is now state U1

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU1_24(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U1 by procedure in table 26.8.1.2/4. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI contains the transcation identifier from the MS, and TCV_TI0 contains the transcation identifier for test system to send CC message. The values of them are used in test body.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+CCEstablishMO_TCH(slot, tsc, ta)			1.
2		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
3		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_04)		
4		ACTIVATE(OtherEvents)			Restore Normal default
5		L!DL_DatRqIdRq	IDReq(TCV_chTch, IDRequest_01('0001'B))		IMSI.
6		L?DL_DatInIdRes	IDRes(IDResponse_30(MiImsi_01))		
7		+Ciphering_on(TCV_chTch)			
8		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
9		+CCstatuschk_02(TCV_chTch, C_U1, TCV_TI, TCV_TI0)			

Detailed Comments: 1. To assign TCH/F channel or TCH/H channel.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU1(ta:TA; cksn:B_3; rand:RAND)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To establish a mobile originating call and put the MS under test in the CC state U1.
Default: OtherEvents
Comments: test case variable TCV_TI0 holds the transaction IDused by the MS, and TCV_TI used by test system. The calling tree shall prepare two variables for the step: TCV_ch for the SDCCH4 subchannel; TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+InitCall(TCV_Service)			1.
2		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)		
3		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
4		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
5		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
6		ACTIVATE(OtherEvents)			Restore Normal default
7		+Authentication(TCV_ch, cksn, rand)			
8		+Ciphering_on(TCV_ch)			
9		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			2.
10		+CCstatuschk_02(TCV_ch, C_U1, TCV_TI, TCV_TI0)			

Detailed Comments: 1. To attempt a outgoing call at the MS.
2. In the state U1 now.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU3(ta:TA; cksn:B_3; rand:RAND)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To establish a mobile originating call and put the MS under test in the CC state U3.
Default: OtherEvents
Comments: test case variable TCV_TI0 holds the transaction ID used by the MS, and TCV_TI is used by test system. The calling tree shall preare two variables for the step: TCV_ch for the SDCCH4 subchannel; TCV_CphKey for ciphering key. The call setup is generic setup procedure.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU1(ta, cksn, rand)			1.
2		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		2.
3		+CCstatuschk_02(TCV_ch, C_U3, TCV_TI, TCV_TI0)			

Detailed Comments:
1. To initiate a mobile originating call at the MS.
2. Now in the state U3.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU3_21(sub:B_2; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U3 by procedure in table 26.8.1.2/1. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body. The calling tree shall preare two variables for the step: TCV_ch for the SDCCH4 subchannel; TCV_CphKey for ciphering key

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU1_21(sub, ta)			
2		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
3		+CCstatuschk_02(TCV_ch, C_U3, TCV_TI, TCV_TI0)			

Detailed Comments:
1. To assign SDCCH4 channel.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU3_22(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U3 by procedure in table 26.8.1.2/2. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU1_22(slot, tsc, ta)			
2		L!DL_DatRqCallProc	CallProc(TCV_chTch, TCV_CallProc)		
3		+CCstatuschk_02(TCV_chTch, C_U3, TCV_TI, TCV_TI0)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU3_23(sub:B_2; ta:TA; cksn:B_3; rand:RAND)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U3 by procedure in table 26.8.1.2/3. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. Their values are used in test body. The calling tree shall preare two variables for the step: TCV_ch for the SDCCH4 subchannel; TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+CCEstablishMO_SDCCH4(sub, ta)			
2		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
3		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_04)		
4		ACTIVATE(OtherEvents)			Restore Normal default tree
5		+Ciphering_on(TCV_ch)			
6		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
7		+Authentication(TCV_ch, cksn, rand)			
8		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
9		+CCstatuschk_02(TCV_ch, C_U3, TCV_TI, TCV_TI0)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU3_24(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U3 by procedure in table 26.8.1.2/4. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU1_24(slot, tsc, ta)			
2		+CCModifyTCH(slot, tsc)			
3		L!DL_DatRqCallProc	CallProc(TCV_chTch, TCV_CallProc)		
4		+CCstatuschk_02(TCV_chTch, C_U3, TCV_TI, TCV_TI0)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU4_21(sub:B_2; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U4 by procedure in table 26.8.1.2/1. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI contains the transcation identifier from the MS, and TCV_TI0 contains the transcation identifier for test system to send CC message. The values of them are used in test body.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU3_21(sub, ta)			
2		L!DL_DatRqAlert	AlertSnd(TCV_ch, Alerting_01(TCV_TI))		
3		+CCstatuschk_02(TCV_ch, C_U4, TCV_TI, TCV_TI0)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU4_22(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U4 by procedure in table 26.8.1.2/2. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU3_22(slot, tsc, ta)			
2		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
3		+CCstatuschk_02(TCV_chTch, C_U4, TCV_TI, TCV_TI0)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU4_23(sub:B_2; slot:SN; tsc:TSC; ta:TA; cksn:B_3; rand:RAND)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U4 by procedure in table 26.8.1.2/3. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body. The calling tree shall preare two variables for the step: TCV_ch for the SDCCH4 subchannel; TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU3_23(sub, ta, cksn, rand)			
2		+CCAssignTCH(slot, tsc)			
3		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
4		+CCstatuschk_02(TCV_chTch, C_U4, TCV_TI, TCV_TI0)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU4_24(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U4 by procedure in table 26.8.1.2/4. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU3_24(slot, tsc, ta)			
2		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
3		+CCstatuschk_02(TCV_chTch, C_U4, TCV_TI, TCV_TI0)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU6_32(ta:TA; sub:B_2; cksn:B_3)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U6 by procedure in table 26.8.1.3/2. This is used in CC testing.
Default: OtherEvents
Comments: State U6 is a transient state.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Setup_mt.sig :=Signal_01)			
2		+CCEstablishMT_SDCCH4(ta, sub, cksn)			1.
3		+Ciphering_on(TCV_ch)			
4		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		

Detailed Comments: 1. To assign SDCCH4 channel.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU7_31(ta:TA; sub:B_2; cksn:B_3; rand:RAND)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U7 by procedure in table 26.8.1.3/1. This is used in CC testing.
Default: OtherEvents
Comments: used only for the MS not support immediate connection. State U7 is transiet state if the MS supports automatic connect after a specific time. The calling tree shall prepare two variables for the step: TCV_ch for SDCCH4 subchannel, TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU9_31(ta, sub, cksn, rand)			
2		L?DL_DatInAlert	AlertRcv(AlertingInd_02(TI_01))		
3		+CCstatuschk_02(TCV_ch, C_U7, TI_02, TI_01)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU7_32(ta:TA; sub:B_2; cksn:B_3)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U7 by procedure in table 26.8.1.3/2. This is used in CC testing.
Default: OtherEvents
Comments: used only for the MS not support immediate connection. State U7 is transiet state if the MS supports automatic connect after a specific time. The calling tree shall prepare two variables for the step: TCV_ch for SDCCH4 subchannel, TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU9_32(ta, sub, cksn)			
2		L?DL_DatInAlert	AlertRcv(AlertingInd_02(TI_01))		
3		+CCstatuschk_02(TCV_ch, C_U7, TI_02, TI_01)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU7_33(slot:SN; tsc:TSC; ta:TA; cksn:B_3; rand:RAND)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U7 by procedure in table 26.8.1.3/3. This is used in CC testing.
Default: OtherEvents
Comments: used only for the MS not support immediate connection. State U7 is transiet state if the MS supports automatic connect after a specific time. The calling tree shall prepare two variables for the step: TCV_chTch for traffic channel, TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU9_33(slot, tsc, ta, cksn, rand)			
2		L?DL_DatInAlert	AlertRcv(AlertingInd_02(TI_01))		
3		+CCstatuschk_02(TCV_chTch, C_U7, TI_02, TI_01)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU8_31(ta:TA; sub:B_2; cksn:B_3; rand:RAND)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U8 by procedure in table 26.8.1.3/1. This is used in CC testing.
Default: OtherEvents
Comments: The calling tree shall prepare two variables for the step: TCV_ch for SDCCH4 subchannel, TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU9_31(ta, sub, cksn, rand)			
2		L?DL_DatInAlert	AlertRcv(AlertingInd_02(TI_01))		
3		(TCV_Null := OO_HookOff())			1.
4		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
5		+CCstatuschk_02(TCV_ch, C_U8, TI_02, TI_01)			2.
6		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
7		+CCstatuschk_02(TCV_ch, C_U8, TI_02, TI_01)			2.

Detailed Comments:
1. To accept the call by operator.
2. To check whether the MS is in the expected initial state U8.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU8_32(slot:SN; tsc:TSC; ta:TA; sub:B_2; cksn:B_3)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U8 by procedure in table 26.8.1.3/2. This is used in CC testing.
Default: OtherEvents
Comments: The calling tree shall prepare three variables for the step: TCV_ch for SDCCH4 subchannel, TCV_chTch for traffic channel, TCV_CphKey for ciphering key, TCV_AssCmd for ASSIGNMENT message.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU9_32(ta, sub, cksn)			
2		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
3		+CCAssignTCH(slot, tsc)			
4		+CCstatuschk_02(TCV_chTch, C_U8, TI_02, TI_01)			2.
5		L?DL_DatInAlert	AlertRcv(AlertingInd_02(TI_01))		
6		(TCV_Null := OO_HookOff())			1.
7		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
8		+CCAssignTCH(slot, tsc)			
9		+CCstatuschk_02(TCV_chTch, C_U8, TI_02, TI_01)			2.

Detailed Comments:
1. To accept the call by operator.
2. To verify whether the MS is in the expected initial state U8.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU8_33(slot:SN; tsc:TSC; ta:TA; cksn:B_3; rand:RAND)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U8 by procedure in table 26.8.1.3/3. This is used in CC testing.
Default: OtherEvents
Comments: The calling tree shall prepare three variables for the step: TCV_ch for SDCCH4 subchannel, TCV_chTch for traffic channel, TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU9_33(slot, tsc, ta, cksn, rand)			
2		L?DL_DatInAlert	AlertRcv(AlertingInd_02(TI_01))		
3		(TCV_Null := OO_HookOff())			
4		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
5		+CCstatuschk_02(TCV_chTch, C_U8, TI_02, TI_01)			3.
6		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
7		+CCstatuschk_02(TCV_chTch, C_U8, TI_02, TI_01)			3.

Detailed Comments:
1. Immediate connection is not supported.
2. Immediate connection is supported.
3. To verify whether the MS is in the expected initial state U8.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU8_34_SS
Group: GSM_L3_MS_v4220/Preambles/
Objective: To establish an MT call entering U8 with early assignment for SS test subgroup.
Default: OtherEventsFail
Comments: Used var's: TCV_cellid, TCV_chdescr_arfcn, TCV_Bcap1, TCV_chTch, TCV_asscmd_ts

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEstRRConn(TCV_slot, TCV_tsc, TimingAdv(0))			
2		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		
3		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
4		L?DL_DatInCallCo(TCV_CallCfm:=DL_DatInCallCo.msg)	CallCfm(CallConfirm_01(TI_01))		
5		[NOT TSPX_Immconn]			
6		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
7		L?DL_DatInAlert	AlertRcv(AlertingInd_01(TI_01))		
8		(TCV_Null:=OO_HookOff())			
9		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
10		[TSPX_Immconn]			
11		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
12		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU9_31(ta:TA; sub:B_2; cksn:B_3; rand:RAND)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U9 by procedure in table 26.8.1.3/1. This is used in CC testing.
Default: OtherEvents
Comments: The state U9 is a transient state when the signal IE is included in the SETUP message. The calling tree shall prepare three variables for the test step: TCV_ChRate for the type of the channel, TCV_CphKey for the ciphering key and TCV_ChMod for the channel mode.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Setup_mt.sig := Signal_01)			
2		+CCEstablishMT_SDCCH4(ta, sub, cksn)			1.
3		+Authentication(TCV_ch, cksn, rand)			2.
4		+Ciphering_on(TCV_ch)			3.
5		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		
6		L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))		

Detailed Comments:
1. To establish a MT SDCCH/4.
2. To initiate authentication procedure.
3. To start ciphering on the traffic channel.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU9_32(ta:TA; sub:B_2; cksn:B_3)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U9 by procedure in table 26.8.1.3/2. This is used in CC testing.
Default: OtherEvents
Comments: The state U9 in the test step is a transient state. The calling tree shall prepare two variables for the test step TCV_ch of SDCCH4 subchannel, TCV_CphKey.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU6_32(ta, sub, cksn)			
2		L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))		

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU9_33(slot:SN; tsc:TSC; ta:TA; cksn:B_3; rand:RAND)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U9 by procedure in table 26.8.1.3/3. This is used in CC testing.
Default: OtherEvents
Comments: The state U9 is a transient state when the signal IE is included in the SETUP message. The calling tree shall prepare three variables for the test step: TCV_ChRate for the type of the channel, TCV_CphKey for the ciphering key and TCV_ChMod for the channel mode.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Setup_mt.sig := Signal_01)			
2		+CCEstablishMT_TCH(slot, tsc, ta)			1.
3		+Authentication(TCV_chTch, cksn, rand)			2.
4		+Ciphering_on(TCV_chTch)			3.
5		+CCModifyTCH(slot, tsc)			4.
6		L!DL_DatRqSetup	SetupSnd(TCV_chTch, TCV_Setup_mt)		
7		L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))		

Detailed Comments:
1. To establish an MT on TCH/F or TCH/H.
2. To initiate authentication procedure.
3. To start ciphering on the traffic channel.
4. To modify the channel mode.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU9_34(ta:TA; sub:B_2; cksn:B_3)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U9 by procedure in table 26.8.1.3/4. This is used in CC testing.
Default: OtherEvents
Comments: The supported bearer capability is specified in the input partameter setup.The calling tree shall prepare two variables for the step: TCV_ch for SDCCH4 subchannel, TCV_CphKey for ciphering key. The `setup` shall contain no signal IE. The test step is used for the MS does not support immediate connect.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+CCEstablishMT_SDCCH4(ta, sub, cksn)			1.
2		+Ciphering_on(TCV_ch)			
3		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		2.
4		L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))		
5		+CCstatuschk_02(TCV_ch, C_U9, TI_02, TI_01)			3.

Detailed Comments:
1. To assign SDCCH4 channel.
2. SETUP message without SIGNAL IE.
3. To check whether the MS is in the initial state U9, if no the test step ends with inconclusive verdict in the default tree.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU10(sub:B_2; setup:SETUP_MT_PDU; slot:SN; tsc:TSC; ta:TA; cksn:B_3; rand:RAND)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To establish a mobile terminating call for the supported bearer capability and put the MS under test in the CC state U10.

Default: OtherEvents

Comments: The supported bearer capability is specified in the input partameter setup, test case variable TCV_TI holds the transaction ID and the transaction ID =0. This is generic call set up procedure. The calling tree shall prepare three variables for the step: TCV_ch for SDCCH4 subchannel, TCV_chTch for traffic channel, TCV_CphKey for ciphering key. This test step is used for non RR testing.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+CCEstablishMT_SDCCH4(ta, sub, cksn)			
2		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(cksn, rand))		
3		L?DL_DatInAuthRes (TCV_Sres := DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
4		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, rand))			
5	L_1003	[NOT TCV_Res]		(I)	1.
6		+PostLinkRelEnd(TCV_ch)			
7		[TCV_Res]			
8		+Ciphering_on(TCV_ch)			
9		L!DL_DatRqSetup (TCV_TI.ti_f := '0'B, TCV_TI.ti_v := '000'B)	SetupSnd(TCV_ch, setup)		
10		L?DL_DatInCallCo(TCV_CallCfm := DL_DatInCallCo.msg)	CallCfm(CallConfirm_01(TI_01))		
11		[TSPX_Immconn]			
12		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
13		+CCAssignTCH(slot, tsc)			
14		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
15		+CCstatuschk_02(TCV_chTch, C_U10, TI_02, TI_01)			
16		[NOT TSPX_Immconn]			
17		+CCAssignTCH(slot, tsc)			
18		L?DL_DatInAlert	AlertRcv(AlertingInd_02(TI_01))		
19		(TCV_Null := OO_HookOff())			
20		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
21		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
22		+CCstatuschk_02(TCV_chTch, C_U10, TI_02, TI_01)			

Detailed Comments: 1. Authentication fails, inconclusive.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU10_late(sub:B_2; setup:SETUP_MT_PDU; slot:SN; tsc:TSC; ta:TA; cksn:B_3; rand:RAND)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To establish a mobile terminating call for the supported bearer capability and put the MS under test in the CC state U10.

Default: OtherEvents

Comments: The supported bearer capability is specified in the input partameter setup, test case variable TCV_TI holds the transaction ID and the transaction ID =0. This is generic call set up procedure. The calling tree shall prepare three variables for the step: TCV_ch for SDCCH4 subchannel, TCV_chTch for traffic channel, TCV_CphKey for ciphering key. This test step is used for non RR testing.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+CCEstablishMT_SDCCH4(ta, sub, cksn)			
2		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(cksn, rand))		
3		L?DL_DatInAuthRes (TCV_Sres := DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
4		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, rand))			
5	L_1004	[NOT TCV_Res]		(I)	1.
6		+PostLinkRelEnd(TCV_ch)			
7		[TCV_Res]			
8		+Ciphering_on(TCV_ch)			
9		(TCV_Setup_mt.sig := Signal_01)			
10		L!DL_DatRqSetup (TCV_TI.ti_f := '0'B, TCV_TI.ti_v := '000'B)	SetupSnd(TCV_ch, setup)		
11		L?DL_DatInCallCo (TCV_CallCfm := DL_DatInCallCo.msg)	CallCfm(CallConfirm_01(TI_01))		
12		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
13		+localtree			
14		L?DL_DatInAlert	AlertRcv(AlertingInd_02(TI_01))		
15		(TCV_Null := OO_HookOff())			
16		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
17		+localtree			
		localtree			
18		+CCAssignTCH(slot, tsc)			
19		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
20		+CCstatuschk_02(TCV_chTch, C_U10, TI_02, TI_01)			

Detailed Comments: 1. Authentication fails, inconclusive.

Test Step Dynamic Behaviour

Test Step Name:	PreEnterCCstateU10_r01(Ta:TA; sub:B_2; powerlevelg,powerleveld:INTEGER; cksn:B_3; rand:RAND; slot:SN; tsc:TSC; tch:B_1; rate:IA5String)
Group:	GSM_L3_MS_v4220/Preambles/
Objective:	To establish a mobile terminating call for the supported bearer capability and put the MS in the CC state U10.
Default:	OtherEvents
Comments:	The supported bearer capability is specified in the input partameter setup, test case variable TCV_TI holds the transaction ID and the transaction ID =0. This is generic call set up procedure. The calling tree shall prepare three variables for the step: TCV_ch for SDCCH4 subchannel, TCV_chTch for traffic channel, TCV_CphKey for ciphering key. The timing advance is parameter. For RR testing.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
2		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
3		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
4		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, sub, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, Ta))		
5		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
6		ACTIVATE(OtherEvents)			Restore Normal default
7		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(cksn, rand))		
8		L?DL_DatInAuthRes (TCV_Sres := DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
9		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, rand))			
10	L_1005	[NOT TCV_Res]		I	1.
11		[TCV_Res]			
12		+Ciphering_on(TCV_ch)			
13		+localtree			
		localtree			
14		(TCV_Setup_mt.sig := Signal_01)			
15		L!DL_DatRqSetup (TCV_TI.ti_f := '0'B, TCV_TI.ti_v := '000'B)	SetupSnd(TCV_ch, TCV_Setup_mt)		
16		L?DL_DatInCallCo (TCV_CallCfm:=DL_DatInCallCo.msg)	CallCfm(CallConfirm_01(TI_01))		
17		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
18		+localtree1			
19		L?DL_DatInAlert	AlertRcv(AlertingInd_02(TI_01))		
20		(TCV_Null := OO_HookOff())			
21		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
22		+localtree1			
		localtree1			
23		+AssCmdGen(TCV_cellid, rate, slot, tsc, tch)			
24		+Adjust_gsmanddcs_powerlvl(powerlevelg, powerleveld, TCV_AssCmd)			
25		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
26		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		

27		+CCstatuschk_02(TCV_chTch, C_U10, TI_02, TI_01)			
Detailed Comments:		1. Authentication fails, inconclusive.			

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU10_21(sub:B_2; slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U10 by procedure in table 26.8.1.2/1. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body. The calling tree shall preare two variables for the step: TCV_ch for the SDCCH4 subchannel; TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU4_21(sub, ta)			
2		+CCAssignTCH(slot, tsc)			1.
3		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
4		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
5		+CCstatuschk_02(TCV_chTch, C_U10, TCV_TI, TCV_TI0)			
Detailed Comments:		1. To assign the suitable traffic channel to the MS.			

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU10_22(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U10 by procedure in table 26.8.1.2/2. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU4_22(slot, tsc, ta)			
2		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
3		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
4		+CCstatuschk_02(TCV_chTch, C_U10, TCV_TI, TCV_TI0)			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU11_23(sub:B_2; slot:SN; tsc:TSC; ta:TA; cksn:B_3; rand:RAND)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To bring the MS into CC state U11 by procedure in table 26.8.1.2/3. This is used in CC testing.

Default: OtherEvents

Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body. The calling tree shall preare two variables for the step: TCV_ch for the SDCCH4 subchannel; TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU4_23(sub, slot, tsc, ta, cksn, rand)			
2		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
3		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
4		(TCV_Null := OO_TermCall())			
5		L?DL_DatInDisc (TCV_Cau0 := DL_DatInDisc.msg.cau, TCV_Fn := DL_DatInDisc.fn)	DiscRcv(TCV_chTch, DisconnR(TCV_TI0, Cause_Def))		
6		+CCstatuschk_02(TCV_chTch, C_U11, TCV_TI, TCV_TI0)			

Detailed Comments:
1. To assign SDCCH4 channel.
2. Full rate channel needed, to setup a physical channel as full rate traffic channel.
3. Half rate channel needed, to setup a physical channel as half rate traffic channel.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU11_23Timer(sub:B_2; slot:SN; tsc:TSC; ta:TA; cksn:B_3; rand:RAND)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To bring the MS into CC state U11 by procedure in table 26.8.1.2/3. This is used in CC testing.

Default: OtherEvents

Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body. The calling tree shall preare two variables for the step: TCV_ch for the SDCCH4 subchannel; TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU4_23(sub, slot, tsc, ta, cksn, rand)			
2		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
3		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
4		(TCV_Null := OO_TermCall())			
5		L?DL_DatInDisc (TCV_Cau0 := DL_DatInDisc.msg.cau, TCV_Fn := DL_DatInDisc.fn) START T_dly(45000)	DiscRcv(TCV_chTch, DisconnR(TCV_TI0, Cause_Def))		
6		+CCstatuschk_02(TCV_chTch, C_U11, TCV_TI, TCV_TI0)			

Detailed Comments:
1. To assign SDCCH4 channel.
2. Full rate channel needed, to setup a physical channel as full rate traffic channel.
3. Half rate channel needed, to setup a physical channel as half rate traffic channel.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU11_24(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U11 by procedure in table 26.8.1.2/4. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU4_24(slot, tsc, ta)			
2		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
3		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
4		(TCV_Null := OO_TermCall())			
5		L?DL_DatInDisc	DiscRcv(TCV_chTch, DisconnR(TCV_TI0, Cause_Def))		
6		+CCstatuschk_02(TCV_chTch, C_U11, TCV_TI, TCV_TI0)			1.

Detailed Comments: 1. Now in CC state U11 and cause = #30.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU12_21(sub:B_2; slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U12 by procedure in table 26.8.1.2/1. This is used in CC testing.
Default: OtherEvents
Comments: The DISCONNECT message containing progress indicator #8. TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body. The calling tree shall preare two variables for the step: TCV_ch for the SDCCH4 subchannel; TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU10_21(sub, slot, tsc, ta)			
2		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_03iei, UuInfo_omit))		1.
3		+CCstatuschk_02(TCV_chTch, C_U12, TCV_TI, TCV_TI0)			

Detailed Comments: 1. Progress indicator = #8.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU12_22(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U12 by procedure in table 26.8.1.2/2. This is used in CC testing.
Default: OtherEvents
Comments: The DISCONNECT message containing progress indicator #8. TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU10_22(slot, tsc, ta)			
2		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_03iei, UuInfo_omit))		
3		+CCstatuschk_02(TCV_chTch, C_U12, TCV_TI, TCV_TI0)			1.

Detailed Comments: 1. Now in CC state U12

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU12_23(sub:B_2; slot:SN; tsc:TSC; ta:TA; cksn:B_3; rand:RAND)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To bring the MS into CC state U12 by procedure in table 26.8.1.2/3. This is used in CC testing.

Default: OtherEvents

Comments: The DISCONNECT message containing progress indicator #8. TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body. The calling tree shall preare two variables for the step: TCV_ch for the SDCCH4 subchannel; TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU4_23(sub, slot, tsc, ta, cksn, rand)			
2		L!DL_DatRqConn	ConnSnd(TCV_chTch , Connect_02(TCV_TI))		
3		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
4		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_03iei, UuInfo_omit))		1.
5		+CCstatuschk_02(TCV_chTch, C_U12, TCV_TI, TCV_TI0)			

Detailed Comments: 1. Progress indicator = #8.

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU19_21(sub:B_2; slot:SN; tsc:TSC; ta:TA)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To bring the MS into CC state U19 by procedure in table 26.8.1.2/1.

Default: OtherEvents

Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body. The calling tree shall preare two variables for the step: TCV_ch for the SDCCH4 subchannel; TCV_CphKey for ciphering key.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU10_21(sub, slot, tsc, ta)			
2		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
3		L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))		
4		+CCstatuschk_02(TCV_chTch, C_U19, TCV_TI, TCV_TI0)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU19_24(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U19 by procedure in table 26.8.1.2/4. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU4_24(slot, tsc, ta)			
2		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
3		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
4		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
5		L?DL_DatInRel (TCV_Fn := DL_DatInRel.fn)	ReleaseRcv(Release_10(TCV_TI0))		
6		+CCstatuschk_02(TCV_chTch, C_U19, TCV_TI, TCV_TI0)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEnterCCstateU19_24Timer(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To bring the MS into CC state U19 by procedure in table 26.8.1.2/4. This is used in CC testing.
Default: OtherEvents
Comments: TCV_TI0 contains the transcation identifier from the MS, and TCV_TI contains the transcation identifier for test system to send CC message. The values of them are used in test body.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEnterCCstateU4_24(slot, tsc, ta)			
2		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
3		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
4		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
5		L?DL_DatInRel (TCV_Fn := DL_DatInRel.fn) START T_dly(45000)	ReleaseRcv(Release_10(TCV_TI0))		
6		+CCstatuschk_02(TCV_chTch, C_U19, TCV_TI, TCV_TI0)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEstRRConn(slot:SN; tsc:TSC; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To establish a RR connection on SDCCH subchannel defined by TSPX_SDCCH4SubDef in cell A
Default: OtherEvents
Comments: The calling tree shall prepare variable for the step: TCV_ch for the SDCCH4 subchannel TSPX_SDCCH4SubDef. This test step is used for non RR testing.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
2		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
3		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
4		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, slot, tsc, TCV_chdescr_arfcn, ta))		
5		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
6		ACTIVATE(OtherEvents)			Restore Normal default

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEstRRC_MM(par_mi:MI; cksn:BITSTRING; ta:TA)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To establish a RR connection on C_SDCCH4_A_1
Default: OtherEvents
Comments: used var's:
TCV_Rr, TCV_Fn, TCV_Pgch, TCV_ia_ts, TCV_chdescr_arfcn

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_30(par_mi))		
2		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
3		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
4		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
5	L_1006	L?DL_EstInPgRes	PagingRes(PagingRes_30(par_mi, cksn))	(P)	
6		ACTIVATE(OtherEvents)			Restore Normal default

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreModifySetup(sub:B_2; slot:SN; tsc: TSC; ta:TA; cksn:B_3; rand:RAND)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To setup dual mode call and initiate MO incall modification. This is used in CC test group 26.8.1.4.5.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+InitCall(TCV_Service)			1.
2		+CCEstablishMO_SDCCH4(sub, ta)			
3		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
4		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_04)		
5		ACTIVATE(OtherEvents)			Restore Normal default
6		+ltree_continue			
		ltree_continue			
7		+Ciphering_on(TCV_ch)			
8		+RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
9		+Authentication(TCV_ch, cksn, rand)			
10		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
11		+CCAssignTCH(slot, tsc)			
12		L!DL_DatRqAlert	AlertSnd(TCV_chTch, Alerting_01(TCV_TI))		
13		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
14		L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))		
15		+InCallModi1(TSPX_MO_DualModSvc)			2.
16		L?DL_DatInModify (TCV_Fn := DL_DatInModify.fn, TCV_Modify := DL_DatInModify.msg)	ModifyRcv(ModifyInd_01(TCV_TI0, TCV_Bcap2))		3.

Detailed Comments:
1. To attempt a dual mode call.
2. MMI action to initiate in-call modification.
3. The expected MODIFY message received.

Test Step Dynamic Behaviour

Test Step Name: SpeechService
Group: GSM_L3_MS_v4220/Preambles/
Objective: To selecte a speech service, if the MS supports Telephony the selected speech service is telephony.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_Serv_TS11]			
2		+BasicServiceMO(C_Telephony, TSPX_Telephony_Rate)			
3		[TSPC_Serv_TS61]			
4		+BasicServiceMO(C_AltSpchFax, C_Full)			
5		[TSPC_Serv_BS61]			
6		+BasicServiceMO(C_AltSpchData, C_Full)			
7		[TSPC_Serv_BS81]			
8		+BasicServiceMO(C_SpchData, C_Full)			
9	L_1007	[C_Yes]		I	1.

Detailed Comments:
1. No speech service available

Test Step Dynamic Behaviour					
Test Step Name:	StartCellA(par_bspwr:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING; cchdgsm, cchddcs:CCHD; bcchflgsm, bcchfldcs:NCD; Re:B_1)				
Group:	GSM_L3_MS_v4220/Preambles/				
Objective:	To setup a physical channel as combined BCCH, CCCH and SDCCH4 then broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with default parameters of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_sacch:= OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
2		+CombinedBCCH_A(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, C_arfcnA, C_arfcnAd, acttype, slot, tsc, ta, babr, cch_con, bpm)			
3		+SysInfoSending_fh(C_SCH_A, C_BCCH_A_1, t, retr, neci, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lac, CellOpt_01, cchdgsm, cchddcs, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchflgsm, bcchfldcs, Re, C_BCC, C_NCC, C_NCCP_2)			
4		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
5		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	StartCellA_1(par_bspwr:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING; bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs:NCD; Re:B_1)				
Group:	GSM_L3_MS_v4220/Preambles/				
Objective:	To set up a physical channel with parameters different from default and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A, then broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with default parameters of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
2		+CombinedBCCH_A(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, C_arfcnA, C_arfcnAd, acttype, slot, tsc, ta, babr, cch_con, bpm)			
3		+SysInfoSending_nfh(C_SCH_A, C_BCCH_A_1, t, retr, neci, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lac, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs, Re, C_BCC, C_NCC_1, C_NCCP_2)			1.
4		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
5		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: StartCellB(par_bspwr, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING; cchdgsm, cchddcs:CCHD; bcchflgsm, bcchfldcs:NCD; Re:BITSTRING; bcc:BCC; ncc:NCC)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To setup a physical channel as combined BCCH, CCCH and SDCCH4 then broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with default parameters of cell B.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_sacch_B := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
2		+ltree_chconf			
3		+SysInfoSending_fh(C_SCH_B, C_BCCH_B_1, t, retr, neci, att, babr, cch_con, bpm, t3212, C_ci_cellB, mcc, mnc, lac, CellOpt_01, cchdgsm, cchddcs, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchflgsm, bcchfldcs, Re, bcc, ncc, C_NCCP_2)			
4		(TCV_Ccd0B := TCV_Ccd0H, TCV_sysinfo5_B := TCV_sysinf5, TCV_sysinfo6_B := TCV_sysinf6)			
5		[cch_con = C_cch_1Comb]			
6		+SysInfo_SacchSending(TCV_sacch_B, TCV_sysinfo5_B, TCV_sysinfo6_B)			
7		[cch_con = C_cch_1nonComb]			
8	L_1008	[(cch_con <> C_cch_1nonComb) AND(cch_con <> C_cch_1Comb)]		I	
		ltree_chconf			
9		[cch_con = C_cch_1Comb]			
10		+CombinedBCCH_B(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, arfcn1, arfcn2, acttype, slot, tsc, ta, babr, cch_con, bpm)			
11		[cch_con = C_cch_1nonComb]			
12		+NonCombinedBCCH_B(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, arfcn1, arfcn2, acttype, slot, tsc, ta, babr, cch_con, bpm)			
13	L_1009	[(cch_con <> C_cch_1nonComb) AND(cch_con <> C_cch_1Comb)]		I	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name:	StartCellB_1(par_bspwr:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING; Re:BITSTRING; bcc:BCC; ncc:NCC)
Group:	GSM_L3_MS_v4220/Preambles/
Objective:	To set up a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B, then start transmission of default system information's for cell B.
Default:	OtherEvents
Comments:	The location area code is different from the cell A.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_sacch_B := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
2		+CombinedBCCH_B(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, C_arfcnB, C_arfcnBd, acttype, slot, tsc, ta, babr, cch_con, bpm)			
3		+SysInfoSending_nfh(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellB, mcc, mnc, lac, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_01, BcchFreqLst_48, BcchFreqLst_48, Re, bcc, ncc, C_NCCP_2)			1.
4		(TCV_Ccd0B := TCV_Ccd0H, TCV_sysinfo5_B := TCV_sysinf5, TCV_sysinfo6_B := TCV_sysinf6)			
5		+SysInfo_SacchSending(TCV_sacch_B, TCV_sysinfo5_B, TCV_sysinfo6_B)			

Detailed Comments: 1. To send SYSTEM INFORMATION message with default parameter for cell B.

Test Step Dynamic Behaviour

Test Step Name: StartCellB_egsm(par_bspwr, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING; cchdgsm:CCHD; bcchfl1, bcchfl2:NCD; Re:BITSTRING; bcc:BCC; ncc:NCC)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To setup a physical channel as combined BCCH, CCCH and SDCCH4 then broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with EGSM default parameters of cell B.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_sacch_B := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
2		+ltree_chconf			
3		+SysInfoSending_e(C_SCH_B, C_BCCH_B_1, t, retr, neci, att, babr, cch_con, bpm, t3212, C_ci_cellB, mcc, mnc, lac, CellOpt_01, cchdgsm, C_CellReselectHys12, C_MaxPwrLvlG, bcchfl1, bcchfl2, Re, bcc, ncc, C_NCCP_2)			
4		(TCV_Ccd0B := TCV_Ccd0H, TCV_sysinfo5_B := TCV_sysinf5, TCV_sysinfo5bis_B := TCV_sysinfo5bis, TCV_sysinfo6_B := TCV_sysinf6)			
5		[cch_con = C_cch_1Comb]			
6		+SysInfo_SacchSending(TCV_sacch_B, TCV_sysinfo5_B, TCV_sysinfo6_B)			
7		[cch_con = C_cch_1nonComb]			
8	L_1010	[(cch_con <> C_cch_1nonComb) AND(cch_con <> C_cch_1Comb)] **ltree_chconf**		I	
9		[cch_con = C_cch_1Comb]			
10		+CombinedBCCH_B(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, arfcn1, arfcn2, acttype, slot, tsc, ta, babr, cch_con, bpm)			
11		[cch_con = C_cch_1nonComb]			
12		+NonCombinedBCCH_B(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, arfcn1, arfcn2, acttype, slot, tsc, ta, babr, cch_con, bpm)			
13	L_1011	[(cch_con <> C_cch_1nonComb) AND(cch_con <> C_cch_1Comb)]		I	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: StartCellB_ho(par_bspwr, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING; cchdgsm, cchddcs:CCHD; bcchflgsm, bcchfldcs:NCD; Re:BITSTRING; bcc:BCC; ncc:NCC; td, fn:INTEGER)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To setup a physical channel as combined BCCH, CCCH and SDCCH4 then broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with default parameters of cell B.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_sacch_B := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
2		+ltree_chconf			
3		+SysInfoSending_fh(C_SCH_B, C_BCCH_B_1, t, retr, neci, att, babr, cch_con, bpm, t3212, C_ci_cellB, mcc, mnc, lac, CellOpt_01, cchdgsm, cchddcs, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchflgsm, bcchfldcs, Re, bcc, ncc, C_NCCP_2)			
4		(TCV_Ccd0B := TCV_Ccd0H, TCV_sysinfo5_B := TCV_sysinf5, TCV_sysinfo6_B := TCV_sysinf6)			
5		[cch_con = C_cch_1Comb]			
6		+SysInfo_SacchSending(TCV_sacch_B, TCV_sysinfo5_B, TCV_sysinfo6_B)			
7		[cch_con = C_cch_1nonComb]			
8	L_1012	[(cch_con <> C_cch_1nonComb) AND(cch_con <> C_cch_1Comb)]		I	
		ltree_chconf			
9		[cch_con = C_cch_1Comb]			
10		+CombinedBCCH_B_ho(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, arfcn1, arfcn2, acttype, slot, tsc, ta, babr, cch_con, bpm, td, fn)			
11		[cch_con = C_cch_1nonComb]			
12		+NonCombinedBCCH_B_ho(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, arfcn1, arfcn2, acttype, slot, tsc, ta, babr, cch_con, bpm, td, fn)			
13	L_1013	[(cch_con <> C_cch_1nonComb) AND(cch_con <> C_cch_1Comb)]		I	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: StartCellC(par_bspwr:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING; Re:BITSTRING; bcc:BCC; ncc:NCC)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To setup a physical channel as combined BCCH, CCCH and SDCCH4 then broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with default parameters of cell C.

Default: OtherEvents

Comments: IMSI attach/detach not allowed (ATT=0). Cell C belongs to PLMN1.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_sacch_C := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellC))			
2		+CombinedBCCH_C(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, C_arfcnC, C_arfcnCd, acttype, slot, tsc, ta, babr, cch_con, bpm)			
3		+SysInfoSending_nfh(C_SCH_C, C_BCCH_C_1, 5, 1, 0, att, babr, cch_con, bpm, t3212, C_ci_cellC, mcc, mnc, lac, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_07, BcchFreqLst_07, BcchFreqLst_15, BcchFreqLst_15, Re, bcc, ncc, C_NCCP_2)			1.
4		(TCV_Ccd0C := TCV_Ccd0H, TCV_sysinfo5_C := TCV_sysinf5, TCV_sysinfo6_C := TCV_sysinf6)			
5		+SysInfo_SacchSending(TCV_sacch_C, TCV_sysinfo5_C, TCV_sysinfo6_C)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Start_2cells(acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lacA, lacB:OCTETSTRING)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To start cell A and cell B with default parameters except power level of cell A and power level of cell B and to bring the MS in Idle updated state on given cell.

A and B are from different location areas belonging to PLMN2. PLMN2 is different from HPLMN.

IMSI attach detach is allowed in both cells.

T3212 value is 1/10 in both cells.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+StartCellB(C_E_default, C_arfcnB, C_arfcnBd, acttype, slot, tsc, ta, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, mcc, mnc, lacB, CellChDes_04, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC)			
2		+StartCellA(C_E_neighbourdefault, acttype, slot, tsc, ta, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, mcc, mnc, lacA, CellChDes_02, CellChDes_03, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name:	StartMultiCells_01(acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc2, mnc2, mcc3, mnc3, mcc4, mnc4, mcc5, mnc5, mcc6, mnc6, mcc7, mnc7, mcc8, mnc8, lac:OCTETSTRING)
Group:	GSM_L3_MS_v4220/Preambles/
Objective:	To setup 8 or 7 physical channels representing different cells then broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 in multiple cells for idle mode testing.
Default:	OtherEvents
Comments:	The parameters for SYSTEM INFORMATION messages are defined in section 26.3.1 of GSM 11.10

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_sacch_B := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB), TCV_sacch_C := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellC), TCV_sacch_D := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellD), TCV_sacch_E := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellE), TCV_sacch_F := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellF), TCV_sacch_G := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellG), TCV_sacch_H := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellH))			
2		+cell2			
3		+cell3			
4		+cell4			
5		+cell5			
6		+cell6			
7		[TSPC_PGSM OR TSPC_EGSM]			
8		+cell7g			
9		+cell8			
10		[TSPC_DCS]			
11		+cell7d			
		cell2			
12		+CombinedBCCH_B(63, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, 7, 580, acttype, slot, tsc, ta, babr, cch_con, bpm)			
13		+SysInfoSending_nfh(C_SCH_B, C_BCCH_B_1, C_TxInt_10, C_Max_2, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellB, mcc2, mnc2, lac, CellOpt_02, C_CellReselectHys4, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, BcchFreqLst_06, BcchFreqLst_06, BcchFreqLst_14, BcchFreqLst_14, C_noRestablishment, C_BCC, C_NCC_1, C_NCCP_2)			
14		(TCV_Ccd0B := TCV_Ccd0H, TCV_sysinfo5_B := TCV_sysinf5, TCV_sysinfo6_B := TCV_sysinf6)			
15		+SysInfo_SacchSending(TCV_sacch_B, TCV_sysinfo5_B, TCV_sysinfo6_B)			
		cell3			
16		+CombinedBCCH_C(61, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, 39, 610, acttype, slot, tsc, ta, babr, cch_con, bpm)			
17		+SysInfoSending_nfh(C_SCH_C, C_BCCH_C_1, C_TxInt_10, C_Max_2, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellC, mcc3, mnc3, lac, CellOpt_02, C_CellReselectHys4, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, BcchFreqLst_07, BcchFreqLst_07, BcchFreqLst_15, BcchFreqLst_15, C_noRestablishment, C_BCC, C_NCC_2, C_NCCP_2)			
18		(TCV_Ccd0C := TCV_Ccd0H, TCV_sysinfo5_C := TCV_sysinf5, TCV_sysinfo6_C := TCV_sysinf6)			
19		+SysInfo_SacchSending(TCV_sacch_C, TCV_sysinfo5_C, TCV_sysinfo6_C)			
		cell4			

20		+CombinedBCCH_D(55, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, 65, 702, acttype, slot, tsc, ta, babr, cch_con, bpm)			
21		+SysInfoSending_nfh(C_SCH_D, C_BCCH_D_1, C_TxInt_10, C_Max_2, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellD, mcc4, mnc4, lac, CellOpt_02, C_CellReselectHys4, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, BcchFreqLst_08, BcchFreqLst_08, BcchFreqLst_16, BcchFreqLst_16, C_noRestablishment, C_BCC, C_NCC_3, C_NCCP_2)			
22		(TCV_Ccd0D := TCV_Ccd0H, TCV_sysinfo5_D := TCV_sysinf5, TCV_sysinfo6_D := TCV_sysinf6)			
23		+SysInfo_SacchSending(TCV_sacch_D, TCV_sysinfo5_D, TCV_sysinfo6_D)			
		cell5			
24		+CombinedBCCH_E(59, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, 66, 703, acttype, slot, tsc, ta, babr, cch_con, bpm)			
25		+SysInfoSending_nfh(C_SCH_E, C_BCCH_E_1, C_TxInt_10, C_Max_2, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellE, mcc5, mnc5, lac, CellOpt_02, C_CellReselectHys4, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, BcchFreqLst_09, BcchFreqLst_09, BcchFreqLst_17, BcchFreqLst_17, C_noRestablishment, C_BCC, C_NCC_4, C_NCCP_2)			
26		(TCV_Ccd0E := TCV_Ccd0H, TCV_sysinfo5_E := TCV_sysinf5, TCV_sysinfo6_E := TCV_sysinf6)			
27		+SysInfo_SacchSending(TCV_sacch_E, TCV_sysinfo5_E, TCV_sysinfo6_E)			
		cell6			
28		+CombinedBCCH_F(57, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, 85, 830, acttype, slot, tsc, ta, babr, cch_con, bpm)			
29		+SysInfoSending_nfh(C_SCH_F, C_BCCH_F_1, C_TxInt_10, C_Max_2, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellF, mcc6, mnc6, lac, CellOpt_02, C_CellReselectHys4, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, BcchFreqLst_10, BcchFreqLst_10, BcchFreqLst_18, BcchFreqLst_18, C_noRestablishment, C_BCC, C_NCC_5, C_NCCP_2)			
30		(TCV_Ccd0F := TCV_Ccd0H, TCV_sysinfo5_F := TCV_sysinf5, TCV_sysinfo6_F := TCV_sysinf6)			
31		+SysInfo_SacchSending(TCV_sacch_F, TCV_sysinfo5_F, TCV_sysinfo6_F)			
		cell7g			
32		+CombinedBCCH_G_sp(55, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, 97, 985, 885, acttype, slot, tsc, ta, babr, cch_con, bpm)			
33		+SysInfoSending_nfh(C_SCH_G, C_BCCH_G_1, C_TxInt_10, C_Max_2, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellG, mcc7, mnc7, lac, CellOpt_02, C_CellReselectHys4, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, BcchFreqLst_11, BcchFreqLst_11, BcchFreqLst_19, BcchFreqLst_19, C_noRestablishment, C_BCC, C_NCC_6, C_NCCP_2)			
34		(TCV_Ccd0G := TCV_Ccd0H, TCV_sysinfo5_G := TCV_sysinf5, TCV_sysinfo6_G := TCV_sysinf6)			
35		+SysInfo_SacchSending(TCV_sacch_G, TCV_sysinfo5_G, TCV_sysinfo6_G)			

		cell7d			
36		+CombinedBCCH_G_sp(55, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, 97, 985, 885, acttype, slot, tsc, ta, babr, cch_con, bpm)			
37		+SysInfoSending_nfhCb(C_SCH_G, C_BCCH_G_1, C_TxInt_10, C_Max_2, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellG, mcc8, mnc8, lac, CellOpt_02, C_CellReselectHys4, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, BcchFreqLst_11, BcchFreqLst_11, BcchFreqLst_19, BcchFreqLst_19, C_noRestablishment, C_BCC, C_NCC_6, C_NCCP_2)			
38		(TCV_Ccd0G := TCV_Ccd0H, TCV_sysinfo5_G := TCV_sysinf5, TCV_sysinfo6_G := TCV_sysinf6)			
39		+SysInfo_SacchSending(TCV_sacch_G, TCV_sysinfo5_G, TCV_sysinfo6_G)			
		cell8			
40		+CombinedBCCH_H(53, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, 124, 124, acttype, slot, tsc, ta, babr, cch_con, bpm)			
41		+SysInfoSending_nfh(C_SCH_H, C_BCCH_H_1, C_TxInt_10, C_Max_2, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellH, mcc8, mnc8, lac, CellOpt_02, C_CellReselectHys4, TSPX_MSTxpwrMax, TSPX_MSTxpwrMax, BcchFreqLst_12, BcchFreqLst_12, BcchFreqLst_12, BcchFreqLst_12, C_noRestablishment, C_BCC, C_NCC_7, C_NCCP_2)			
42		(TCV_sysinfo5_H := TCV_sysinf5, TCV_sysinfo6_H := TCV_sysinf6)			
43		+SysInfo_SacchSending(TCV_sacch_H, TCV_sysinfo5_H, TCV_sysinfo6_H)			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: StartMultiCells_02(bcchfl_gsm, bcchfl_dcs:NCD; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lacB, lacC, lacD, lacE, lacF, lacG, lacH:OCTETSTRING)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To broadcast SYSTEM INFORMATION messages 1, 2, 3, 4, 5 and 6 in multiple cells for measurement testing. Neighbour cells description for cell S1 is a formal parameter.

Default: OtherEvents

Comments: The parameters for SYSTEM INFORMATION messages are defined in section 26.6.3 of GSM 11.10

Nr	Label	Behaviour Description	CRef	V	Comments
1		+cellN1			
2		+cellN2			
3		+cellN3			
4		+cellN4			
5		+cellN5			
6		+cellN6			
7		+cellN7			
		cellN1			
8		+CombinedBCCH_B(28, C_MaxPwrLvlG, C_MaxPwrLvlD, 8, 530, acttype, slot, tsc, ta, babr, cch_con, bpm)			
9		+SysInfoSending_fh(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellB, mcc, mnc, lacB, CellOpt_01, CellChDes_05, CellChDes_05d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			
10		(TCV_Ccd0B := TCV_Ccd0H)			
		cellN2			
11		+CombinedBCCH_C(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 14, 602, acttype, slot, tsc, ta, babr, cch_con, bpm)			
12		+SysInfoSending_fh(C_SCH_C, C_BCCH_C_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellC, mcc, mnc, lacC, CellOpt_01, CellChDes_06, CellChDes_06d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC_7, C_NCC, C_NCCP_2)			
13		(TCV_Ccd0C := TCV_Ccd0H)			
		cellN3			
14		+CombinedBCCH_D(38, C_MaxPwrLvlG, C_MaxPwrLvlD, 20, 665, acttype, slot, tsc, ta, babr, cch_con, bpm)			
15		+SysInfoSending_fh(C_SCH_D, C_BCCH_D_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellD, mcc, mnc, lacD, CellOpt_01, CellChDes_07, CellChDes_07d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC_1, C_NCC, C_NCCP_2)			ncc = 1
16		(TCV_Ccd0D := TCV_Ccd0H)			
		cellN4			
17		+CombinedBCCH_E(58, C_MaxPwrLvlG, C_MaxPwrLvlD, 26, 762, acttype, slot, tsc, ta, babr, cch_con, bpm)			
18		+SysInfoSending_fh(C_SCH_E, C_BCCH_E_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellE, mcc, mnc, lacE, CellOpt_01, CellChDes_08, CellChDes_08d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs,			ncc = 1

		C_noRestablishment, C_BCC_3, C_NCC, C_NCCP_2)			
19		(TCV_Ccd0E := TCV_Ccd0H)			
		cellN5			
20		+CombinedBCCH_F(63, C_MaxPwrLvlG, C_MaxPwrLvlD, 32, 686, acttype, slot, tsc, ta, babr, cch_con, bpm)			
21		+SysInfoSending_fh(C_SCH_F, C_BCCH_F_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellF, mcc, mnc, lacF, CellOpt_01, CellChDes_09, CellChDes_09d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			ncc = 1
22		(TCV_Ccd0F := TCV_Ccd0H)			
		cellN6			
23		+CombinedBCCH_G(68, C_MaxPwrLvlG, C_MaxPwrLvlD, 38, 549, acttype, slot, tsc, ta, babr, cch_con, bpm)			
24		+SysInfoSending_fh(C_SCH_G, C_BCCH_G_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellG, mcc, mnc, lacG, CellOpt_01, CellChDes_10, CellChDes_10d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC_7, C_NCC, C_NCCP_2)			
25		(TCV_Ccd0G := TCV_Ccd0H)			
		cellN7			
26		+CombinedBCCH_H(73, C_MaxPwrLvlG, C_MaxPwrLvlD, 44, 810, acttype, slot, tsc, ta, babr, cch_con, bpm)			
27		+SysInfoSending_fh(C_SCH_H, C_BCCH_H_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellH, mcc, mnc, lacH, CellOpt_01, CellChDes_11, CellChDes_11d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC_1, C_NCC, C_NCCP_2)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: StartMultiCells_02e(bcchfl_gsm, bcchfl_dcs:NCD; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lacA, lacB, lacC, lacD, lacE, lacF, lacG, lacH:OCTETSTRING)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To broadcast SYSTEM INFORMATION messages 1, 2, 3, 4, 5 and 6 in multiple cells for measurement testing. Neighbour cells description for cell S1 is a formal parameter.

Default: OtherEvents

Comments: The parameters for SYSTEM INFORMATION messages are defined in section 26.10.2 of GSM 11.10

Nr	Label	Behaviour Description	CRef	V	Comments
1		+cellN1			
2		+cellN2			
3		+cellN3			
4		+cellN4			
5		+cellN5			
6		+cellN6			
7		+cellN7			
		cellN1			
8		+CombinedBCCH_B(28, C_MaxPwrLvlG, C_MaxPwrLvlD, C_arfcnEgsm, C_arfcnEgsm, acttype, slot, tsc, ta, babr, cch_con, bpm)			
9		+SysInfoSending_fh(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellB, mcc, mnc, lacB, CellOpt_01, CellChDes_05, CellChDes_05d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			
10		(TCV_Ccd0B := TCV_Ccd0H)			
		cellN2			
11		+CombinedBCCH_C(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 1005, 1005, acttype, slot, tsc, ta, babr, cch_con, bpm)			
12		+SysInfoSending_fh(C_SCH_C, C_BCCH_C_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellC, mcc, mnc, lacC, CellOpt_01, CellChDes_06, CellChDes_06d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC_7, C_NCC, C_NCCP_2)			
13		(TCV_Ccd0C := TCV_Ccd0H)			
		cellN3			
14		+CombinedBCCH_D(38, C_MaxPwrLvlG, C_MaxPwrLvlD, 0, 0, acttype, slot, tsc, ta, babr, cch_con, bpm)			
15		+SysInfoSending_fh(C_SCH_D, C_BCCH_D_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellD, mcc, mnc, lacD, CellOpt_01, CellChDes_07, CellChDes_07d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC_1, C_NCC, C_NCCP_2)			ncc = 1
16		(TCV_Ccd0D := TCV_Ccd0H)			
		cellN4			
17		+CombinedBCCH_E(58, C_MaxPwrLvlG, C_MaxPwrLvlD, 26, 26, acttype, slot, tsc, ta, babr, cch_con, bpm)			
18		+SysInfoSending_fh(C_SCH_E, C_BCCH_E_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellE, mcc, mnc, lacE, CellOpt_01, CellChDes_08, CellChDes_08d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs,			ncc = 1

		C_noRestablishment, C_BCC_3, C_NCC, C_NCCP_2)			
19		(TCV_Ccd0E := TCV_Ccd0H)			
		cellN5			
20		+CombinedBCCH_F(63, C_MaxPwrLvlG, C_MaxPwrLvlD, 1020, 1020, acttype, slot, tsc, ta, babr, cch_con, bpm)			
21		+SysInfoSending_fh(C_SCH_F, C_BCCH_F_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellF, mcc, mnc, lacF, CellOpt_01, CellChDes_09, CellChDes_09d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			ncc = 1
22		(TCV_Ccd0F := TCV_Ccd0H)			
		cellN6			
23		+CombinedBCCH_G(68, C_MaxPwrLvlG, C_MaxPwrLvlD, 38, 38, acttype, slot, tsc, ta, babr, cch_con, bpm)			
24		+SysInfoSending_fh(C_SCH_G, C_BCCH_G_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellG, mcc, mnc, lacG, CellOpt_01, CellChDes_10, CellChDes_10d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC_7, C_NCC, C_NCCP_2)			
25		(TCV_Ccd0G := TCV_Ccd0H)			
		cellN7			
26		+CombinedBCCH_H(73, C_MaxPwrLvlG, C_MaxPwrLvlD, 1003, 1003, acttype, slot, tsc, ta, babr, cch_con, bpm)			
27		+SysInfoSending_fh(C_SCH_H, C_BCCH_H_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellH, mcc, mnc, lacH, CellOpt_01, CellChDes_11, CellChDes_11d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC_1, C_NCC, C_NCCP_2)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: StartMultiCells_03(bcchfl_gsm, bcchfl_dcs:NCD; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lacA, lacC, lacD, lacE, lacF, lacG, lacH:OCTETSTRING)

Group: GSM_L3_MS_v4220/Preambles/

Objective: To broadcast SYSTEM INFORMATION messages 1, 2, 3, 4, 5 and 6 in multiple cells for measurement testing. Neighbour cells description for cell S1 is a formal parameter.

Default: OtherEvents

Comments: The parameters for SYSTEM INFORMATION messages are defined in section 26.6.3 of GSM 11.10

Nr	Label	Behaviour Description	CRef	V	Comments
1		+cellS			
2		+cellN2			
3		+cellN3			
4		+cellN4			
5		+cellN5			
6		+cellN6			
7		+cellN1			
		cellS			
8		+CombinedBCCH_A(C_E_neighbourdefault, C_MaxPwrLvlG, C_MaxPwrLvlD, C_arfcn_2, C_arfcnAd_2, acttype, slot, tsc, ta, babr, cch_con, bpm)			
9		+SysInfoSending_fh(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lacA, CellOpt_03, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_35, BcchFreqLst_27, C_noRestablishment, C_BCC_3, C_NCC, C_NCCP_2)			
10		(TCV_Ccd0A := TCV_Ccd0H)			
		cellN2			
11		+CombinedBCCH_C(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 14, 602, acttype, slot, tsc, ta, babr, cch_con, bpm)			
12		+SysInfoSending_fh(C_SCH_C, C_BCCH_C_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellC, mcc, mnc, lacC, CellOpt_01, CellChDes_06, CellChDes_06d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC_7, C_NCC, C_NCCP_2)			
13		(TCV_Ccd0C := TCV_Ccd0H)			
		cellN3			
14		+CombinedBCCH_D(38, C_MaxPwrLvlG, C_MaxPwrLvlD, 20, 665, acttype, slot, tsc, ta, babr, cch_con, bpm)			
15		+SysInfoSending_fh(C_SCH_D, C_BCCH_D_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellD, mcc, mnc, lacD, CellOpt_01, CellChDes_07, CellChDes_07d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC_1, C_NCC, C_NCCP_2)			ncc = 1
16		(TCV_Ccd0D := TCV_Ccd0H)			
		cellN4			
17		+CombinedBCCH_E(58, C_MaxPwrLvlG, C_MaxPwrLvlD, 26, 762, acttype, slot, tsc, ta, babr, cch_con, bpm)			
18		+SysInfoSending_fh(C_SCH_E, C_BCCH_E_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellE, mcc, mnc, lacE, CellOpt_01, CellChDes_08, CellChDes_08d, C_CellReselectHys12, C_MaxPwrLvlG,			ncc = 1

		C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC_3, C_NCC, C_NCCP_2)			
19		(TCV_Ccd0E := TCV_Ccd0H)			
		cellN5			
20		+CombinedBCCH_F(63, C_MaxPwrLvlG, C_MaxPwrLvlD, 32, 686, acttype, slot, tsc, ta, babr, cch_con, bpm)			
21		+SysInfoSending_fh(C_SCH_F, C_BCCH_F_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellF, mcc, mnc, lacF, CellOpt_01, CellChDes_09, CellChDes_09d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			ncc = 1
22		(TCV_Ccd0F := TCV_Ccd0H)			
		cellN6			
23		+CombinedBCCH_G(68, C_MaxPwrLvlG, C_MaxPwrLvlD, 38, 549, acttype, slot, tsc, ta, babr, cch_con, bpm)			
24		+SysInfoSending_fh(C_SCH_G, C_BCCH_G_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellG, mcc, mnc, lacG, CellOpt_01, CellChDes_10, CellChDes_10d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC_7, C_NCC, C_NCCP_2)			
25		(TCV_Ccd0G := TCV_Ccd0H)			
		cellN1			
26		+CombinedBCCH_H(28, C_MaxPwrLvlG, C_MaxPwrLvlD, 8, 530, acttype, slot, tsc, ta, babr, cch_con, bpm)			
27		+SysInfoSending_fh(C_SCH_H, C_BCCH_H_1, C_TxInt_5, C_Max_1, C_NECI_0, att, babr, cch_con, bpm, t3212, C_ci_cellH, mcc, mnc, lacH, CellOpt_01, CellChDes_05, CellChDes_05d, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchfl_gsm, bcchfl_dcs, C_noRestablishment, C_BCC, C_NCC, C_NCCP_2)			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: TS61MO(srv:SERVICES)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To generate an MO SETUP message with appropriate Bcap for TS61 service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_srv := srv, TCV_ur := TSPX_TS_61_ur1, TCV_CE := C_BothT)			
2		+local_selectCe			
3		+local_bcap			get the right BC
4		[(NOT TSPC_EFR_Speech_v2) AND(NOT TSPC_EFR_Speech_v3)]			
5		(TCV_Setup_mo := Setup_11_3100(C_RI_alternate, Bcap_Speech_MO(C_Rchr_dontcare), TCV_BcapMO1, Llcmp_NotApplicable, Llcmp_AnyOrOmit, C_RI_alternate, C_RI_alternate, Hlcmp_NotApplicable, Hlcmp_TS61))			Non EFR Mobile - generate SETUP
6		+Chmod(TCV_ChRate, TCV_ce, TCV_ur, srv)			initiate channel mode variables
7		(TCV_Null := OO_IFsetup_TS6x_MO(TCV_srv, TCV_ur, TCV_ce, TCV_ChRate))			ask operator to configure the MS
8		[TSPC_EFR_Speech_v2 OR TSPC_EFR_Speech_v3]			
9		(TCV_Setup_mo := Setup_11_3100(C_RI_alternate, Bcap_Speech_Efr_MO(C_Rchr_dontcare), TCV_BcapMO1, Llcmp_NotApplicable, Llcmp_AnyOrOmit, C_RI_alternate, C_RI_alternate, Hlcmp_NotApplicable, Hlcmp_TS61))			EFR Mobile - generate SETUP
10		+Chmod(TCV_ChRate, TCV_ce, TCV_ur, srv)			initiate channel mode variables
11		(TCV_Null := OO_IFsetup_TS6x_MO(TCV_srv, TCV_ur, TCV_ce, TCV_ChRate))			ask operator to configure the MS
		local_bcap			
12		[TCV_ce = C_transparent]			
13		[TCV_ur = C_9600bs]			
14		+local_TS61_FAX3_T(C_9600bs, C_ir_16kbs)			
15		[(TCV_ur = C_4800bs) OR (TCV_ur = C_2400bs)]			
16		+local_TS61_FAX3_T(TCV_ur, C_ir_8kbs)			
17	L_1014	[(TCV_ur <> C_4800bs) AND(TCV_ur <> C_2400bs) AND(TCV_ur <> C_9600bs)]		I	IXIT parameter error
18		[TCV_ce = C_notransparent]			
19		+local_TS61_FAX3_NT(TCV_ur)			
		local_TS61_FAX3_T(ur:B_4; ir:B_2)			
20		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_FAX3, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, ir, C_parity_NA, C_transparent, C_modemt_none))			11.8.2.7.1
		local_TS61_FAX3_NT(ur:B_4)			
21		(TCV_bcapListIndicator := TCV_bcapListIndicator + 2)			
22		[ur=C_9600bs]			
23		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_FAX3, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, TCV_CE, C_modemt_none, C_X25_flct), TCV_bcapListIndicator))			11.8.2.7.2
24		[(ur=C_2400bs) OR (ur = C_4800bs)]			

25		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(TCV_Rchr, C_FAX3, C_SDUintegrity, C_nirr_dontcare, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, TCV_CE, C_modemt_none, C_X25_flct), TCV_bcapListIndicator))			11.8.2.7.2
26	L_1015	[(ur <> C_2400bs) AND(ur <> C_4800bs) AND(ur <> C_9600bs)]		I	IXIT parameter error
		local_selectCe			
27		[TSPX_TS_61_T_NT]			
28		(TCV_ce := C_notransparent)			
29		[NOT TSPX_TS_61_T_NT]			
30		(TCV_ce := TSPX_TS_61_ce)			

Detailed Comments:

The TS61 Bearer Capabilities and SETUP messages for MO is constructed according to section 11.8.2.7.1 and section 11.8.2.7.2 of GSM 11.10 and section B.1.10.2 and section B.2.10 of GSM 07.01.

The algorithm for selecting parameters:

```
IF (TSPX_BS_61_T_NT)
        TCV_ce := C_notransparent
ELSE
        TCV_ce := TSPX_BS_61_ce
```

use (TCV_ce, TSPX_BS_61_ur1, svc) for OO_IFsetup_TS6x_MO

Test Step Dynamic Behaviour

Test Step Name: TS62MO(srv:SERVICES)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To generate an MO SETUP message with appropriate Bcap for TS62 service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_srv := srv, TCV_ur := TSPX_TS_62_ur1, TCV_CE := C_BothT)			variables used in InitCall
2		+local_selectCe			
3		+local_bcap			get the right BC
4		(TCV_Setup_mo := Setup_10_3100(TCV_BcapMO1, Llcmp_AnyOrOmit, Hlcmp_TS62))			
5		+Chmod(TCV_ChRate, TCV_ce, TCV_ur, srv)			
6		(TCV_Null := OO_IFsetup_TS6x_MO(TCV_srv, TCV_ur, TCV_ce, TCV_ChRate))			
		local_bcap			
7		[TCV_ce = C_transparent]			
8		[TCV_ur = C_9600bs]			
9		+local_TS62_FAX3_T(C_9600bs, C_ir_16kbs)			
10		[(TCV_ur = C_4800bs) OR (TCV_ur = C_2400bs)]			
11		+local_TS62_FAX3_T(TCV_ur, C_ir_8kbs)			
12	L_1016	[(TCV_ur <> C_4800bs) AND(TCV_ur <> C_2400bs) AND(TCV_ur <> C_9600bs)]		I	IXIT parameter error
13		[TCV_ce = C_notransparent]			
14		+local_TS62_FAX3_NT(TCV_ur)			
		local_TS62_FAX3_T(ur:B_4; ir:B_2)			
15		(TCV_BcapMO1 := Bcap_MO(C_Rchr_dontcare, C_FAX3, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, ir, C_parity_NA, C_transparent, C_modemt_none))			11.8.2.7.1
		local_TS62_FAX3_NT(ur:B_4)			
16		(TCV_bcapListIndicator := TCV_bcapListIndicator + 2)			
17		[ur = C_9600bs]			
18		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(C_Rchr_dontcare, C_FAX3, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, TCV_CE, C_modemt_none, C_X25_flct), TCV_bcapListIndicator))			11.8.2.7.2
19		[(ur = C_2400bs) OR (ur = C_4800bs)]			
20		(TCV_BcapMO1 := OC_ModifyBcap(BcapX_MO(C_Rchr_dontcare, C_FAX3, C_SDUintegrity, C_nirr_dontcare, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, TCV_CE, C_modemt_none, C_X25_flct), TCV_bcapListIndicator))			11.8.2.7.2
21	L_1017	[(ur <> C_2400bs) AND(ur <> C_4800bs) AND(ur <> C_9600bs)]		I	IXIT parameter error
		local_selectCe			
22		[TSPX_TS_62_T_NT]			
23		(TCV_ce := C_notransparent)			
24		[NOT TSPX_TS_62_T_NT]			
25		(TCV_ce := TSPX_TS_62_ce)			

Detailed Comments: The TS62 Bearer Capabilities and SETUP messages for MO is constructed according to section 11.8.2.7.1 and section 11.8.2.7.2 of GSM 11.10 and section B.1.10.2 and section B.2.11 of GSM 07.01.

```
The algorithm for selecting parameters:
IF (TSPX_BS_62_T_NT)
        TCV_ce := C_notransparent
ELSE
        TCV_ce := TSPX_BS_62_ce

use (TCV_ce, TSPX_BS_62_ur1, svc) for OO_IFsetup_TS6x_MO
```

Test Step Dynamic Behaviour

Test Step Name: TS61MT(srv:SERVICES)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To generate an MT SETUP message with appropriate Bcap for TS61 service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+local_maintree			
2		+local_IFsetup			
		local_maintree			
3		(TCV_srv := srv, TCV_2ndtest := C_Yes)			
4		(TCV_BcapMT1 := Bcap_Speech_MT)			1st Bcap
5		+local_first_SETUP_msg			
6		+local_second_SETUP_msg			
		local_first_SETUP_msg			
7		(TCV_ur := TSPX_TS_61_ur1)			
8		[TSPX_TS_61_T_NT OR (TSPX_TS_61_ce = C_notransparent)]			
9		+local_bcap2(TCV_ur, C_notransparent)			
10		(TCV_Setup_mt := Setup_21(C_RI_alternate, TCV_BcapMT1, TCV_BcapMT2, TI_02))			
11		+Chmod(TCV_ChRate, C_notransparent, TCV_ur, srv)			2nd channel mode
12		(TCV_ChMod.mode := C_ChMod_r)			Speech mode
13		[NOT TSPX_TS_61_T_NT AND (TSPX_TS_61_ce = C_transparent)]			
14		+local_bcap2(TCV_ur, C_transparent)			
15		(TCV_Setup_mt := Setup_21(C_RI_alternate, TCV_BcapMT1, TCV_BcapMT2, TI_02))			
16		+Chmod(TCV_ChRate, C_transparent, TCV_ur, srv)			2nd channel mode
17		(TCV_ChMod.mode := C_ChMod_r)			Speech mode
		local_second_SETUP_msg			
18		+local_userrate			
19		[TSPX_TS_61_T_NT]			
20		+local_bcap2(TCV_ur2, C_transparent)			2nd Bcap
21		(TCV_Setup_mt1 := Setup_21(C_RI_alternate, TCV_BcapMT1, TCV_BcapMT2, TI_02))			
22		[NOT TSPX_TS_61_T_NT]			
23		+local_bcap2(TCV_ur2, C_BothT)			
24		(TCV_Setup_mt1 := Setup_21(C_RI_alternate, TCV_BcapMT1, TCV_BcapMT2, TI_02))			
		local_bcap2(ur:B_4; ce:B_2)			
25		[ce = C_transparent]			
26		[(ur = C_2400bs) OR (ur = C_4800bs)]			
27		+local_TS61_FAX3_T(ur, C_ir_8kbs)			
28		[ur = C_9600bs]			
29		+local_TS61_FAX3_T(ur, C_ir_16kbs)			
30	L_1018	[(ur <> C_2400bs) AND(ur <> C_4800bs) AND(ur <> C_9600bs)]		I	
31		[ce = C_BothT]			
32		+local_TS61_FAX3_NT(ur, C_BothT)			
33		[ce = C_notransparent]			
34		+local_TS61_FAX3_NT(ur, C_notransparent)			
35	L_1019	[(ce <> C_transparent) AND(ce <> C_BothT) AND(ce<> C_notransparent)]		I	
		local_TS61_FAX3_T(ur:B_4; ir:B_2)			
36		(TCV_BcapMT2 := Bcap_MT(C_FAX3, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, ir, C_parity_NA, C_transparent, C_modemt_none))			11.8.1.5.1

		local_TS61_FAX3_NT(ur:B_4; ce1:B_2)			
37		(TCV_BcapMT2 := Bcap_MT(C_FAX3, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, ce1, C_modemt_none))			11.8.1.5.2
		local_userrate			
38		[TSPX_TS_61_more_ur]			
39		(TCV_ur2 := TSPX_TS_61_ur2)			
40		[NOT TSPX_TS_61_more_ur]			
41		(TCV_ur2 := TSPX_TS_61_ur1)			
		local_IFsetup			
42		[TCV_supported]			
43		(TCV_Null := OO_IFsetup_TS6x_MT(TCV_srv, TCV_ur))			
44		[NOT TCV_supported]			
45		[TCV_SpecialCase]			
46		START T_dly(C_T_Wait)			
47		?TIMEOUT T_dly			
48	L_1020	[NOT TCV_SpecialCase]		I	Parameter error

Detailed Comments: The algorithm for derivation of TS61 Bearer Capabilities:

The 1st facsimile group 3 Bearer Capability (used for an MT call):
ur= TSPX_TS_61_ur1
IF TSPX_TS_61_T_NT OR (TSPX_TS_61_ce = NT)
use coding 11.8.1.5.2 with (ur & ce=NT)
ELSE
use coding 11.8.1.5.1 with (ur & ce=T)

The 2nd facsimile group 3 Bearer Capabiility (used for TC_11_1_1):
IF TSPX_TS_61_more_ur
ur=TSPX_TS_61_ur2
ELSE
ur=TSPX_TS_61_ur1
IF TSPX_TS_61_T_NT
use coding 11.8.1.5.1 with (ur & ce=T)
ELSE
use coding 11.8.1.5.2 with(ur & ce=both/T)

Test Step Dynamic Behaviour

Test Step Name: TS62MT(srv:SERVICES)
Group: GSM_L3_MS_v4220/Preambles/
Objective: To generate an MT SETUP message with appropriate Bcap for TS62 service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+local_maintree			
2		(TCV_Null := OO_IFsetup_TS6x_MT(TCV_srv, TCV_ur))			
		local_maintree			
3		(TCV_srv := srv, TCV_2ndtest := C_Yes)			
4		+local_first_SETUP_msg			
5		+local_second_SETUP_msg			
		local_first_SETUP_msg			
6		(TCV_ur := TSPX_TS_62_ur1)			
7		[TSPX_TS_62_T_NT OR (TSPX_TS_62_ce = C_notransparent)]			
8		+local_bcap2(TCV_ur, C_notransparent)			
9		(TCV_Setup_mt := Setup_20(TCV_BcapMT2, TI_02))			
10		+Chmod(TCV_ChRate, C_notransparent, TCV_ur, srv)			
11		[NOT TSPX_TS_62_T_NT AND (TSPX_TS_62_ce = C_transparent)]			
12		+local_bcap2(TCV_ur, C_transparent)			
13		(TCV_Setup_mt := Setup_20(TCV_BcapMT2, TI_02))			
14		+Chmod(TCV_ChRate, C_transparent, TCV_ur, srv)			
		local_second_SETUP_msg			
15		+local_userrate			
16		[TSPX_TS_62_T_NT]			
17		+local_bcap2(TCV_ur2, C_transparent)			
18		(TCV_Setup_mt1 := Setup_21(C_RI_alternate, TCV_BcapMT1, TCV_BcapMT2, TI_02))			
19		[NOT TSPX_TS_62_T_NT]			
20		+local_bcap2(TCV_ur2, C_BothT)			
21		(TCV_Setup_mt1 := Setup_21(C_RI_alternate, TCV_BcapMT1, TCV_BcapMT2, TI_02))			
		local_bcap2(ur:B_4; ce:B_2)			
22		[ce = C_transparent]			
23		[(ur = C_2400bs) OR (ur = C_4800bs)]			
24		+local_TS62_FAX3_T(ur, C_ir_8kbs)			
25		[ur = C_9600bs]			
26		+local_TS62_FAX3_T(ur, C_ir_16kbs)			
27	L_1021	[(ur <> C_2400bs) AND(ur <> C_4800bs) AND(ur <> C_9600bs)]		I	
28		[ce = C_BothT]			
29		+local_TS62_FAX3_NT(ur, C_BothNT)			
30		[ce = C_notransparent]			
31		+local_TS62_FAX3_NT(ur, C_notransparent)			
32	L_1022	[(ce <> C_transparent) AND(ce <> C_BothT) AND(ce<> C_notransparent)]		I	
		local_TS62_FAX3_T(ur:B_4; ir:B_2)			
33		(TCV_BcapMT2 := Bcap_MT(C_FAX3, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, ir, C_parity_NA, C_transparent, C_modemt_none))			11.8.1.5.1
		local_TS62_FAX3_NT(ur:B_4; ce1:B_2)			

34		(TCV_BcapMT2 := Bcap_MT(C_FAX3, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, ce1, C_modemt_none))			11.8.1.5.2
		local_userrate			
35		[TSPX_TS_62_more_ur]			
36		(TCV_ur2 := TSPX_TS_62_ur2)			
37		[NOT TSPX_TS_62_more_ur]			
38		(TCV_ur2 := TSPX_TS_62_ur1)			
		local_IFsetup			
39		[TCV_supported]			
40		(TCV_Null := OO_IFsetup_TS6x_MT(TCV_srv, TCV_ur))			
41		[NOT TCV_supported]			
42		[TCV_SpecialCase]			
43		START T_dly(C_T_Wait)			
44		?TIMEOUT T_dly			
45	L_1023	[NOT TCV_SpecialCase]		I	Parameter error

Detailed Comments: The algorithm for derivation of TS62 Bearer Capabilities:

The 1st Bearer Capability (used for an MT call):
ur= TSPX_TS_62_ur1
IF TSPX_TS_62_T_NT OR (TSPX_TS_62_ce = NT)
    use coding 11.8.1.5.2 with (ur & ce=NT)
ELSE
    use coding 11.8.1.5.1 with (ur & ce=T)

The 2nd Bearer Capabiility (used for TC_11_1_1):
IF TSPX_TS_62_more_ur
    ur=TSPX_TS_62_ur2
ELSE
    ur=TSPX_TS_62_ur1
IF TSPX_TS_62_T_NT
    use coding 11.8.1.5.1 with (ur & ce=T)
ELSE
    use coding 11.8.1.5.2 with(ur & ce=both/T)

Defaults Library

Default Dynamic Behaviour					
Default Name:	DTMFsig				
Group:	GSM_L3_MS_v4220/				
Objective:	To match DTMF signalling when an MMI key in the MS is depressed or released in the active state				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L?DL_DatInStartDtmf(TCV_Char := DL_DatInStartDtmf.msg.kpf.kpf_info)	StartDTMFRcv(StartDtmf_02(TCV_TI0))		
2		L!DL_DatRqStartDtmfAck	StartDTMFAckSnd(TCV_chTch, StartDtmfAck_01(TCV_TI, TCV_Char))		
3		RETURN			
4		L?DL_DatInStopDtmf	StopDTMFRcv(StopDtmf_01(TCV_TI0))		
5		L!DL_DatRqStopDtmfAck	StopDTMFAckSnd(TCV_chTch, StopDtmfAck_01(TCV_TI))		
6		RETURN			
Detailed Comments:					

Default Dynamic Behaviour

Default Name: OtherEvents
Group: GSM_L3_MS_v4220/
Objective: To match unexpected events and sign final verdict for preambles.
Comments: used in preambles.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_02		4.
2		RETURN			
3		?TIMEOUT T_guard			1.
4		+ltree_SDCCH1_release			
5		+ltree_SDCCH2_release			
6		+ltree_TCH1_release			
7		+ltree_TCH2_release			
8		(TCV_Null := OO_TguardTimeOut())			
9	L_1024	CANCEL		I	3.
10		L?OTHERWISE			2.
11		+ltree_SDCCH1_release			
12		+ltree_SDCCH2_release			
13		+ltree_TCH1_release			
14		+ltree_TCH2_release			
15		L?DL_RelIn	DLRelInd_01		
16	L_1025	CANCEL		I	3.
17		?TIMEOUT T_guard			1.
18	L_1026	CANCEL		I	3.
		ltree_SDCCH1_release			
19		[TCV_ch <> "dummy"]			
20		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
21		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
22		[TCV_ch1 <> "dummy"]			
23		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
24		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
25		[TCV_chTch <> "dummy"]			
26		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
27		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
28		[TCV_chTch1 <> "dummy"]			
29		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
30		[TCV_chTch1 = "dummy"]			

Detailed Comments:
1. The guard timer times out, inconclusive.
2. Unexpected events, inconclusive.
3. Cancel of all running timers.
4. Local end termination request allowed.

Default Dynamic Behaviour

Default Name: OtherEventsFail
Group: GSM_L3_MS_v4220/
Objective: To match unexpected events and fail the test case.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_02		
2		RETURN			
3		?TIMEOUT T_guard			1.
4		+ltree_SDCCH1_release			
5		+ltree_SDCCH2_release			
6		+ltree_TCH1_release			
7		+ltree_TCH2_release			
8		(TCV_Null := OO_TguardTimeOut())			
9	L_1027	CANCEL		I	3.
10		L?OTHERWISE			2.
11		+ltree_SDCCH1_release			
12		+ltree_SDCCH2_release			
13		+ltree_TCH1_release			
14		+ltree_TCH2_release			
15		L?DL_RelIn	DLRelInd_01		
16	L_1028	CANCEL		F	3.
17		?TIMEOUT T_guard			
18	L_1029	CANCEL		F	3.
		ltree_SDCCH1_release			
19		[TCV_ch <> "dummy"]			
20		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
21		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
22		[TCV_ch1 <> "dummy"]			
23		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
24		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
25		[TCV_chTch <> "dummy"]			
26		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
27		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
28		[TCV_chTch1 <> "dummy"]			
29		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
30		[TCV_chTch1 = "dummy"]			

Detailed Comments:
1. The guard timer times out, inconclusive.
2. Unexpected events, fail.
3. Cancel of all running timers.

Default Dynamic Behaviour

Default Name: OtherEventsFail_01
Group: GSM_L3_MS_v4220/
Objective: To match irrelevant CHANNEL REQUEST msg and MEASUREMENT REPORT msg and return or match other unexpected events and fail the test case.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_01		6.
2		RETURN			
3		?TIMEOUT T_guard			1.
4		+ltree_SDCCH1_release			
5		+ltree_SDCCH2_release			
6		+ltree_TCH1_release			
7		+ltree_TCH2_release			
8		(TCV_Null := OO_TguardTimeOut())			
9	L_1030	CANCEL		I	5.
10		L?DL_RacInChRq	ChReq(ChRequest_02)		2.
11		RETURN			
12		L?DL_UdatInMsrRpt	MsrRept(MsrReport_02)		3.
13		RETURN			
14		L?OTHERWISE			4.
15		+ltree_SDCCH1_release			
16		+ltree_SDCCH2_release			
17		+ltree_TCH1_release			
18		+ltree_TCH2_release			
19		L?DL_RelIn	DLRelInd_01		
20	L_1031	CANCEL		F	5.
21		?TIMEOUT T_guard			
22	L_1032	CANCEL		F	5.
		ltree_SDCCH1_release			
23		[TCV_ch <> "dummy"]			
24		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
25		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
26		[TCV_ch1 <> "dummy"]			
27		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
28		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
29		[TCV_chTch <> "dummy"]			
30		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
31		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
32		[TCV_chTch1 <> "dummy"]			
33		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
34		[TCV_chTch1 = "dummy"]			

Detailed Comments:
1. The guard timer times out, inconclusive.
2. To throw away any irrelevant channel request.
3. To throw away any measurement report.
4. Other unexpected events, fail.
5. Cancel of all running timers.
6. Cover also DLRelInd_02 for local end termination.

Default Dynamic Behaviour

Default Name: OtherEventsFail_02

Group: GSM_L3_MS_v4220/

Objective: To match unexpected events and fail the test case but ignore channel request messages that are sent before the lower tester has sent (and the MS received) the Immediate Assigment message.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_02		
2		RETURN			
3		L?DL_RacInChRq	ChReq(ChRequest_02)		1.
4		RETURN			
5		?TIMEOUT T_guard			2.
6		+ltree_SDCCH1_release			
7		+ltree_SDCCH2_release			
8		+ltree_TCH1_release			
9		+ltree_TCH2_release			
10		(TCV_Null := OO_TguardTimeOut())			
11	L_1033	CANCEL		I	4.
12		L?OTHERWISE			3.
13		+ltree_SDCCH1_release			
14		+ltree_SDCCH2_release			
15		+ltree_TCH1_release			
16		+ltree_TCH2_release			
17		L?DL_RelIn	DLRelInd_01		
18	L_1034	CANCEL		F	4.
19		?TIMEOUT T_guard			
20	L_1035	CANCEL		F	4.
		ltree_SDCCH1_release			
21		[TCV_ch <> "dummy"]			
22		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
23		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
24		[TCV_ch1 <> "dummy"]			
25		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
26		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
27		[TCV_chTch <> "dummy"]			
28		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
29		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
30		[TCV_chTch1 <> "dummy"]			
31		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
32		[TCV_chTch1 = "dummy"]			

Detailed Comments:

1. To throw away any irrelevant channel request.
2. The guard timer times out, inconclusive.
3. Unexpected events, fail.
4. Cancel of all running timers.

Default Dynamic Behaviour

Default Name: OtherEvents_01
Group: GSM_L3_MS_v4220/
Objective: To match irrelevant messages and return

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?OTHERWISE			1.
2		RETURN			

Detailed Comments: 1. Cover also DLRelInd_02 for local end termination.

Default Dynamic Behaviour

Default Name: OtherEvents_02

Group: GSM_L3_MS_v4220/

Objective: To match unexpected events and sign final verdict for preambles but ignore channel request messages that are sent before the lower tester has sent (and the MS received) the Immediate Assigment message.

Comments: used in preambles.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_02		
2		RETURN			
3		L?DL_RacInChRq	ChReq(ChRequest_02)		1.
4		RETURN			
5		?TIMEOUT T_guard			2.
6		+ltree_SDCCH1_release			
7		+ltree_SDCCH2_release			
8		+ltree_TCH1_release			
9		+ltree_TCH2_release			
10		(TCV_Null := OO_TguardTimeOut())			
11	L_1036	CANCEL		I	4.
12		L?OTHERWISE			3.
13		+ltree_SDCCH1_release			
14		+ltree_SDCCH2_release			
15		+ltree_TCH1_release			
16		+ltree_TCH2_release			
17		L?DL_RelIn	DLRelInd_01		
18	L_1037	CANCEL		I	4.
19		?TIMEOUT T_guard			2.
20	L_1038	CANCEL		I	4.
		ltree_SDCCH1_release			
21		[TCV_ch <> "dummy"]			
22		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
23		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
24		[TCV_ch1 <> "dummy"]			
25		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
26		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
27		[TCV_chTch <> "dummy"]			
28		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
29		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
30		[TCV_chTch1 <> "dummy"]			
31		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
32		[TCV_chTch1 = "dummy"]			

Detailed Comments:

1. To throw away any irrelevant channel request.
2. The guard timer times out, inconclusive.
3. Unexpected events, inconclusive.
4. Cancel of all running timers.

Default Dynamic Behaviour

Default Name: RcvHdOvAcc
Group: GSM_L3_MS_v4220/
Objective: To match any HANDOVER ACCESS then return to calling tree.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RacInHoacc	HndOvAcc_01		
2		RETURN			

Detailed Comments:

Annex B (normative): Partial IXIT proforma

Notwithstanding the provisions of the copyright clause related to the text of this ETS, ETSI grants that users of this ETS may freely reproduce the PIXIT proforma in this annex so that it can be used for its intended purposes and may further publish the completed PIXIT.

Introduction

This partial IXIT proforma contained in this ETS, after augmented by the Test Realizer, is proposed to be provided to the client for completion, when the related Abstract Test Suite is to be used against client's Implementation Under Test (IUT).

Text in *italics* is comments for guidance for the production of a IXIT, and is not to be included in the actual IXIT.

The completed IXIT will normally be used in conjunction with the completed ICS, as it adds precision to the information provided by the ICS.

B.1 Identification Summary

This table is completed by the test laboratory. The item "Contract References" is optional.

Table 1: Identification Summary

IXIT Reference Number	
Test Laboratory Name	
Date of Issue	
Issued to (name of client)	
Contract References	

B.2 Abstract Test Suite Summary

In the following table the test laboratory provides the version number of the protocol specification and the version number of ATS which are used in the conformance testing.

Table 2: ATS Summary

Protocol Specification	ETS 300 557
Version of Protocol Specification	
TSS & TP Specification	ETS 300 607-1
Version of TSS & TP Specification	
ATS Specification	ETS 300 607-3
Version of ATS Specification	
Abstract Test Method	Remote Test Method

B.3 Test Laboratory

B.3.1 Test Laboratory Identification

The test laboratory provides the following information.

Table 3: Test Laboratory Identification

Name of Test Laboratory	
Postal Address	
Office address	
Telephone Number	
FAX Number	

B.3.2 Accreditation status of the test service

The test laboratory provides the following information.

Table 4: Accreditation status of the test service

Accreditation status	
Accreditation Reference	

B.3.3 Manager of Test Laboratory

The test laboratory provides the information about the manager of test laboratory in the following table.

Table 5: Manager of Test Laboratory

Name of Manager of Test Laboratory	
Telephone Number	
FAX Number	
E-mail Address	

B.3.4 Contact person of Test Laboratory

The test laboratory provides the information about the contact person of test laboratory in the following table.

Table 6: Contact person of Test Laboratory

Name of Contact of Test Laboratory	
Telephone Number	
FAX Number	
E-mail Address	

B.3.5 Means of Testing

In the table below, the test laboratory provides a statement of conformance of the Means Of Testing (MOT) to the reference standardized ATS, and identifies all restrictions for the test execution required by the MOT beyond those stated in the reference standardized ATS.

Table 7: Means of Testing

Means of Testing

B.3.6 Instructions for Completion

In this table, the test laboratory provides any specific instructions necessary for completion and return of the proforma from the client.

Table 8: Instruction for Completion

Instructions for Completion

B.4 Client

B.4.1 Client Identification

The client provides the identification in the following table.

Table 9: Client Identification

Name of Client	
Postal Address	
Office Address	
Telephone Number	
FAX Number	

B.4.2 Client Test Manager

In this table the client provides information about the test manager.

Table 10: Client Test Manager

Name of Client Test Manager	
Telephone Number	
FAX Number	
E-mail Address	

B.4.3 Client Contact person

In this table the client provides information about the test contact person.

Table 11: Client Contact person

Name of Client contact person	
Telephone Number	
FAX Number	
E-mail Address	

B.4.4 Test Facilities Required

In the following table, the client records the particular facilities required for testing, if a range of facilities is provided by the test laboratory.

Table 12: Test Facilities Required

Test Facilities Required

B.5 System Under Test

B.5.1 SUT Information

The client provides information about the SUT in the table below.

Table 13: SUT Information

System Name	
System Version	
SCS Reference	
Machine Configuration	
Operating System Identification	
IUT Identification	
ICS Reference for the IUT	

B.5.2 Limitations of the SUT

In the table below, the client provides information explaining if any of the abstract tests cannot be executed.

Table 14: Limitation of the SUT

Limitations of the SUT

B.5.3 Environmental Conditions

In the table below the client provides information about any tighter environmental conditions for the correct operation of the SUT.

Table 15: Environmental Conditions

Environmental Conditions

B.6 Ancillary Protocols

This clause is completed by the client in conjunction with the test laboratory.

In the following tables, the client identifies relevant information concerning each ancillary protocol in the SUT other than the IUT itself. One table for one ancillary protocol.

Based on the MOT the test laboratory should create question proforma for each ancillary protocol in the blank space following each table. The information required is dependent on the MOT and the SUT, and covers all the addressing, parameter values, timer values and facilities (relevant to ETSs) as defined by the ICS for the ancillary protocol.

B.6.1 Ancillary Protocols 1

Table 16: Ancillary Protocol 1

Protocol Name	ETS 300
Version number	
ICS Reference (optional)	
IXIT Reference (optional)	
PCTR Reference (optional)	

B.6.2 Ancillary Protocols 2

Table 17: Ancillary Protocol 2

Protocol Name	ETS 300
Version number	
ICS Reference (optional)	
IXIT Reference (optional)	
PCTR Reference (optional)	

B.7 Protocol Layer Information for L3 of Mobile Station

B.7.1 Information provided for test purposes by the MS supplier

Item	Description	Type/Allowed values	Supported Value
TSPC_24DataF	2.4 k full rate data mode supported	BOOLEAN	
TSPC_24DataH	2.4 k half rate data mode supported	BOOLEAN	
TSPC_48DataF	4.8 k full rate data mode supported	BOOLEAN	
TSPC_48DataH	4.8 k half rate data mode supported	BOOLEAN	
TSPC_96Data	9.6 k full rate data mode supported	BOOLEAN	
TSPC_Feat_A51	ciphering algorithm A5/1 supported	BOOLEAN	
TSPC_Feat_A52	ciphering algorithm A5/2 supported	BOOLEAN	
TSPC_AddCharSet	A, B, C, D chars supported	BOOLEAN	
TSPC_AddInfo_PseudoSynch	Pseudo synchronised supported	BOOLEAN	
TSPC_AlertInd	alerting indication to the user supported	BOOLEAN	
TSPC_SvcOnTCH	at least one service on traffic channel supported	BOOLEAN	

Item	Description	Type/Allowed values	Supported Value
TSPC_SMS	at least one short message service supported	BOOLEAN	
TSPC_SS	at least one supplementary service supported	BOOLEAN	
TSPC_AutoAutoMode	automatically enter automatic selection of PLMN mode supported	BOOLEAN	
TSPC_BasCharSet	Chars 0-9, *, # supported	BOOLEAN	
TSPC_CalledNumDisp	called number display supported	BOOLEAN	
TSPC_DCS	testing DCS1 800	BOOLEAN	
TSPC_DetachOnPwrDn	detach on power down supported	BOOLEAN	
TSPC_DetachOnSIMRmv	detach on SIM remove supported	BOOLEAN	
TSPC_DispRcvSMS	display of received SMS supported	BOOLEAN	
TSPC_DualRate	dual rate channel types supported	BOOLEAN	
TSPC_EGSM	both standard and extended GSM band supported	BOOLEAN	
TSPC_EmgOnly	the only circuit switched basic service is emergency call	BOOLEAN	
TSPC_followOnReq	follow-on request procedure supported	BOOLEAN	
TSPC_HalfRateData	half rate data supported	BOOLEAN	
TSPC_HalfRateSpeech	half rate speech mode supported	BOOLEAN	
TSPC_InCallMod	In-Call modification supported	BOOLEAN	
TSPC_NoimmConn	at least one service not support immediate connection	BOOLEAN	
TSPC_Feat_FND	FND feature supported	BOOLEAN	
TSPC_NonCallSS	non call related supplementary service supported	BOOLEAN	
TSPC_MTsvc	at least one MT circuit switched basic service supported	BOOLEAN	
TSPC_MOsvc	at least one MO circuit switched basic service supported	BOOLEAN	
TSPC_PGSM	only standard GSM band supported	BOOLEAN	
TSPC_RefusalCall	refusal of call supported	BOOLEAN	
TSPC_ReplaceSMS	replace SMS supported	BOOLEAN	
TSPC_ReplyProc	(SMS) reply procedures supported	BOOLEAN	
TSPC_RFAmp	RF amplification supported	BOOLEAN	
TSPC_SDCCHOnly	only SDCCH supported	BOOLEAN	

Item	Description	Type/Allowed values	Supported Value
TSPC_Serv_SS_AoCC	AOCC SS supported	BOOLEAN	
TSPC_Serv_SS_BAIC	BAIC SS supported	BOOLEAN	
TSPC_Serv_SS_BI	BI SS supported	BOOLEAN	
TSPC_Serv_SS_BICRoam	BICRoam SS supported	BOOLEAN	
TSPC_Serv_SS_BAOC	BAOC SS supported	BOOLEAN	
TSPC_Serv_SS_BOIC	BOIC SS supported	BOOLEAN	
TSPC_Serv_SS_BOICexHC	BOICExHC SS supported	BOOLEAN	
TSPC_Serv_SS_CFB	CFB supported	BOOLEAN	
TSPC_Serv_SS_CFNRy	CFNry supported	BOOLEAN	
TSPC_Serv_SS_CFNRc	CFNRc supported	BOOLEAN	
TSPC_Serv_SS_CFU	CFU supported	BOOLEAN	
TSPC_Serv_SS_HOLD	Call Hold SS supported	BOOLEAN	
TSPC_Serv_SS_MPTY	multiparty SS supported	BOOLEAN	
TSPC_Serv_SS_unstruct	USSD supported	BOOLEAN	
TSPC_Serv_TS11	telephony supported	BOOLEAN	
TSPC_Serv_TS12	emergency call supported	BOOLEAN	
TSPC_Serv_TS21	SMS MT/PP supported	BOOLEAN	
TSPC_Serv_TS22	SMS MO/PP supported	BOOLEAN	
TSPC_Serv_TS23	SMS cell broadcast supported	BOOLEAN	
TSPC_Serv_TS61	alternate speech and G3 fax (TS61) supported	BOOLEAN	
TSPC_Serv_TS62	automatic G3 fax (TS62) supported	BOOLEAN	
TSPC_Serv_BS21	data circuit duplex async 300 bit/s supported	BOOLEAN	
TSPC_Serv_BS22	data circuit duplex async 1 200 bit/s supported	BOOLEAN	
TSPC_Serv_BS23	data circuit duplex async 1 200/75 bit/s supported	BOOLEAN	
TSPC_Serv_BS24	data circuit duplex async 2 400 bit/s supported	BOOLEAN	
TSPC_Serv_BS25	data circuit duplex async 4 800 bit/s supported	BOOLEAN	
TSPC_Serv_BS26	data circuit duplex async 9 600 bit/s supported	BOOLEAN	
TSPC_Serv_BS31	data circuit duplex sync 1 200 bit/s supported	BOOLEAN	

Item	Description	Type/Allowed values	Supported Value
TSPC_Serv_BS32	data circuit duplex sync 2 400 bit/s supported	BOOLEAN	
TSPC_Serv_BS33	data circuit duplex sync 4 800 bit/s supported	BOOLEAN	
TSPC_Serv_BS34	data circuit duplex sync 9 600 bit/s supported	BOOLEAN	
TSPC_Serv_BS41	PAD access 300 bit/s supported	BOOLEAN	
TSPC_Serv_BS42	PAD access 1 200 bit/s supported	BOOLEAN	
TSPC_Serv_BS43	PAD access 1 200/75 bit/s supported	BOOLEAN	
TSPC_Serv_BS44	PAD access 2 400 bit/s supported	BOOLEAN	
TSPC_Serv_BS45	PAD access 4 800 bit/s supported	BOOLEAN	
TSPC_Serv_BS46	PAD access 9 600 bit/s supported	BOOLEAN	
TSPC_Serv_BS51	packet access 2 400 bit/s supported	BOOLEAN	
TSPC_Serv_BS52	packet access 4 800 bit/s supported	BOOLEAN	
TSPC_Serv_BS53	packet access 9 600 bit/s supported	BOOLEAN	
TSPC_Serv_BS61	alternate speech/data (BS61) supported	BOOLEAN	
TSPC_Serv_BS81	speech followed data (BS81) supported	BOOLEAN	
TSPC_SMSStatusRepCap	SMS status report capabilities supported	BOOLEAN	
TSPC_StoreRcvSMSME	Storage of received SMS in ME supported	BOOLEAN	
TSPC_StoreRcvSMSSIM	Storage of received SMS in SIM supported	BOOLEAN	
TSPC_SIMRmv	SIM removable without power down supported	BOOLEAN	
TSPC_SwitchOnOff	switch on/off supported	BOOLEAN	
TSPC_TranspDataOnly	only transparent data service supported	BOOLEAN	
TSPX_TE_stopbit	TE conf. number stop bits. default value : '0'B, 1 bit	BITSTRING[1]	
TSPX_TE_databit	TE conf. number data bits. default value : '1'B, 8 bits	BITSTRING[1]	
TSPX_TE_FLCT	TE flow control. default : 0, outband flow control	0, 1, 2	
TSPX_TE_parity	TE conf. parity. default : '001'B, no parity	BITSTRING[3]	
TSPX_BS_21_itc1	itc value supported for BS21. default value : '001'B, UDI	BITSTRING[3]	
TSPX_BS_21_more_itc	more than one itc value supported by BS21. default value : FALSE, only one itc	BOOLEAN	
TSPX_BS_21_T_NT	both T and NT supported by BS21. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	

Item	Description	Type/Allowed values	Supported Value
TSPX_BS_21_ce	ce value used in MO call and if the answer to TSPX_BS_21_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent mode	BITSTRING[2]	
TSPX_BS_21_sacp	sacp value used in MO call. default value : '001'B, I440/450	BITSTRING[3]	
TSPX_BS_21_itc2	other itc value supported by BS21, it shall be differenet from the previous one if the answer to TSPX_BS_21_more_itc is TRUE. default value :'001'B, UDI	BITSTRING[3]	
TSPX_BS_22_itc1	itc value supported by BS22. default value : '010'B, 3.1kHz audio, exPLMN	BITSTRING[3]	
TSPX_BS_22_more_itc	more than one itc value supported by BS22. default value : FALSE, only one itc	BOOLEAN	
TSPX_BS_22_T_NT	both T and NT supported by BS22. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_BS_22_ce	ce value used in MO call and if the answer to TSPX_BS_22_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent	BITSTRING[2]	
TSPX_BS_22_sacp	sacp value used in MO call. default value : '001'B, I440/450	BITSTRING[3]	
TSPX_BS_22_itc2	other itc value supported by BS22, it shall be differenet from the previous one if the answer to TSPX_BS_22_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN	BITSTRING[3]	
TSPX_BS_23_itc	itc value supported by BS23. default value : '010'B, 3.1kHz audio, exPLMN	BITSTRING[3]	
TSPX_BS_23_T_NT	both T and NT supported for BS23. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_BS_23_ce	ce value used in MO call. default value : '00'B, transparent	BITSTRING[2]	
TSPX_BS_23_sacp	sacp value used in MO call. default value : '001'B, I440/450	BITSTRING[3]	
TSPX_BS_24_itc1	itc value supported by BS24. default value : '010'B, 3.1kHz audio, exPLMN	BITSTRING[3]	

Item	Description	Type/Allowed values	Supported Value
TSPX_BS_24_more_itc	more than one itc value supported by BS24. default value : FALSE, only one itc	BOOLEAN	
TSPX_BS_24_T_NT	both T and NT supported by BS24. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_BS_24_ce	ce value used in MO call and if the answer to TSPX_BS_24_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent	BITSTRING[2]	
TSPX_BS_24_sacp	sacp value used in MO call. default value : '001'B, I440/450	BITSTRING[3]	
TSPX_BS_24_itc2	other itc value supported by BS24, it shall be differenet from the previous one if the answer to TSPX_BS_24_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN	BITSTRING[3]	
TSPX_BS_25_itc1	itc value supported by BS25. default value : '001'B, UDI	BITSTRING[3]	
TSPX_BS_25_more_itc	more than one itc value supported by BS25. default value : FALSE, only one itc	BOOLEAN	
TSPX_BS_25_T_NT	both T and NT supported by BS25. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_BS_25_ce	ce value used in MO call and if the answer to TSPX_BS_25_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent	BITSTRING[2]	
TSPX_BS_25_sacp	sacp value used in MO call. default value : '001'B, I440/450	BITSTRING[3]	
TSPX_BS_25_itc2	other itc value supported by BS25, it shall be differenet from the previous one if the answer to TSPX_BS_25_more_itc is TRUE. default value : '001'B, UDI	BITSTRING[3]	
TSPX_BS_26_itc1	itc value supported by BS26. default value : '010'B, 3.1kHz audio, exPLMN	BITSTRING[3]	
TSPX_BS_26_more_itc	more than one itc value supported by BS26. default value : FALSE, only one itc	BOOLEAN	
TSPX_BS_26_T_NT	both T and NT supported by BS26. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	

Item	Description	Type/Allowed values	Supported Value
TSPX_BS_26_ce	ce value used in MO call and if the answer to TSPX_BS_26_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent	BITSTRING[2]	
TSPX_BS_26_sacp	sacp value used in MO call. default value : '001'B, I440/450	BITSTRING[3]	
TSPX_BS_26_itc2	other itc value supported by BS26, it shall be differenet from the previous one if the answer to TSPX_BS_26_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN	BITSTRING[3]	
TSPX_BS_31_more_itc	more than one itc value supported by BS31. default value : FALSE, only one itc	BOOLEAN	
TSPX_BS_31_more_sacp	more than one sacp value supported by BS31. default value : FALSE, only one sacp	BOOLEAN	
TSPX_BS_31_itc1	itc value supported by BS31. default value : '001'B, UDI	BITSTRING[3]	
TSPX_BS_31_sacp1	sacp value supported by BS31. default value : '001'B, I440/450	BITSTRING[3]	
TSPX_BS_31_itc2	other itc value supported by BS31, it shall be differenet from the previous one if the answer to TSPX_BS_31_more_itc is TRUE. default value : '001'B, UDI	BITSTRING[3]	
TSPX_BS_31_sacp2	other sacp value supported by BS31, it shall be differenet from the previous one if the answer to TSPX_BS_31_more_sacp is TRUE. default value : '001'B, I440/450	BITSTRING[3]	
TSPX_BS_32_itc1	itc value supported by BS32. default value : '010'B, 3.1kHz audio, exPLMN	BITSTRING[3]	
TSPX_BS_32_more_itc	more than one itc value supported by BS32. default value : FALSE, only one itc	BOOLEAN	
TSPX_BS_32_more_sacp	more than one sacp value supported by BS32. default value : FALSE, only one sacp	BOOLEAN	
TSPX_BS_32_sacp1	sacp value supported by BS32. default value : '110'B, X.32	BITSTRING[3]	
TSPX_BS_32_X32_T_NT	BS 32 both T and NT supported for X32 3.1kHz. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	

Item	Description	Type/Allowed values	Supported Value
TSPX_BS_32_X32_ce	if the answer to TSPX_BS_32_X32_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent	BITSTRING[2]	
TSPX_BS_32_itc2	other itc value supported by BS32, it shall be differenet from the previous one if the answer to TSPX_BS_32_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN	BITSTRING[3]	
TSPX_BS_32_sacp2	other sacp value supported by BS32, it shall be differenet from the previous one if the answer to TSPX_BS_32_more_sacp is TRUE. default value : '110'B, X.32	BITSTRING[3]	
TSPX_BS_33_itc1	itc value supported by BS33. default value : '001'B, UDI	BITSTRING[3]	
TSPX_BS_33_more_itc	more than one itc value supported by BS33. default value : FALSE, only one itc	BOOLEAN	
TSPX_BS_33_more_sacp	more than one sacp value supported by BS33. default value : FALSE, only one sacp	BOOLEAN	
TSPX_BS_33_sacp1	sacp value supported by BS33. default value : '010'B, X.21	BITSTRING[3]	
TSPX_BS_33_X32_T_NT	BS 33 both T and NT supported for X32 3.1kHz. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_BS_33_X32_ce	if the answer to TSPX_BS_33_X32_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent	BITSTRING[2]	
TSPX_BS_33_itc2	other itc value supported by BS32, it shall be differenet from the previous one if the answer to TSPX_BS_33_more_itc is TRUE. default value : '001'B, UDI	BITSTRING[3]	
TSPX_BS_33_sacp2	other sacp value supported by BS33, it shall be differenet from the previous one if the answer to TSPX_BS_33_more_sacp is TRUE. default value : '010'B, X.21	BITSTRING[3]	
TSPX_BS_34_itc1	itc value supported by BS34. default value : '010'B, 3.1kHz audio, exPLMN	BITSTRING[3]	

Item	Description	Type/Allowed values	Supported Value
TSPX_BS_34_more_itc	more than one itc value supported by BS34. default value : FALSE, only one itc	BOOLEAN	
TSPX_BS_34_more_sacp	more than one sacp value supported by BS34. default value : FALSE, only one sacp	BOOLEAN	
TSPX_BS_34_sacp1	sacp value supported by BS34. default value : '001'B, I440/450	BITSTRING[3]	
TSPX_BS_34_X32_T_NT	BS 34 both T and NT supported for X32 3.1kHz. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_BS_34_X32_ce	if the answer to TSPX_BS_34_X32_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent	BITSTRING[2]	
TSPX_BS_34_itc2	other itc value supported by BS34, it shall be differenet from the previous one if the answer to TSPX_BS_34_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN	BITSTRING[3]	
TSPX_BS_34_sacp2	other sacp value supported by BS34, it shall be differenet from the previous one if the answer to TSPX_BS_34_more_sacp is TRUE. default value : '001'B, I440/450	BITSTRING[3]	
TSPX_BS_41_T_NT	BS41 both T and NT supported . default value : FALSE, not support both	BOOLEAN	
TSPX_BS_41_ce	ce value used in BS41 MO call. default value : '00'B, transparent	BITSTRING[2]	
TSPX_BS_42_T_NT	BS42 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_BS_42_ce	ce value used in BS42 MO call. default value : '010'B, non transparent	BITSTRING[2]	
TSPX_BS_43_T_NT	BS43 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_BS_43_ce	ce value used in BS43 MO call. default value : '00'B, transparent	BITSTRING[2]	

Item	Description	Type/Allowed values	Supported Value
TSPX_BS_44_T_NT	BS44 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_BS_44_ce	ce value used in BS44 MO call. default value : '00'B, transparent	BITSTRING[2]	
TSPX_BS_45_T_NT	BS45 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_BS_45_ce	ce value used in BS45 MO call. default value : '01'B, non transparent	BITSTRING[2]	
TSPX_BS_46_T_NT	BS46 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_BS_46_ce	ce value used in BS46 MO call. default value : '00'B, transparent	BITSTRING[2]	
TSPX_BS_61_S	synchronous data supported by BS 61 . default value : TRUE, synchronous mode	BOOLEAN	
TSPX_BS_61_S_more_ur	more than one user rate for synchronous data service supported by BS61. default value : FALSE, only ine user rate	BOOLEAN	
TSPX_BS_61_S_ur1	usrer rate value supported for synchronous data service of BS61. default value : '0100'B, 4.8 kbit/s	BITSTRING[4]	
TSPX_BS_61_S_ur2	another usrer rate value supported for synchronous data service of BS61. it shall be different from the previous one if the answer to TSPX_BS_61_S_more_ur is TRUE. default value : '0100'B, 4.8 kbit/s	BITSTRING[4]	
TSPX_BS_61_A	asynchronous data service supported by BS61 . default value : TRUE, asynchronous	BOOLEAN	
TSPX_BS_61_A_ur1	usrer rate value supported for asynchronous data service of BS61. default value : '0101'B, 9.6 kbit/s	BITSTRING[4]	
TSPX_BS_61_A_ur1_T_NT	both T and NT supported for the rate TSPX_BS_61_A_ur1 of BS61. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	

Item	Description	Type/Allowed values	Supported Value
TSPX_BS_61_A_ur1_ce	if the answer to TSPX_BS_61_A_ur1_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent	BITSTRING[2]	
TSPX_BS_61_A_more_ur	more than one user rate for asynchronous data service supported by BS61. default value : FALSE, only one user rate	BOOLEAN	
TSPX_BS_61_A_ur2	another usrer rate value supported for asynchronous data service of BS61. it shall be different from the previous one if the answer to TSPX_BS_61_A_more_ur is TRUE. default value : '0101'B, 9.6 kbit/s	BITSTRING[4]	
TSPX_BS_61_A_ur2_T_NT	both T and NT supported for the rate TSPX_BS_61_A_ur2 of BS61. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_BS_81_S	synchronous data supported by BS 81. default value : TRUE, synchronous mode	BOOLEAN	
TSPX_BS_81_S_more_ur	more than one user rate for synchronous data service supported by BS81. default value : FALSE, only one user rate	BOOLEAN	
TSPX_BS_81_S_ur1	usrer rate value supported for synchronous data service of BS81. default value : '0101'B, 9.6 kbit/s	BITSTRING[4]	
TSPX_BS_81_S_ur2	another usrer rate value supported for synchronous data service of BS81. it shall be different from the previous one if the answer to TSPX_BS_81_S_more_ur is TRUE. default value : '0101'B, 9.6 kbit/s	BITSTRING[4]	
TSPX_BS_81_A	asynchronous data service supported by BS81. default value : TRUE, asynchronous mode	BOOLEAN	
TSPX_BS_81_A_ur1	usrer rate value supported for asynchronous data service of BS81. default value : '0100'B, 4.8 kbit/s	BITSTRING[4]	
TSPX_BS_81_A_ur1_T_NT	both T and NT supported for the rate TSPX_BS_81_A_ur1 of BS81. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	

Item	Description	Type/Allowed values	Supported Value
TSPX_BS_81_A_ur1_ce	if the answer to TSPX_BS_81_A_ur1_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent	BITSTRING[2]	
TSPX_BS_81_A_more_ur	more than one user rate for asynchronous data service supported by BS81. default value : FALSE, only one user rate	BOOLEAN	
TSPX_BS_81_A_ur2	another usrer rate value supported for asynchronous data service of BS81. it shall be different from the previous one if the answer to TSPX_BS_81_A_more_ur is TRUE. default value : '0100'B, 4.8 kbit/s	BITSTRING[4]	
TSPX_BS_81_A_ur2_T_NT	both T and NT supported for the rate TSPX_BS_81_A_ur2 of BS81. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_TS_61_T_NT	both T and NT supported for TS61(fax). default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_TS_61_ce	if the answer to TSPX_TS_61_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent	BITSTRING[2]	
TSPX_TS_61_ur1	user rate for TS61(fax). default value : '0101'B, 9.6 kbit/s	BITSTRING[4]	
TSPX_TS_61_more_ur	more than one user rate supported for TS61. default value : FALSE, only one user rate	BOOLEAN	
TSPX_TS_61_ur2	another user rate for TS61, it shall be different from the previous one if the answer to TSPX_TS_61_more_ur is TRUE. default value : '0101'B, 9.6 kbit/s	BITSTRING[4]	
TSPX_TS_62_T_NT	both T and NT supported for TS62. default value : FALSE, not support both transparent and non transparent modes	BOOLEAN	
TSPX_TS_62_ce	if the answer to TSPX_TS_62_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent	BITSTRING[2]	
TSPX_TS_62_ur1	user rate for TS62. default value : '0100'B, 4.8 kbit/s	BITSTRING[4]	

Item	Description	Type/Allowed values	Supported Value
TSPX_TS_62_more_ur	more than one user rate supported by TS62. default value : FALSE, only one user rate	BOOLEAN	
TSPX_TS_62_ur2	another user rate for TS62, it shall be different from the previous one if the answer to TSPX_TS_62_more_ur is TRUE. default value : '0100'B, 4.8 kbit/s	BITSTRING[4]	
TSPX_DTMF	call control capabilities: value '1'B means MS supports DTMF	BITSTRING[1]	
TSPX_modF	any non signalling full rate channel mode for TC_26_6_4_1	BITSTRING[8]	
TSPX_modH	any non signalling half channel mode for TC_26_6_4_1	BITSTRING[8]	
TSPX_anymod	channel mode for TC_62_10_2_4_2. Any supported value except signalling and Full rate speech. possible values are '00000011'B' 12k data, '00001011'B 6k data, '0010011'B 3.6k data	BITSTRING[8]	
TSPX_CKSNA	cipher key sequence number	BITSTRING[3]	
TSPX_CKSNB	cipher key sequence number	BITSTRING[3]	
TSPX_CKSNC	cipher key sequence number	BITSTRING[3]	
TSPX_CKSNDef	default cipher key sequence number	BITSTRING[3]	
TSPX_RfPwrCap	RF power capability	BITSTRING[3]	
TSPX_RevLevel	Revision level of classmark 1	BITSTRING[2]	
TSPX_CiphAlgA5_1	A5/1 algorithm: '1'B : available, '0'B: not available	BITSTRING[1]	
TSPX_CiphAlgA5_2	A5/2 algorithm: '1'B : available, '0'B: not available	BITSTRING[1]	
TSPX_CiphAlgA5_3	A5/3 algorithm: '1'B : available, '0'B: not available	BITSTRING[1]	
TSPX_CiphAlgA5_4	A5/4 algorithm: '1'B : available, '0'B: not available	BITSTRING[1]	
TSPX_CiphAlgA5_5	A5/5 algorithm: '1'B : available, '0'B: not available	BITSTRING[1]	
TSPX_CiphAlgA5_6	A5/6 algorithm: '1'B : available, '0'B: not available	BITSTRING[1]	
TSPX_CiphAlgA5_7	A5/7 algorithm: '1'B : available, '0'B: not available	BITSTRING[1]	
TSPX_pSyncCap	pseudo synchronisation capability, '0'B: not present '1'B: present	BITSTRING[1]	
TSPX_SSscrnInd	SS screen indicator	BITSTRING[2]	

Item	Description	Type/Allowed values	Supported Value
TSPX_SMCap	short message capability, '0'B: not present '1'B: present	BITSTRING[1]	
TSPX_frqCap	frequency capability. DCS: reserved, GSM: '0'B not support extension band, '1'B support extension band	BITSTRING[1]	
TSPX_ClsMk3	classmark 3 indicator, '0'B no addition capability info, '1'B additional capability info in class mark 3	BITSTRING[1]	
TSPX_RfPwrCapAmp	RF power capability with external RF amplifier	BITSTRING[3]	
TSPX_DTMFInd	DTMF indication to user supported	BOOLEAN	
TSPX_IMEI	IMEI of the MS.	HEXSTRING	
TSPX_IMEISV	IMEISV of the MS. Used in: TC_26_6_8_5, TC_26_7_3_1	HEXSTRING	
TSPX_TC1M	timer value for GSM timer TC1M (for SMS)	INTEGER	
TSPX_MaxCPDataRetx	max. number of CP data retransmissions for SMS	INTEGER	
TSPX_Telephony_Immconn	Immediate connect for telephony supported ?	BOOLEAN	
TSPX_Uupd	user-user protocol discriminator	BITSTRING[8]	
TSPX_UuInfo	user-user information	OCTETSTRING	

B.7.2 MMI information

How is the Accumulated Call Meter (ACM) on the SIM read ? NOTE: Used in Operation OO_ACMIncCHK and OO_ACMReading.
How is the ACM on the SIM reset to zero, and ACMmax set to 2 units? NOTE: Used in Operation OO_ACMSetting.
How is power amplification added to the mobile and removed? NOTE: Used in Operation OO_AddPwrAmp and OO_RemvPwrAmp.

What alerting indication is given by the mobile? NOTE: Used in Operation OO_AltIndCHK.
How is the called party number displayed on the mobile? NOTE: Used in Operation OO_CalledNumCHK and OO_CalledPtyNumCHK.
How is a call placed on Hold? NOTE: Used in Operation OO_CallHold.
How is the presence of short messages checked in the mobile and how are they displayed? NOTE: Used in Operation OO_CheckAllSMPresentBut4th and OO_CheckMessageDisplayed and OO_DisplaySMAndSendReplySM and OO_ReadSMAndRemove and OO_RecallAndDisplaySM.
How is the presence of Cell Broadcast short messages in the mobile checked? NOTE: Used in Operation OO_CheckCBSMReceived.
How is it checked whether the memory capacity of the SIM has been exceeded? NOTE: Used in Operation OO_CheckMCEFOnSIM and OO_CheckMCEFOnSIMUnset.
How and where is the SIM inserted and removed? NOTE: Used in Operation OO_ConnectSIMSimulator and OO_SIMIns and OO_SIM2Ins and OO_SIM3Ins and OO_SIMRmv.
Does the MS have an 'END' key? If not, what method is used to replace its function? NOTE: Used in Operation OO_DepressEndKey.

How is a number entered in order to make an outgoing call? NOTE: Used in Operation OO_DiallCalledNum.
How is a reply to a received short message sent? NOTE: Used in Operation OO_DisplaySMAndSendReplySM.
How is it checked which DTMF character has been sent? NOTE: Used in Operation OO_DTMFIndCHK.
How is the message store of the MS emptied? NOTE: Used in OO_EmptyMessageStorage.
How is a password on the MS entered? NOTE: Used in Operation OO_EnterPswd.
How is an incoming call answered? NOTE: Used in Operation OO_HookOff.
For each service on the MS which requires the MS to be connected to an interface on other equipment (e.g. a computer), how is that interface set up for the purpose? NOTE: Used in Operation OO_IFsetup.
How is an In-Call modification initiated on the MS? NOTE: Used in Operation OO_InCallMod.

For each mobile originating basic service supported by the MS, how is the MS configured to make an outgoing call on that service? NOTE: Used in Operation OO_InitCall.
How is the MS to configured accept standard MMI sequences for the initiation of supplementary services? NOTE: Used in Operation OO_InitSS
How is the MS checked whether it is in the normal service state (idle, updated). NOTE: Used in Operation OO_InServiceCHK and OO_PressKeyWhenInService.
How is a Multi-Party call initiated on the MS. NOTE: Used in Operation OO_MptyCall
How is the MS configured to store received Class 1 short messages in the mobile equipment (rather than in the SIM)? NOTE: Used in Operation OO_MSSetupStoreClass1SMInMEMemory.
How is the list of available PLMNs checked? NOTE: Used in Operation OO_PLMNsCHK.
How is the PLMN selection mode switched to automatic selection. NOTE: Used in operation OO_PLMNselModeAuto.
How is the PLMN selection mode switched to manual selection. NOTE: Used in operation OO_PLMNselModeMan.

How is the power source for the MS removed and replaced. (Note, this is not the same as switching the Mobile on and off using the on/off switch.) NOTE: Used in operation OO_PowerUp and OO_PowerDown.
How is a short message removed from the message store? NOTE: Used in operation OO_ReadSMAndRemove.
When in manual PLMN selection mode, how is a PLMN selected? NOTE: Used in OperationOO_SelPLMN
How is a short message sent from the MS? NOTE: Used in operation OO_SendMOShortMessage.
How is an SMS COMMAND message sent to delete the last short message sent? NOTE: Used in OO_SendSMSCOMMANDDe.
How is an SMS COMMAND message sent enquiring about the last short message sent? NOTE: Used in OO_SendSMS_COMMANDEnq
How is Call Refusal set on the MS? NOTE: Used in Operation OO_SetRefuseCall.
How is a DTMF digit sent? NOTE: Used in operation OO_ShortKeyDepr.

How is it checked whether a supplementary service result is correct? NOTE: Used in Operation OO_SSresultCHK
How is the MS switched on and off? NOTE: Used in operation OO_SwitchOn and OO_SwitchOff.
How is a call cleared? NOTE: Used in operation OO_TermCall.

B.7.3 Test house specified parameters

Item	Description	Type/Allowed values	Value chosen
TSPX_AltNb	use alternative neighbour cells description. default : FALSE	BOOLEAN	
TSPX_CphAlgA	any supported ciphering algorithm	OCTETSTRING [1]	
TSPX_CphAlgB	ciphering algorithm supported	OCTETSTRING [1]	
TSPX_CphAlgC	ciphering algorithm supported	OCTETSTRING [1]	
TSPX_CphAlgD	ciphering algorithm supported	OCTETSTRING [1]	
TSPX_CphAlgE	ciphering algorithm supported	OCTETSTRING [1]	
TSPX_CphAlgDef	default ciphering algorithm	OCTETSTRING [1]	
TSPX_NoOfHoAccA	number of handover access bursts, value range 10 - 20	INTEGER	
TSPX_NoOfHoAccB	number of handover access bursts, value range 10 - 20	INTEGER	
TSPX_NoOfHoAccC	number of handover access bursts, value range 10 - 20	INTEGER	
TSPX_NoOfHoAccD	number of handover access bursts, value range 5 - 10	INTEGER	
TSPX_NoOfHoAccE	number of handover access bursts, value range 5 - 10	INTEGER	
TSPX_NoOfHoAccF	number of handover access bursts, value range 5 - 10	INTEGER	
TSPX_NoOfHoAccG	number of handover access bursts, value range 2 - 5	INTEGER	
TSPX_NoOfHoAccH	number of handover access bursts, value range 2 - 5	INTEGER	
TSPX_NoOfHoAccI	number of handover access bursts, value range 2 - 5	INTEGER	

Item	Description	Type/Allowed values	Value chosen
TSPX_HoRefA	Hand over reference, GSM 04.08, 10.5.2.15	BITSTRING [8]	
TSPX_HoRefB	Hand over reference, GSM 04.08, 10.5.2.15	BITSTRING [8]	
TSPX_HoRefC	Hand over reference, GSM 04.08, 10.5.2.15	BITSTRING [8]	
TSPX_HoRefD	Hand over reference, GSM 04.08, 10.5.2.15	BITSTRING [8]	
TSPX_HoRefE	Hand over reference, GSM 04.08, 10.5.2.15	BITSTRING [8]	
TSPX_HoRefF	Hand over reference, GSM 04.08, 10.5.2.15	BITSTRING [8]	
TSPX_HoRefG	Hand over reference, GSM 04.08, 10.5.2.15	BITSTRING [8]	
TSPX_HoRefH	Hand over reference, GSM 04.08, 10.5.2.15	BITSTRING [8]	
TSPX_HoRefI	Hand over reference, GSM 04.08, 10.5.2.15	BITSTRING [8]	
TSPX_HoRefJ	Hand over reference, GSM 04.08, 10.5.2.15	BITSTRING [8]	
TSPX_HSN	Hopping sequence number value range: 0 - 63	INTEGER	
TSPX_IMSI	IMSI of the MS	HEXSTRING	
TSPX_Ki	default authentication key used in testing	BITSTRING	
TSPX_MAIO	mobile allocation index offset, value range: 0 - 63	INTEGER	
TSPX_MaxRetrans	Max-Retrans	INTEGER	
TSPX_MSTxpwrMax	maximum output power from MS	0..31	
TSPX_PwrlvlA	power level, value between 2 to 15	INTEGER	
TSPX_PwrlvlB	power level, value between 2 to 15	INTEGER	
TSPX_PwrlvlC	power level, value between 2 to 15	INTEGER	
TSPX_PwrlvlD	power level, value between 2 to 15	INTEGER	
TSPX_RANDA	challenge RAND	BITSTRING [128]	
TSPX_RANDB	challenge RAND	BITSTRING [128]	
TSPX_RANDC	challenge RAND	BITSTRING [128]	
TSPX_RANDDef	default challenge RAND	BITSTRING [128]	
TSPX_SDCCH4SubA	TDMA offset of SDCCH/4 subchannel	BITSTRING[2]	
TSPX_SDCCH4SubB	TDMA offset of SDCCH/4 subchannel	BITSTRING[2]	
TSPX_SDCCH4SubC	TDMA offset of SDCCH/4 subchannel	BITSTRING[2]	
TSPX_SDCCH4SubDef	TDMA offset of default SDCCH/4 subchannel	BITSTRING[2]	

Item	Description	Type/Allowed values	Value chosen
TSPX_SDCCH8SubA	TDMA offset of SDCCH/8 subchannel	BITSTRING[3]	
TSPX_SDCCH8SubB	TDMA offset of SDCCH/8 subchannel	BITSTRING[3]	
TSPX_SDCCH8SubC	TDMA offset of SDCCH/8 subchannel	BITSTRING[3]	
TSPX_SDCCH8SubD	TDMA offset of SDCCH/8 subchannel	BITSTRING[3]	
TSPX_SDCCH8SubE	TDMA offset of SDCCH/8 subchannel	BITSTRING[3]	
TSPX_SDCCH8SubF	TDMA offset of SDCCH/8 subchannel	BITSTRING[3]	
TSPX_SDCCH8SubG	TDMA offset of SDCCH/8 subchannel	BITSTRING[3]	
TSPX_SDCCH8SubDef	TDMA offset of default SDCCH/8 subchannel	BITSTRING[3]	
TSPX_TimadvA	timing advance	INTEGER	
TSPX_TimadvB	timing advance	INTEGER	
TSPX_TimadvC	timing advance	INTEGER	
TSPX_TmSltA	time slot	BITSTRING [3]	
TSPX_TmSltB	time slot	BITSTRING [3]	
TSPX_TmSltC	time slot	BITSTRING [3]	
TSPX_TmSltD	time slot	BITSTRING [3]	
TSPX_TmSltE	time slot	BITSTRING [3]	
TSPX_TmSltF	time slot	BITSTRING [3]	
TSPX_TmSltG	time slot	BITSTRING [3]	
TSPX_TmSltDef	default time slot	BITSTRING [3]	
TSPX_TmSltNotZero	time slot, arbitrarily value, but not zero.	BITSTRING [3]	
TSPX_TmSltNotZero1	time slot, arbitrarily value, but not zero and not TSPX_TmSltNotZero	BITSTRING [3]	
TSPX_Txint	Tx-Integer	INTEGER	
TSPX_TscA	training sequence code	BITSTRING [3]	
TSPX_TscB	training sequence code	BITSTRING [3]	
TSPX_TscC	training sequence code	BITSTRING [3]	
TSPX_TscD	training sequence code	BITSTRING [3]	
TSPX_TscE	training sequence code	BITSTRING [3]	
TSPX_TscF	training sequence code	BITSTRING [3]	

Item	Description	Type/Allowed values	Value chosen
TSPX_TscG	training sequence code	BITSTRING [3]	
TSPX_TscDef	default TSC	BITSTRING [3]	
TSPX_T3122	value of timer T3122, HEXSTRING[2]	HEXSTRING [2]	
TSPX_TCHcarrierA_ho	the value can be choosen arbitrarily from cell allocation of cell A (GSM), but not BCCH carrier .possible value is : 10, 17, 26, 34, 42, 45, 46, 52, 59, 66, 73, 74, 75, 76, 108, 114	INTEGER	
TSPX_TCHcarrierA_hod	the value can be choosen arbitrarily from cell allocation of cell A (DCS), but not BCCH carrier . possible value is : 734,741,754,759,762,766,767, 773,775,779,782,791,798,829, 832,844	INTEGER	
TSPX_TCHcarrierB	TCH and SDCCH channel frequency number of cell B (GSM)	INTEGER	
TSPX_TCHcarrierBd	TCH and SDCCH channel frequency number of cell B (DCS)	INTEGER	
TSPX_TCHcarrierB_ho	the value can be choosen arbitrarily from cell allocation of cell B (GSM), but not BCCH carrier . possibles value is : 14,18,22,24,30,31,38,60,66,73 ,74,75,76,108,114	INTEGER	
TSPX_TCHcarrierB_hod	not BCCH carrier of cell B. the value can be choosen arbitrarily from cell allocation of cell B (DCS), which is (739,743,746,749,756,758,761 ,771,779,782,791,798,829,832 ,844)	INTEGER	
TSPX_TCHcarrierB2_ho	Choosen arbitrarly from cell allocation B for GSM HO cases, but not BCCH carrier!	INTEGER	
TSPX_TCHcarrierB2_hod	Choosen arbitrarly from cell allocation B for DCS HO cases, but not BCCH carrier!	INTEGER	
TSPX_TCHHSubA	TDMA offset of half rate subchannel	BITSTRING[1]	
TSPX_TCHHSubDef	TDMA offset of default half rate subchannel	BITSTRING[1]	
TSPX_TMSI	TMSI of the MS used in test	OCTETSTRING[4]	
TSPX_TMSI1	another TMSI used in test which shall differ from TSPX_TMSI, TSPX_TMSI + '01'O TSPX_TMSI + '02'O TSPX_TMSI + '03'O	OCTETSTRING[4]	
TSPX_k	timing of cell A before cell B k bit periods for TC_26_6_5_4, TC_26_6_5_3.	INTEGER	

Item	Description	Type/Allowed values	Value chosen
TSPX_y	timing advance for TC_26_6_5_4, TC_26_6_5_3. the values of TSPX_k and TSPX_y shall be 0 < (2*TSPX_k + TSPX_y) MOD 256 < 60.	INTEGER	
TSPX_k1	timing of cell A before cell B k bit periods for TC_26_6_5_5	INTEGER	
TSPX_k2	timing of cell A before cell B k bit periods for TC_26_6_5_6	INTEGER	
TSPX_y2	timing advance for TC_26_6_5_6, value range : 11 - 62.	INTEGER	
TSPX_k3	timing of cell A before cell B k bit periods for TC_26_6_5_7	INTEGER	
TSPX_y3	timing advance for TC_26_6_5_7	INTEGER	
TSPX_nPara	the n'th ChReq for TC_26_6_1_2, shall be choose to [1..8], MAXRETRANS for TC_26_6_1_2 is 7 (See 11.10 for more explaination).	INTEGER	
TSPX_kPara	the k'th ChReq for TC_26_6_1_2	INTEGER	
TSPX_rPara	the r'th ChReq for TC_26_6_1_2	INTEGER	
TSPX_i1Para	for TC_26_1_2	INTEGER	
TSPX_i2Para	for TC_26_1_2	INTEGER	
TSPX_i3Para	the r'th ChReq for TC_26_6_1_2	INTEGER	
TSPX_n1Para	the n'th ChReq for TC_26_6_1_3	INTEGER	
TSPX_i4Para	for TC_26_6_1_3	INTEGER	
TSPX_xPara	t3122 for TC_26_6_1_3	INTEGER	
TSPX_AGBLKS1	BS-AG-BLKS-RES for TC_26_6_2_3_1, TC_26_6_2_1_1	INTEGER	
TSPX_PAMFRMS1	BS-PA-MFRMS for TC_26_6_2_3_1 (shall not be set to 9), TC_26_6_2_1_1	INTEGER	
TSPX_CcchConf1	paging subchannel for TC_26_6_2_3_1, TC_26_6_2_1_1	BITSTRING[3]	
TSPX_PgSubch	paging subchannel for TC_26_6_2_3_1	INTEGER	
TSPX_AGBLKS2	BS-AG-BLKS-RES for TC_26_6_2_3_2, TC_26_6_2_1_2	INTEGER	
TSPX_PAMFRMS2	BS-PA-MFRMS for TC_26_6_2_3_2, TC_26_6_2_1_2	INTEGER	

Item	Description	Type/Allowed values	Value chosen
TSPX_CcchConf2	paging subchannel for TC_26_6_2_3_2, TC_26_6_2_1_2	BITSTRING[3]	
TSPX_AGBLKS3	BS-AG-BLKS-RES for TC_26_6_2_4, TC_26_6_2_2, TC_26_6_2_1_3	INTEGER	
TSPX_PAMFRMS3	BS-PA-MFRMS for TC_26_6_2_4, TC_26_6_2_2, TC_26_6_2_1_3	INTEGER	
TSPX_CcchConf3	paging subchannel for TC_26_6_2_4, TC_26_6_2_2, TC_26_6_2_1_3	BITSTRING[3]	
TSPX_AGBLKS4	BS-AG-BLKS-RES for TC_26_6_2_5	INTEGER	
TSPX_PAMFRMS4	BS-PA-MFRMS for TC_26_6_2_5	INTEGER	
TSPX_CcchConf4	paging subchannel for TC_26_6_2_5 (shall be in the set ('010', '100', '110'))	BITSTRING[3]	
TSPX_Chtp1	channel type for TC_26_6_13_1, any value supported by the MS	BITSTRING[5]	
TSPX_ChMod1	channel mode for TC_26_6_13_1, any value for the channel type of TSPX_Chtp1	BITSTRING[8]	
TSPX_Tm1	a value to calculate the starting time for TC_26_6_13_1, value between 60 -100	INTEGER	
TSPX_Maio1	hopping parameter for TC_26_6_13_1, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma1, default : 2	INTEGER	
TSPX_Ma1	mobile allocation for TC_26_6_13_1, its value shall indicate number of frequencies between 1 and 17. default : '010101'O	OCTETSTRING[3]	
TSPX_Hsn1	hopping parameter for TC_26_6_13_1. default: 1	INTEGER	
TSPX_Maio2	hopping parameter for TC_26_6_13_1, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma2, default : 8	INTEGER	
TSPX_Ma2	mobile allocation for TC_26_6_13_1, its value shall indicate number of frequencies between 1 and 17. default :'01362A5'O	OCTETSTRING[3]	
TSPX_Hsn2	hopping parameter for TC_26_6_13_1, default : 6	INTEGER	

Item	Description	Type/Allowed values	Value chosen
TSPX_Maio3	hopping parameter for TC_26_6_13_1, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma3, default : 14	INTEGER	
TSPX_Ma3	mobile allocation for TC_26_6_13_1, its value shall indicate number of frequencies between 1 and 17. default :'01FFFA'O	OCTETSTRING[3]	
TSPX_Hsn3	hopping parameter for TC_26_6_13_1, default : 0	INTEGER	
TSPX_Chtp2	channel type for TC_26_6_13_2, any value supported by the MS	BITSTRING[5]	
TSPX_ChMod2	channel mode for TC_26_6_13_2	BITSTRING[8]	
TSPX_Maio4	hopping parameter for TC_26_6_13_2, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma4, default : 3	INTEGER	
TSPX_Ma4	mobile allocation for TC_26_6_13_2, its value shall indicate number of frequencies between 1 and 17. default : '001141'O	OCTETSTRING[3]	
TSPX_Hsn4	hopping parameter for TC_26_6_13_2, default : 0	INTEGER	
TSPX_Maio5	hopping parameter for TC_26_6_13_2, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma5, default : 9	INTEGER	
TSPX_Ma5	mobile allocation for TC_26_6_13_2, its value shall indicate number of frequencies between 1 and 17. default : '01ABCB'O	OCTETSTRING[3]	
TSPX_Hsn5	hopping parameter for TC_26_6_13_2, default : 16	INTEGER	
TSPX_Chtp3	channel type for TC_26_6_13_3, any value supported by the MS	BITSTRING[5]	
TSPX_Maio6	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma6, default : 4	INTEGER	
TSPX_Ma6	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 1 and 17. default :'006248'O	OCTETSTRING[3]	
TSPX_Hsn6	hopping parameter for TC_26_6_13_3, default : 4	INTEGER	

Item	Description	Type/Allowed values	Value chosen
TSPX_Maio7	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma7, default : 10	INTEGER	
TSPX_Ma7	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 2 and 17. default : '009AFB'O	OCTETSTRING[3]	
TSPX_Chtp4	channel type for TC_26_6_13_3, any value supported by the MS	BITSTRING[5]	
TSPX_Maio8	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma8, default : 1	INTEGER	
TSPX_Ma8	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 2 and 17. default : '002800'O	OCTETSTRING[3]	
TSPX_Hsn8	hopping parameter for TC_26_6_13_3, default : 40	INTEGER	
TSPX_Maio9	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma9, default : 15	INTEGER	
TSPX_Ma9	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 2 and 17. default : '01FFFB'O	OCTETSTRING[3]	
TSPX_Hsn9	hopping parameter for TC_26_6_13_3, default : 8	INTEGER	
TSPX_Chtp5	channel type for TC_26_6_13_4	BITSTRING[5]	
TSPX_Maio10	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma10, default : 5	INTEGER	
TSPX_Ma10	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 1 and 17. default : '01D082'O	OCTETSTRING[3]	
TSPX_Hsn10	hopping parameter for TC_26_6_13_4, default: 0	INTEGER	

Item	Description	Type/Allowed values	Value chosen
TSPX_Maio11	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma11, default : 11	INTEGER	
TSPX_Ma11	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 2 and 17. default : '00CFF3'O	OCTETSTRING[3]	
TSPX_Chtp6	channel type for TC_26_6_13_4	BITSTRING[5]	
TSPX_Maio12	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma12, default : 1	INTEGER	
TSPX_Ma12	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 1 and 17. default : '010100'O	OCTETSTRING[3]	
TSPX_Hsn12	hopping parameter for TC_26_6_13_4, default : 39	INTEGER	
TSPX_Maio13	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma13, default : 6	INTEGER	
TSPX_Ma13	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 1 and 17. default : '00E690'O	OCTETSTRING[3]	
TSPX_Hsn13	hopping parameter for TC_26_6_13_4, default: 42	INTEGER	
TSPX_Chtp7	channel type for TC_26_6_13_5	BITSTRING[5]	
TSPX_ChMod4	channel mode for TC_26_6_13_5	BITSTRING[8]	
TSPX_Tm2	a value to calculate the starting time for TC_26_6_13_5, value between 60 -100	INTEGER	
TSPX_Maio14	hopping parameter for TC_26_6_13_5, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma14, default : 5	INTEGER	
TSPX_Ma14	mobile allocation for TC_26_6_13_5, its value shall indicate number of frequencies between 1 and 17. default : '014A28'O	OCTETSTRING[3]	
TSPX_Hsn14	hopping parameter for TC_26_6_13_5, default : 50	INTEGER	

Item	Description	Type/Allowed values	Value chosen
TSPX_Maio15	hopping parameter for TC_26_6_13_5, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma15, default : 12	INTEGER	
TSPX_Ma15	mobile allocation for TC_26_6_13_5, its value shall indicate number of frequencies between 1 and 16. default : '00FF79'O	OCTETSTRING[3]	
TSPX_Hsn15	hopping parameter for TC_26_6_13_5, default : 33	INTEGER	
TSPX_Maio16	hopping parameter for TC_26_6_13_5, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma16, default : 2	INTEGER	
TSPX_Ma16	mobile allocation for TC_26_6_13_5, its value shall indicate number of frequencies between 1 and 16. default : '001110'O	OCTETSTRING[3]	
TSPX_Hsn16	hopping parameter for TC_26_6_13_5, default : 21	INTEGER	
TSPX_Chtp8	channel type for TC_26_6_13_6	BITSTRING[5]	
TSPX_ChMod5	channel mode for TC_26_6_13_6	BITSTRING[8]	
TSPX_Maio17	hopping parameter for TC_26_6_13_6, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma17, default : 7	INTEGER	
TSPX_Ma17	mobile allocation for TC_26_6_13_6, its value shall indicate number of frequencies between 1 and 17. default : '00E6A1'O	OCTETSTRING[3]	
TSPX_Hsn17	hopping parameter for TC_26_6_13_6, default : 22	INTEGER	
TSPX_Maio18	hopping parameter for TC_26_6_13_6, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma18, default : 13	INTEGER	
TSPX_Ma18	mobile allocation for TC_26_6_13_6, its value shall indicate number of frequencies between 1 and 16. default : '00FFF9'O	OCTETSTRING[3]	
TSPX_Hsn18	hopping parameter for TC_26_6_13_6, default : 9	INTEGER	

Item	Description	Type/Allowed values	Value chosen
TSPX_Maio19	hopping parameter for TC_26_6_13_6, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma19, default : 3	INTEGER	
TSPX_Ma19	mobile allocation for TC_26_6_13_6, its value shall indicate number of frequencies between 1 and 16. default : '001111'O	OCTETSTRING[3]	
TSPX_Hsn19	hopping parameter for TC_26_6_13_6, default : 44	INTEGER	
TSPX_Chtp9	channel type for TC_26_6_13_7	BITSTRING[5]	
TSPX_ChMod6	channel mode for TC_26_6_13_7	BITSTRING[8]	
TSPX_Maio20	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma20, default : 8	INTEGER	
TSPX_Ma20	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 17. default : '016699'O	OCTETSTRING[3]	
TSPX_Hsn20	hopping parameter for TC_26_6_13_7, default : 30	INTEGER	
TSPX_Maio21	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma21, default : 14	INTEGER	
TSPX_Ma21	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 17. default : '01FEFB'O	OCTETSTRING[3]	
TSPX_Chtp10	channel type for TC_26_6_13_7	BITSTRING[5]	
TSPX_Maio22	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma22, default : 4	INTEGER	
TSPX_Ma22	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 16. default : '006241'O	OCTETSTRING[3]	
TSPX_Hsn22	hopping parameter for TC_26_6_13_7, default : 11	INTEGER	

Item	Description	Type/Allowed values	Value chosen
TSPX_Maio23	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma23, default : 11	INTEGER	
TSPX_Ma23	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 16. default : '00FF1B'O	OCTETSTRING[3]	
TSPX_Hsn23	hopping parameter for TC_26_6_13_7, default : 60	INTEGER	
TSPX_Chtp11	channel type for TC_26_6_13_8	BITSTRING[5]	
TSPX_ChMod7	channel mode for TC_26_6_13_8	BITSTRING[8]	
TSPX_Maio24	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma24, default : 9	INTEGER	
TSPX_Ma24	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 17. default : '00E6E9'O	OCTETSTRING[3]	
TSPX_Hsn24	hopping parameter for TC_26_6_13_8, default : 7	INTEGER	
TSPX_Maio25	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma25, default : 15	INTEGER	
TSPX_Ma25	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 17. default : '01FFFB'O	OCTETSTRING[3]	
TSPX_Chtp12	channel type for TC_26_6_13_8	BITSTRING[5]	
TSPX_Maio26	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma26, default : 5	INTEGER	
TSPX_Ma26	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 16. default : '009168'O	OCTETSTRING[3]	
TSPX_Hsn26	hopping parameter for TC_26_6_13_8, default : 9	INTEGER	

Item	Description	Type/Allowed values	Value chosen
TSPX_Maio27	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma27, default : 1	INTEGER	
TSPX_Ma27	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 16. default : '004080'O	OCTETSTRING[3]	
TSPX_Hsn27	hopping parameter for TC_26_6_13_8, default : 38	INTEGER	
TSPX_Chtp13	channel type for TC_26_6_13_9	BITSTRING[5]	
TSPX_Tm3	channel mode for TC_26_6_13_9, value between 60 -100	INTEGER	
TSPX_Maio28	hopping parameter for TC_26_6_13_9, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma28, default : 10	INTEGER	
TSPX_Ma28	mobile allocation for TC_26_6_13_9, its value shall indicate number of frequencies between 1 and 17. default : '00EE7A'O	OCTETSTRING[3]	
TSPX_Hsn28	hopping parameter for TC_26_6_13_9, default : 11	INTEGER	
TSPX_Maio29	hopping parameter for TC_26_6_13_9, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma29, default : 6	INTEGER	
TSPX_Ma29	mobile allocation for TC_26_6_13_9, its value shall indicate number of frequencies between 1 and 17. default : '00F070'O	OCTETSTRING[3]	
TSPX_Chtp14	channel type for TC_26_6_13_10	BITSTRING[5]	
TSPX_Maio30	hopping parameter for TC_26_6_13_10, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma30, default : 11	INTEGER	
TSPX_Ma30	mobile allocation for TC_26_6_13_10, its value shall indicate number of frequencies between 1 and 17. default : '017F69'O	OCTETSTRING[3]	
TSPX_Hsn30	hopping parameter for TC_26_6_13_10, default : 62	INTEGER	

Item	Description	Type/Allowed values	Value chosen
TSPX_Maio31	hopping parameter for TC_26_6_13_10, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma31, default : 1	INTEGER	
TSPX_Ma31	mobile allocation for TC_26_6_13_10, its value shall indicate number of frequencies between 1 and 17. default : '000101'O	OCTETSTRING[3]	
TSPX_MOBscSvcA	any supported MO basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MOBscSvcB	any supported MO basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MOBscSvcC	any supported MO basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MOBscSvcD	any supported MO basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MOBscSvcE	any supported MO basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MOBscSvcF	any supported MO basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MOBscSvcG	any supported MO basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MOBscSvcH	any supported MO basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MOBscSvcI	any supported MO basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MOBscSvcJ	any supported MO basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MT_DualModSvc	any supported MT dual mode service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MT_NonSptSvc	any not supported MT basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MTBscSvcA	any supported MT basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MTBscSvcB	any supported MT basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MTBscSvcC	any supported MT basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MTBscSvcD	any supported MT basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MTBscSvcE	any supported MT basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MTBscSvcF	any supported MT basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MTBscSvcG	any supported MT basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MTBscSvcH	any supported MT basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MTBscSvcI	any supported MT basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MTBscSvcJ	any supported MT basic service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MTNIC_BscSvcA	any supported MT basic service without immediate connection	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MTNIC_BscSvcB	any supported MT basic service without immediate connection	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MTNIC_BscSvcC	any supported MT basic service without immediate connection	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MTNIC_BscSvcD	any supported MT basic service without immediate connection	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MTNIC_BscSvcE	any supported MT basic service without immediate connection	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MTNIC_BscSvcF	any supported MT basic service without immediate connection	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MTNIC_BscSvcG	any supported MT basic service without immediate connection	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MTNIC_BscSvcH	any supported MT basic service without immediate connection	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MTNIC_BscSvcI	any supported MT basic service without immediate connection	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MTNIC_BscSvcJ	any supported MT basic service without immediate connection	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_Telephony_Rate	channel rate for TS11, default value : C_Full	"F", "H"	
TSPX_MOChRateA	channel rate for TSPX_MOBscSvcA, default value : C_Full	"F", "H"	
TSPX_MOChRateB	channel rate for TSPX_MOBscSvcB, default value : C_Full	"F", "H"	
TSPX_MOChRateC	channel rate for TSPX_MOBscSvcC, default value : C_Full	"F", "H"	
TSPX_MOChRateD	channel rate for TSPX_MOBscSvcD, default value : C_Full	"F", "H"	
TSPX_MOChRateE	channel rate for TSPX_MOBscSvcE, default value : C_Full	"F", "H"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MOChRateF	channel rate for TSPX_MOBscSvcF, default value : C_Full	"F", "H"	
TSPX_MOChRateG	channel rate for TSPX_MOBscSvcG, default value : C_Full	"F", "H"	
TSPX_MOChRateH	channel rate for TSPX_MOBscSvcH, default value : C_Full	"F", "H"	
TSPX_MOChRateI	channel rate for TSPX_MOBscSvcI, default value : C_Full	"F", "H"	
TSPX_MOChRateJ	channel rate for TSPX_MOBscSvcJ, default value : C_Full	"F", "H"	
TSPX_MTChRateA	channel rate for TSPX_MTBscSvcA, default value : C_Full	"F", "H"	
TSPX_MTChRateB	channel rate for TSPX_MTBscSvcB, default value : C_Full	"F", "H"	
TSPX_MTChRateC	channel rate for TSPX_MTBscSvcC, default value : C_Full	"F", "H"	
TSPX_MTChRateD	channel rate for TSPX_MTBscSvcD, default value : C_Full	"F", "H"	
TSPX_MTChRateE	channel rate for TSPX_MTBscSvcE, default value : C_Full	"F", "H"	
TSPX_MTChRateF	channel rate for TSPX_MTBscSvcF, default value : C_Full	"F", "H"	
TSPX_MTChRateG	channel rate for TSPX_MTBscSvcG, default value : C_Full	"F", "H"	
TSPX_MTChRateH	channel rate for TSPX_MTBscSvcH, default value : C_Full	"F", "H"	
TSPX_MTChRateI	channel rate for TSPX_MTBscSvcI, default value : C_Full	"F", "H"	
TSPX_MTChRateJ	channel rate for TSPX_MTBscSvcJ, default value : C_Full	"F", "H"	
TSPX_MTNIC_ChRateA	channel rate for TSPX_MTNIC_BscSvcA, default value : C_Full	"F", "H"	
TSPX_MTNIC_ChRateB	channel rate for TSPX_MTNIC_BscSvcB, default value : C_Full	"F", "H"	
TSPX_MTNIC_ChRateC	channel rate for TSPX_MTNIC_BscSvcC, default value : C_Full	"F", "H"	
TSPX_MTNIC_ChRateD	channel rate for TSPX_MTNIC_BscSvcD, default value : C_Full	"F", "H"	
TSPX_MTNIC_ChRateE	channel rate for TSPX_MTNIC_BscSvcE, default value : C_Full	"F", "H"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MTNIC_ChRateF	channel rate for TSPX_MTNIC_BscSvcF, default value : C_Full	"F", "H"	
TSPX_MTNIC_ChRateG	channel rate for TSPX_MTNIC_BscSvcG, default value : C_Full	"F", "H"	
TSPX_MTNIC_ChRateH	channel rate for TSPX_MTNIC_BscSvcH, default value : C_Full	"F", "H"	
TSPX_MTNIC_ChRateI	channel rate for TSPX_MTNIC_BscSvcI, default value : C_Full	"F", "H"	
TSPX_MTNIC_ChRateJ	channel rate for TSPX_MTNIC_BscSvcJ, default value : C_Full	"F", "H"	
TSPX_EmgCallRate	Rate for the basic service supported for MO emergency calls	"F", "H"	
TSPX_MO_NonCallSS	any supported MO non-call releated supplementary service	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	

Item	Description	Type/Allowed values	Value chosen
TSPX_MO_BscSvc_SMS	any supported MO SMS calls	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MO_DualModSvc	any supported MO dual mode call	"C_Telephony", "C_EmgCallSRV", "C_AltSpchG3", "C_AutoG3", "C_300cda", "C_1200cda", "C_120075cda", "C_2400cda", "C_4800cda", "C_2400cds", "C_4800cds", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Pkt2400", "C_Pkt4800", "C_Pkt9600", "C_AltSpchData", "C_SpchData"	
TSPX_MO_DualModRate	Rate for the supported MO dual mode call in TSPX_MO_DualModSvc	"F", "H"	

Annex C (normative): PCTR Proforma

Notwithstanding the provisions of the copyright clause related to the text of this ETS, ETSI grants that users of this ETS may freely reproduce the PCTR proforma in this annex so that it can be used for its intended purposes and may further publish the completed PCTR.

PROTOCOL
Conformance Test Report
(PCTR)

Global System for Mobile Communication, GSM,
User-Network Access

Layer 3 Signalling Functions

Test Candidate	
Name :	SUT name
Model :	model
H/W version :	hw
S/W version :	sw
Serial No. :	serienr

Client	
Name :	
Street / No. :	
Postal Code / City:	
Country :	

This Test Report shall not be reproduced except in full without the written permission of TEST LAB REFERENCE, and shall not be quoted out of context.

Annex D (normative): Enhanced Full Rate Abstract Test Suite

The machine processable ATS

The electronic forms of the machine processable files (TTCN MP format) corresponding to the ATS for EFR are contained in the file ef-4m1.mp[2)] (EFR) on the diskette included as a part of this ETS (Part 3).

The graphical form ATS

The TTCN.GR representation of this ATS is contained in the remainder of this annex.

2) This file is located in an archive file named 6073_E8.ZIP. Other file formats are available on request.

Suite overview

Suite structure

Test Suite Structure	
Suite Name:	GSM_MS_EFR_4220
Standards Ref:	ETS 300 557
PICS Ref:	
PIXIT Ref:	
Test Method(s):	Distributed Single Layer Test Method
Comments:	EFR ATS Version 4.22.0

Structure and Objectives			
Test Group Reference	**Selection Ref**	**Test Group Objective**	**Page Nr**
GSM_MS_EFR_4220/EFR/	SelExpr_0005	Enhanced Full Rate signalling	1823
Detailed Comments:			

Test case index

Test Case Index				
Test Group Reference	**Test Case Id**	**Selection ref**	**Description**	**Page Nr**
GSM_MS_EFR_4220 /EFR/	TC_26_12_1	SelExpr_0000	EFR signalling/ test of the channel mode modify procedure	1823
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_1	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1826
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_2	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1828
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_3	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1830
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_4	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1832
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_5	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1834
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_6	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1836
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_7	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1838
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_8	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1840
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_9	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1842
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_10	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1844
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_11	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1846
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_12	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1848
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_13	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1850
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_14	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1852
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_1_15	SelExpr_0001	EFR signalling/ Handover/ active call/ successful case	1854
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_2_1	SelExpr_0001	EFR signalling/ Handover/ successful/ call under establishment/ non-synchronized	1856
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_2_2	SelExpr_0001	EFR signalling/ Handover/ successful/ call under establishment/ non-synchronized	1858
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_2_3	SelExpr_0001	EFR signalling/ Handover/ successful/ call under establishment/ non-synchronized	1860
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_2_4	SelExpr_0001	EFR signalling/ Handover/ successful/ call under establishment/ non-synchronized	1862
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_2_5	SelExpr_0001	EFR signalling/ Handover/ successful/ call under establishment/ non-synchronized	1864
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_2_6	SelExpr_0001	EFR signalling/ Handover/ successful/ call under establishment/ non-synchronized	1866
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_2_7	SelExpr_0001	EFR signalling/ Handover/ successful/ call under establishment/ non-synchronized	1868
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_2_8	SelExpr_0001	EFR signalling/ Handover/ successful/ call under establishment/ non-synchronized	1870
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_2_9	SelExpr_0001	EFR signalling/ Handover/ successful/ call under establishment/ non-synchronized	1872
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_2_10	SelExpr_0001	EFR signalling/ Handover/ successful/ call under establishment/ non-synchronized	1874
GSM_MS_EFR_4220 /EFR/	TC_26_12_2_2_11	SelExpr_0001	EFR signalling/ Handover/ successful/ call under establishment/ non-synchronized	1876
GSM_MS_EFR_4220 /EFR/	TC_26_12_3	SelExpr_0001	EFR Signalling/ Structured procedures/ MS originated call/ late assignment	1878
GSM_MS_EFR_4220 /EFR/	TC_26_12_4	SelExpr_0002	Structured procedures/ MS terminated call/ early assignment	1881
GSM_MS_EFR_4220 /EFR/	TC_26_12_5	SelExpr_0003	Structured procedures/ emergency call	1884
Detailed Comments:				

Test step index

Test Step Index			
Test Step Group Reference	**Test Step Id**	**Description**	**Page Nr**
GSM_MS_EFR_4220/management/ChConfig/	Config_CBMS_A_1		1886
GSM_MS_EFR_4220/management/ChConfig/	Config_FACCHF_A_1	To set one physical channel used as TCHF_ACCH's of cell A.	1887
GSM_MS_EFR_4220/management/ChConfig/	Config_FACCHF_A_2	To set one physical channel used as TCHF_ACCH's for instance 2 of cell A.	1887
GSM_MS_EFR_4220/management/ChConfig/	Config_FACCHF_B_1	To set one physical channel used as TCHF_ACCH's of cell B.	1887
GSM_MS_EFR_4220/management/ChConfig/	Config_FACCHH_A_1	To set one physical channel used as TCHH_ACCH's of cell A.	1888
GSM_MS_EFR_4220/management/ChConfig/	Config_FACCHH_A_2	To set one physical channel used as TCHH_ACCH's of cell A.	1888
GSM_MS_EFR_4220/management/ChConfig/	Config_FACCHH_B_1	To set one physical channel used as TCHH_ACCH's of cell B.	1888
GSM_MS_EFR_4220/management/ChConfig/	Config_BCCH_CCCH_A_1	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell A.	1889
GSM_MS_EFR_4220/management/ChConfig/	Config_BCCH_CCCH_A_ho_1		1889
GSM_MS_EFR_4220/management/ChConfig/	Config_BCCH_CCCH_A_2	To set a physical channel and map the second BCCH, CCCH onto the physical channel which represents cell A.	1889
GSM_MS_EFR_4220/management/ChConfig/	Config_BCCH_CCCH_A_3	To set a physical channel and map the third BCCH, CCCH onto the physical channel which represents cell A.	1890
GSM_MS_EFR_4220/management/ChConfig/	Config_BCCH_CCCH_A_4	To set a physical channel and map the fourth BCCH, CCCH onto the physical channel which represents cell A.	1890
GSM_MS_EFR_4220/management/ChConfig/	Config_BCCH_CCCH_B_1	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell B.	1890
GSM_MS_EFR_4220/management/ChConfig/	Config_BCCH_CCCH_B_ho_1		1891
GSM_MS_EFR_4220/management/ChConfig/	Config_SDCCH4_A_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.	1891
GSM_MS_EFR_4220/management/ChConfig/	Config_SDCCH4_A_ho_1		1891
GSM_MS_EFR_4220/management/ChConfig/	Config_SDCCH4_B_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.	1892
GSM_MS_EFR_4220/management/ChConfig/	Config_SDCCH4_B_ho_1		1892
GSM_MS_EFR_4220/management/ChConfig/	Config_SDCCH4_C_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell C	1893
GSM_MS_EFR_4220/management/ChConfig/	Config_SDCCH8_A_1		1893
GSM_MS_EFR_4220/management/ChConfig/	CombinedBCCH_A_CBMS		1894
GSM_MS_EFR_4220/management/ChConfig/	CombinedBCCH_A	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.	1894
GSM_MS_EFR_4220/management/ChConfig/	CombinedBCCH_A_ho		1894
GSM_MS_EFR_4220/management/ChConfig/	CombinedBCCH_B	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.	1895
GSM_MS_EFR_4220/management/ChConfig/	CombinedBCCH_B_ho		1895
GSM_MS_EFR_4220/management/ChConfig/	CombinedBCCH_C	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell C.	1895
GSM_MS_EFR_4220/management/ChConfig/	NonCombinedBCCH_A	To set one physical channel used as FCHH_SCH_BCCH_CCCH for cell A.	1896
GSM_MS_EFR_4220/management/ChConfig/	NonCombinedBCCH_A_2	To set a physical channel and map the second BCCH, CCCH onto the physical channel which	1896

		represents cell A.	
GSM_MS_EFR_4220/management/ChConfig/	NonCombinedBCCH_A_3	To set a physical channel and map the third BCCH, CCCH onto the physical channel which represents cell A.	1896
GSM_MS_EFR_4220/management/ChConfig/	NonCombinedBCCH_A_4	To set a physical channel and map the fourth BCCH, CCCH onto the physical channel which represents cell A.	1897
GSM_MS_EFR_4220/management/ChConfig/	NonCombinedBCCH_A_ho		1897
GSM_MS_EFR_4220/management/ChConfig/	NonCombinedBCCH_B	To set a physical channel and map FCCH, SCH, BCCH, CCCH onto the physical channel which represents cell B for RR testing.	1898
GSM_MS_EFR_4220/management/ChConfig/	NonCombinedBCCH_B_ho		1898
GSM_MS_EFR_4220/management/ChConfig/	FullRateCh_A_1	To set one physical channel used as TCHF_ACCH's for instance 1 of cell A.	1899
GSM_MS_EFR_4220/management/ChConfig/	FullRateCh_A_1_nociph	To set one physical channel used as TCHF_ACCH's for instance 1 of cell A.	1899
GSM_MS_EFR_4220/management/ChConfig/	FullRateCh_A_2	To set one physical channel used as TCHF_ACCH's for instance 2 of cell A for TC_26_6_13_1.	1899
GSM_MS_EFR_4220/management/ChConfig/	FullRateCh_A_2_nociph	To set one physical channel used as TCHF_ACCH's for instance 2 of cell A for TC_26_6_13_1.	1900
GSM_MS_EFR_4220/management/ChConfig/	FullRateCh_B_1	To set one physical channel used as TCHF_ACCH's for instance 1 of cell B.	1900
GSM_MS_EFR_4220/management/ChConfig/	FullRateCh_B_1_nociph	To set one physical channel used as TCHF_ACCH's for instance 1 of cell B.	1900
GSM_MS_EFR_4220/management/ChConfig/	HalfRateCh_A_1	To set one physical channel used as TCHH_ACCH's for instance 1 of cell A.	1901
GSM_MS_EFR_4220/management/ChConfig/	HalfRateCh_A_1_nociph	To set one physical channel used as TCHH_ACCH's for instance 1 of cell A.	1901
GSM_MS_EFR_4220/management/ChConfig/	HalfRateCh_A_2	To set one physical channel used as TCHH_ACCH's for instance 2 of cell A for TC_26_6_13_1.	1902
GSM_MS_EFR_4220/management/ChConfig/	HalfRateCh_A_2_nociph	To set one physical channel used as TCHH_ACCH's for instance 2 of cell A for TC_26_6_13_1.	1902
GSM_MS_EFR_4220/management/ChConfig/	HalfRateCh_B_1	To set one physical channel used as TCHH_ACCH's for instance 1 of cell B	1902
GSM_MS_EFR_4220/management/ChConfig/	HalfRateCh_B_1_nociph	To set one physical channel used as TCHH_ACCH's for instance 1 of cell B	1903
GSM_MS_EFR_4220/management/ChConfig/	SDCCH8_A_1_nociph		1903
GSM_MS_EFR_4220/Miscellaneous/	Adjust_gsmanddcs_powerlvl		1904
GSM_MS_EFR_4220/Miscellaneous/	AssCh_complete		1904
GSM_MS_EFR_4220/Miscellaneous/	AssCmdGen		1905
GSM_MS_EFR_4220/Miscellaneous/	AssCmdGen_fh		1906
GSM_MS_EFR_4220/Miscellaneous/	Authentication	To execute the authentication procedure.	1906
GSM_MS_EFR_4220/Miscellaneous/	CallProcGen	To generate a CALL PROCEEDING message .	1907
GSM_MS_EFR_4220/Miscellaneous/	CCCH_group_Paging_group		1908
GSM_MS_EFR_4220/Miscellaneous/	CC_Est_MT_Call		1909
GSM_MS_EFR_4220/Miscellaneous/	CC_EstMsTermCall		1910
GSM_MS_EFR_4220/Miscellaneous/	Ciphering_on	To switch on the ciphering.	1910
GSM_MS_EFR_4220/Miscellaneous/	EFR_ChkRcvSpchPrefs		1911
GSM_MS_EFR_4220/Miscellaneous/	EFR_SpchPrefInBCAP		1911
GSM_MS_EFR_4220/Miscellaneous/	EFR_RcvSetupOrEsetup		1912
GSM_MS_EFR_4220/Miscellaneous/	FnArith		1913
GSM_MS_EFR_4220/Miscellaneous/	ImsiDetach		1913
GSM_MS_EFR_4220/Miscellaneous/	MM_PwrOrSimOff		1914
GSM_MS_EFR_4220/Miscellaneous/	MM_PwrOrSimOn		1914
GSM_MS_EFR_4220/Miscellaneous/	RcvSetupOrEsetup		1914
GSM_MS_EFR_4220/Miscellaneous/	RR_hocomp1		1915
GSM_MS_EFR_4220/Miscellaneous/	RRmtcallprepare		1915
GSM_MS_EFR_4220/Miscellaneous/	SelectPagingCh		1916
GSM_MS_EFR_4220/Miscellaneous/	SendSeqNo_chk		1916
GSM_MS_EFR_4220/Miscellaneous/	SvcSupportedChk	To check wether the Basic Service svc is supported against PICS answers.	1917

GSM_MS_EFR_4220/Miscellaneous/	Varinit_fix		1918
GSM_MS_EFR_4220/Miscellaneous/SysInfo/	ChgLAC_A	To change the LAC of cell A.	1918
GSM_MS_EFR_4220/Miscellaneous/SysInfo/	ChgLAC_B	To change the LAC of cell B.	1918
GSM_MS_EFR_4220/Miscellaneous/SysInfo/	ChgLAI_C	To change the LAI of cell C to HPLMN.	1919
GSM_MS_EFR_4220/Miscellaneous/SysInfo/	SysInfoSending_cbms	To send system information messages for the CBMS tests.	1920
GSM_MS_EFR_4220/Miscellaneous/SysInfo/	SysInfoSending_fh	To send system information messages for the L3 tests. The following parameters specified by input parameters.	1921
GSM_MS_EFR_4220/Miscellaneous/SysInfo/	SysInfoSending_nfh	To send system information messages with default parameters defined for L3 tests for which no other special parameters indicated.	1923
GSM_MS_EFR_4220/Miscellaneous/SysInfo/	SysInfoSending_nfhCb	To send system information messages with default parameters defined for L3 tests except that the cell is barred.	1925
GSM_MS_EFR_4220/Miscellaneous/SysInfo/	SysInfoSending_e	Sending of Systeminformation for cell A in EGSM test case TC_26_10_2_2.	1926
GSM_MS_EFR_4220/Miscellaneous/SysInfo/	SysInfo_SacchSending	To send SYSTEM INFORMATION 5 and 6 messages defined by parameters 'sysinfo5_pdu' and 'sysinfo6_pdu' in the parametrised 'ch' channel.	1927
GSM_MS_EFR_4220/Miscellaneous/SysInfo/	SysInfo_5bisSending	To send SYSTEM INFORMATION 5bis message defined by parameters 'sysinfo5bis_pdu' in the parametrised 'ch' channel.	1927
GSM_MS_EFR_4220/Miscellaneous/SysInfo/	SetNECI	To set the NECI =1.	1928
GSM_MS_EFR_4220/Miscellaneous/SysInfo/	SetATT	To set the ATT flag to "MS's in the cell should apply IMSI attach and detach procedure"	1929
GSM_MS_EFR_4220/OperatorOP/	InitCall	To initiate a call of the basic service `srv`.	1930
GSM_MS_EFR_4220/OperatorOP/	InitNonCallSupp	To attempt a non call related supplementary service at the MS under test.	1930
GSM_MS_EFR_4220/OperatorOP/	AtmpShortMsg	To attempt an MO short message service transaction at the MS under test.	1930
GSM_MS_EFR_4220/OperatorOP/	CheckUssdStringDisplayed	To check whether the correct USSD String is displayed on the MS	1930
GSM_MS_EFR_4220/OperatorOP/	PLMNsCHK	To check whether the MS presents a list of available PLMNs.	1930
GSM_MS_EFR_4220/OperatorOP/	RFtransCHK	To check whether the MS transmits any radio signal.	1931
GSM_MS_EFR_4220/Postambles/	PostLinkRelEnd	To release the RR connection and bring the MS back to Idle state. This teststep decides the final verdict and therefor it shall be used only at the end of testcases.	1931
GSM_MS_EFR_4220/Postambles/	PostMainLinkRel	To release the main signalling link `ch`, and bring the MS back to Idle state.	1931
GSM_MS_EFR_4220/Postambles/	ChanRel	To release the RR connection on the channel TCV_ch and bring the MS back to Idle state.	1931
GSM_MS_EFR_4220/Postambles/	ChanRel_P	To release the RR connection on the channel TCV_chmaindcch and bring the MS back to Idle state. This teststep decides the verdict (P).	1932
GSM_MS_EFR_4220/Postambles/	ChanRel_end	To release the RR connection and bring the MS back to Idle state. This teststep decides the final verdict and therefor it shall be used only at the end of testcases.	1932
GSM_MS_EFR_4220/Preambles/	Chmod	To assign correct values to the test case variables TCV_ChMod and TCV_ChModb.	1933
GSM_MS_EFR_4220/Preambles/	EFR_BasicServiceMO		1934
GSM_MS_EFR_4220/Preambles/	EFR_BasicServiceMT		1936
GSM_MS_EFR_4220/Preambles/	Est_MO_Call_init		1937
GSM_MS_EFR_4220/Preambles/	Est_MT_Call_FH		1939
GSM_MS_EFR_4220/Preambles/	Est_MT_CallNonFH		1940
GSM_MS_EFR_4220/Preambles/	IdleUpdated	This test step is called at the preamble of each test case.	1941
GSM_MS_EFR_4220/Preambles/	PreEstRRConn		1948
GSM_MS_EFR_4220/Preambles/	StartCellA_ho		1949
GSM_MS_EFR_4220/Preambles/	StartCellB_ho		1950
Detailed Comments:			

Default index

Default Index			
Default Group Reference	**Default Id**	**Description**	**Page Nr**
GSM_MS_EFR_4220/	OtherEvents	To match unexpected events and sign final verdict for preambles.	1951
GSM_MS_EFR_4220/	OtherEventsFail	To match unexpected events and fail the test case.	1952
GSM_MS_EFR_4220/	OtherEventsFail_01	To match irrelevant CHANNEL REQUEST msg and return or match other unexpected events and fail the test case.	1953
GSM_MS_EFR_4220/	OtherEventsFail_02	To match unexpected events and fail the test case but ignore channel request messages that are sent before the lower tester has sent (and the MS received) the Immediate Assigment message.	1954
GSM_MS_EFR_4220/	OtherEvents_01	To match irrelevant messages and return	1955
GSM_MS_EFR_4220/	OtherEvents_02	To match unexpected events and sign final verdict for preambles but ignore channel request messages that are sent before the lower tester has sent (and the MS received) the Immediate Assigment message.	1956
GSM_MS_EFR_4220/	RcvHdOvAcc	To match any HANDOVER ACCESS then return to calling tree.	1957
Detailed Comments:			

Declarations Part

Test suite type definitions

Simple type definitions

Simple Type Definitions		
Type Name	**Type Definition**	**Comments**
B_1	BITSTRING [1]	Generic type for 1 bit value
B_2	BITSTRING [2]	Generic type for 2 bits value
B_3	BITSTRING [3]	Generic type for 3 bits value
B_4	BITSTRING [4]	Generic type for 4 bits value
B_5	BITSTRING [5]	Generic type for 5 bits value
B_6	BITSTRING [6]	Generic type for 6 bits value
B_8	BITSTRING [8]	Generic type for 8 bits value
BCCHFRQ	BITSTRING [5]	position of a bcch carrier in the bcch channel list
BCC	BITSTRING [3]	BS colour code
BCDN	OCTETSTRING [1..10]	BCD numbers, GSM 04.08, 10.5.4.7, octet 4-13
BSIC	BITSTRING [6]	base station identity code
CCCH_CON	BITSTRING[3]	number of physical channels for ccch GSM 04.08, 10.5.2.11
CCSTATE	INTEGER(1,3,4,6,7,8,9,10,11,12,19,26,27)	
CellID	IA5String	Cell identifier
CHANNEL	BITSTRING [2]	needed channel type
CHMOD_VAL	BITSTRING[8]	value for channel mode GSM 04.08, 10.5.2.6, 10.5.2.7
CH_TDMA	BITSTRING[5]	channel type and TDMA offset GSM 04.08, 10.5.2.5
CI	OCTETSTRING [2]	cell identity, GSM 04.08, 10.5.1.1
CKSN	BITSTRING[3]	ciphering key sequence number(only key sequence) GSM 04.08, 10.5.1.2
CLRSUP	BITSTRING('10100001'B)	CLIR suppression GSM 04.08, 10.5.4.11a
CLRINV	BITSTRING('10100010'B)	CLIR invocation GSM 04.08, 10.5.4.11b
CMSVTYPE	BITSTRING [4]	CM service type
CP_CAU	OCTETSTRING [1]	cp cause GSM 04.11, 8.1.4.2
CPHALG	BITSTRING[3]	Ciphering algorithm identifier GSM 04.08, 10.5.2.9
CS	BITSTRING [2]	coding standard
EstMode	OCTETSTRING [1]	establish mode
EXTB	BITSTRING [1]	extension bit
FCS	HEXSTRING [2]	TP failure cause
FLWCNTL	INTEGER(0, 1, 2)	type of flow control. 0--outband flow control; 1--inband flow control; 2-- no flow control
HORF	BITSTRING [8]	handover reference
HSN	BITSTRING[6]	Hopping sequence number e.g.GSM 04.08 10.5.2.5
IDTYPE	B_4	identity type
IEI_4	BITSTRING [4]	information element identifier, type 1
IEI_8	BITSTRING [8]	information element identifier, type 2-4
L2FMTYPE	INTEGER(1..3)	L2 frame type
LENGTH	OCTETSTRING [1]	IE length or L2 pseudo length. The L2 pseudo length is composed of a L2 pseudo length value, which is 6 bits long, and of 2 additional bits. See GSM 04.08 section 10.5.2.19
LOGICCH	IA5String	

LOGCH	IA5String	
MAC	OCTETSTRING[3]	mac, e.g. GSM 04.08 10.5.2.21
MAIO	BITSTRING[6]	MAIO,GSM 04.08 10.5.2.5
MAXTXPOW	INTEGER(0..31)	Max Tx Power Level GSM 04.08, 10.5.2.4
MR	OCTETSTRING [1]	SMS RP or TP message reference
MT	BITSTRING[8]	message type, structure of it see MT0
MTI	BITSTRING [3]	SMS RP message type indicator
NCC	BITSTRING [3]	PLMN colour code
NCCP	OCTETSTRING [1]	ncc permitted
NTI	BITSTRING('10000000'B, '10000001'B, '10000010'B)	notification indicator GSM 04.08, 10.5.4.20
PD	B_4	Protocol discriminator
PGG	BITSTRING [8]	the paging group of an MS to be paged in binary presentation
RATE	IA5String("F", "H", "Fv2")	
RAND	BITSTRING [128]	random number
RCSD	BITSTRING [8]	reverse call setup direction GSM 04.08, 10.5.4.22a
RelMode	OCTETSTRING [1]	release mode
REJCAU	OCTETSTRING [1]	reject cause GSM 04.08, 10.5.3.6
RPI	BITSTRING [8]	repeat indicator
RRCAU	BITSTRING [8]	RR cause GSM 04.08, 10.5.2.31
RXLEV	BITSTRING [6]	received signal strength
SAPID	OCTETSTRING [1]	sap identifier
SENDINGMODE	INTEGER(1..4)	the mode of sending two consecutive messages containing paging mode IE
SERVICES	IA5String("C_Telephony", "C_EmgCall", "C_AltSpchFax", "C_AutoFax", "C_Async300", "C_Async1200", "C_Async120075", "C_Async2400", "C_Async4800", "C_Async9600", "C_Sync1200", "C_Sync2400", "C_Sync4800", "C_Sync9600", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Packet2400", "C_Packet4800", "C_Packet9600", "C_AltSpchData", "C_SpchData")	
SHOCT	B_4	spare half octet for type 1 information element GSM 04.08, 10.5.1.8
SKI	B_4	Skip indicator
SN	BITSTRING [3]	all possible slot number, GSM 04.08 10.5.2.5
SPB	B_1	spare bit
SP2B	B_2	2 spare bits
SP3B	B_3	3 spare bits
SP5B	B_5	5 spare bits
SP6B	B_6	6 spare bits
SRES	OCTETSTRING [4]	authentication response signature
TPCD	OCTETSTRING [1..157]	TP command data
TPSCTS	HEXSTRING [14]	TP service centre time stamp
TPUD	OCTETSTRING [1..140]	TP user data
TSC	B_3	Training sequence code,GSM 04.08 10.5.2.5
T1	BITSTRING [11]	
T1_	B_5	T1'
T2	B_5	
T3	BITSTRING [6]	
T3_	B_3	T3'
TI_V	B_3	TI value
TMSI_V	OCTETSTRING[4]	TMSI value is octetstring of length 4
TZONES	INTEGER(-79..79)	time zones in 15 minute steps
WI	HEXSTRING [2]	wait indication, unit in second
Detailed Comments:		

Structured Type definitions

Structured Type Definition		
Type Name:	ACST	
Comments:	Auxiliary states (CC information element) GSM 04.08, 10.5.4.4	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb	EXTB	
sp3b	SP3B	
has	BITSTRING [2]	
mpas	BITSTRING [2]	
Detailed Comments:		

Structured Type Definition		
Type Name:	BCAP	
Comments:	Bearer capability (CC information element) GSM 04.08, 10.5.4.5	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
rchr	BITSTRING [2]	
cs	BITSTRING [1]	
tm	BITSTRING [1]	
itc	BITSTRING [3]	
oct_3a_etc_1	EFR_BCAP_3A_ETC	
oct_3a_etc_2	EFR_BCAP_3A_ETC	
oct_3a_etc_3	EFR_BCAP_3A_ETC	
oct_3a_etc_4	EFR_BCAP_3A_ETC	
oct_3a_etc_5	EFR_BCAP_3A_ETC	
oct_3a_etc_6	EFR_BCAP_3A_ETC	
extb4	EXTB	
spb	SPB	
strc	BITSTRING [2]	
dplxm	BITSTRING [1]	
config	BITSTRING [1]	
nirr	BITSTRING [1]	
est	BITSTRING [1]	
extb5	EXTB	
accid	BITSTRING [2]	
ra	BITSTRING [2]	
sacp	BITSTRING [3]	
extb6	EXTB	
l1id	BITSTRING [2]	
uil1	BITSTRING [4]	
sb	BITSTRING [1]	
extb6a	EXTB	
nsb	BITSTRING [1]	
nb	BITSTRING [1]	
ndb	BITSTRING [1]	
ur	BITSTRING [4]	
extb6b	EXTB	
ir	BITSTRING [2]	
nictx	BITSTRING [1]	
nicrx	BITSTRING [1]	
pi	BITSTRING [3]	
extb6c	EXTB	
ce	BITSTRING [2]	
modemt	BITSTRING [5]	
extb7	EXTB	
l2id	BITSTRING [2]	
uil2	BITSTRING [5]	
Detailed Comments:		

Structured Type Definition		
Type Name:	BLOCKTYPE	
Comments:	Block type, GSM 04.12, 3.3.1	
Element Name	**Type Definition**	**Comments**
spare1	SPB	
lpd	BITSTRING[2]	
lb	BITSTRING[1]	
sequence_number	BITSTRING[4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CAU	
Comments:	Cause (CC information element) GSM 04.08, 10.5.4.11	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
cs	CS	
spb	SPB	
location	BITSTRING [4]	
extb3a	EXTB	
rec	BITSTRING [7]	
extb4	EXTB	
cau_class	BITSTRING [3]	
cau_va	BITSTRING [4]	
cau_di	OCTETSTRING [1..28]	
Detailed Comments:	&COMMON_U06	

Structured Type Definition		
Type Name:	CCCAP	
Comments:	Call Control Capabilities GSM 04.08, 10.5.4.5a	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
spr	BITSTRING [7]	
dtmf	BITSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CCD	
Comments:	Control channel description GSM 04.08, 10.5.2.11	
Element Name	**Type Definition**	**Comments**
spr1	BITSTRING [1]	
att	BITSTRING [1]	
babr	BITSTRING [3]	
ccch_con	BITSTRING [3]	
spr2	BITSTRING [5]	
bpm	BITSTRING [3]	
t3212	OCTETSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CCHD	
Comments:	Cell channel description GSM 04.08, 10.5.2.1	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
rfl	OCTETSTRING [16]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CD	
Comments:	Cell description GSM 04.08, 10.5.2.2	
Element Name	**Type Definition**	**Comments**
bcch_arfcn_h	BITSTRING [2]	
ncc	NCC	
bcc	BCC	
bcch_arfcn_l	BITSTRING [8]	
Detailed Comments:	The info element is two octets long.	

Structured Type Definition		
Type Name:	CDPN	
Comments:	Called party BCD number (CC information element) GSM 04.08, 10.5.4.7	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
tonnpi	TON_NPI	
digits	BCDN	
Detailed Comments:		

Structured Type Definition		
Type Name:	CDPS	
Comments:	Called party subaddress (CC information element) GSM 04.08, 10.5.4.8	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
subad	SUBAD	
Detailed Comments:		

Structured Type Definition		
Type Name:	CE	
Comments:	connection element	
Element Name	**Type Definition**	**Comments**
TorBothTorBothNT	B_2	
NTorBothTorBothNT	B_2	
Detailed Comments:		

Structured Type Definition		
Type Name:	CGPN	
Comments:	Calling party BCD number (CC information element) GSM 04.08, 10.5.4.9	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
tonnpi	TON_NPI	
pisi	PI_SI	
digits	BCDN	
Detailed Comments:		

Structured Type Definition		
Type Name:	CGPS	
Comments:	Calling party subaddress (CC information element) GSM 04.08, 10.5.4.10	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
subad	SUBAD	
Detailed Comments:		

Structured Type Definition		
Type Name:	CHD	
Comments:	Channel description GSM 04.08, 10.5.2.5	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
cht_schn	BITSTRING [5]	
tn	BITSTRING [3]	
tsc	BITSTRING [3]	
hch	BITSTRING [1]	
maio	BITSTRING [6]	
hsn	BITSTRING [6]	
spr	BITSTRING [2]	
arfcn	BITSTRING [10]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CHMOD	
Comments:	Channel mode, channel mode2 GSM 04.08, 10.5.2.6, 10.5.2.7	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
mode	B_8	
Detailed Comments:		

Structured Type Definition		
Type Name:	CHNEED	
Comments:	Channels needed GSM 04.08, 10.5.2.8	
Element Name	**Type Definition**	**Comments**
ch2	CHANNEL	
ch1	CHANNEL	
Detailed Comments:		

Structured Type Definition		
Type Name:	CNN	
Comments:	Connected number (CC information element) GSM 04.08, 10.5.4.13	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
tonnpi	TON_NPI	
pisi	PI_SI	
digits	BCDN	
Detailed Comments:		

Structured Type Definition		
Type Name:	CNS	
Comments:	Connected subaddress (CC information element) GSM 04.08, 10.5.4.14	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
subad	SUBAD	
Detailed Comments:		

Structured Type Definition		
Type Name:	CO	
Comments:	Cell options (BCCH / SACCH) GSM 04.08, 10.5.2.3	
Element Name	**Type Definition**	**Comments**
sprb	BITSTRING [1]	
pwrc	BITSTRING [1]	
dtx	BITSTRING [2]	
rlt	BITSTRING [4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	Component_T	
Comments:	Component for sending (downlink) GSM 04.80, 3.6	
Element Name	**Type Definition**	**Comments**
comp_part1	OCTETSTRING	
invokeId	OCTETSTRING [1]	
comp_part2	OCTETSTRING	
comp_part3	OCTETSTRING	
comp_part4	OCTETSTRING	
comp_part5	OCTETSTRING	
comp_part6	OCTETSTRING	
comp_part7	OCTETSTRING	
comp_part8	OCTETSTRING	
comp_part9	OCTETSTRING	
comp_part10	OCTETSTRING	
comp_part11	OCTETSTRING	
comp_part12	OCTETSTRING	
Detailed Comments:	This component is defined for the use in the downlink (from the test system to the IUT) direction.	

Structured Type Definition		
Type Name:	CPDATA	
Comments:	CP-User data element GSM 04.11, 8.1.4.1	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
rpack	RPACK	
rpdata	RPDATA	
rperr	RPERR	
rpsmma	RPSMMA	
Detailed Comments:	One of the 4 types of rpdu is contained in the CPDATA.	

Structured Type Definition		
Type Name:	CPHMS	
Comments:	Cipher mode setting GSM 04.08, 10.5.2.9	
Element Name	**Type Definition**	**Comments**
iei	IEI_4	
algid	BITSTRING [3]	
sc	BITSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CPHKSN	
Comments:	Ciphering key sequence number GSM 04.08, 10.5.1.2	
Element Name	**Type Definition**	**Comments**
sprb	BITSTRING [1]	
ks	BITSTRING [3]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CPH_RES	
Comments:	Cipher response GSM 04.08, 10.5.2.10	
Element Name	**Type Definition**	**Comments**
iei	IEI_4	
sprb	BITSTRING [3]	
cr	BITSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CSP	
Comments:	Cell selection parameters GSM 04.08, 10.5.2.4	
Element Name	**Type Definition**	**Comments**
crh	BITSTRING [3]	
mtmc	BITSTRING [5]	
acs	BITSTRING [1]	
neci	BITSTRING [1]	
ram	BITSTRING [6]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CST	
Comments:	Call status (CC information element) GSM 04.08, 10.5.4.6	
Element Name	**Type Definition**	**Comments**
cs	CS	
csv	BITSTRING [6]	
Detailed Comments:		

Structured Type Definition		
Type Name:	EFR_BCAP_3A_ETC	
Comments:	Speech Versions preferences as supported by the EFR mobile - see GSM 04.08/10.5.4.5/octet 3a etc/"speech version indication"	
Element Name	**Type Definition**	**Comments**
extb	EXTB	
cod	BITSTRING[1]	
spr2	BITSTRING[2]	
svi	BITSTRING [4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	FIE	
Comments:	Facility information element GSM 04.80, 3.6	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
components_1	Components	
components_t	Component_T	
Detailed Comments:	When sending normally only one component is sent, but when receiving any number of components can be received even though normally we are only interested in one component during the testing.	

Structured Type Definition		
Type Name:	FN	
Comments:	The time in reference to the frame numbering corresponding to the absolute frame number modulo 42432.	
Element Name	**Type Definition**	**Comments**
t1_	T1_	
t3	T3	
t2	T2	
Detailed Comments:	In STRT and RQR	

Structured Type Definition		
Type Name:	FRQCHS	
Comments:	Frequency channel sequence GSM 04.08, 10.5.2.12	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
larfcn	OCTETSTRING [1]	
incs	OCTETSTRING [8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	FRQL	
Comments:	Frequency list, frequency short list GSM 04.08, 10.5.2.5, 10.5.2.6	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
fl	OCTETSTRING	
Detailed Comments:	frequency short list has a fixed length of 10 octets and does not contain the length indicator.	

Structured Type Definition		
Type Name:	FRQPARA	
Comments:	Parameters for Description of basic physical channel in frequency domain.	
Element Name	**Type Definition**	**Comments**
hch	BITSTRING[1]	
maio	BITSTRING [6]	
hsn	BITSTRING [6]	
spr	BITSTRING [2]	
arfcn	BITSTRING [10]	
maclength	LENGTH	
mac_8n	BITSTRING [8]	
mac_7n	BITSTRING [8]	
mac_6n	BITSTRING [8]	
mac_5n	BITSTRING [8]	
mac_4n	BITSTRING [8]	
mac_3n	BITSTRING [8]	
mac_2n	BITSTRING [8]	
mac_1n	BITSTRING [8]	
flst	FRQL	
Detailed Comments:		

Structured Type Definition		
Type Name:	HOACC_PARA	
Comments:	Handover Access parameters	
Element Name	**Type Definition**	**Comments**
num_bursts	OCTETSTRING[1]	
min_mspwr	OCTETSTRING[1]	
max_mspwr	OCTETSTRING[1]	
min_msta	BITSTRING[8]	
max_msta	BITSTRING[8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	HLCMP	
Comments:	High layer compatibility (CC information element) GSM 04.08, 10.5.4.16, ITU Q.931	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
cs	BITSTRING [2]	
in	BITSTRING [3]	
pmpp	BITSTRING [2]	
extb4	EXTB	
hlci	BITSTRING [7]	
extb4a	EXTB	
ehlci	BITSTRING [7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	IARESTOCT	
Comments:		
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iaroct1	OCTETSTRING[1..11]	
iaroct2	IARESTOCT2	
Detailed Comments:		

Structured Type Definition		
Type Name:	IARESTOCT2	
Comments:		
Element Name	**Type Definition**	**Comments**
p	BITSTRING[2]	
frqparalen	BITSTRING[6]	
spbt1	BITSTRING[2]	
maio	BITSTRING[6]	
ma	OCTETSTRING[0..8]	
spbt2	OCTETSTRING[0..8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	KPF (keypad facility)	
Comments:	keypad facility GSM 04.08 clause 10.5.4.17	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
extb	EXTB	
kpf_info	IA5String [1]	
Detailed Comments:	Is extb needed?	

Structured Type Definition		
Type Name:	L1HD	
Comments:	Layer 1 Header	
Element Name	**Type Definition**	**Comments**
spr1	BITSTRING[3]	
mspwrlvl	BITSTRING[5]	
spr2	BITSTRING[1]	
ta	BITSTRING[7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	LAI	
Comments:	Location area identification GSM 04.08, 10.5.1.3	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
mcc	OCTETSTRING [2]	
mnc	OCTETSTRING [1]	
lac	OCTETSTRING [2]	
Detailed Comments:		

Structured Type Definition		
Type Name:	LLCMP	
Comments:	Low layer compatibility (CC information element) GSM 04.08, 10.5.4.18 and ETS DE/SPS-5034-1 (update of ETS 300 102-1)	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
cs	BITSTRING [2]	
itc	BITSTRING [5]	
extb3a	EXTB	
negind	BITSTRING[1]	
spb3a	SP6B	
extb4	EXTB	
tm	BITSTRING [2]	
itr	BITSTRING [5]	
extb4_1	EXTB	
rateml	BITSTRING [7]	
extb5	EXTB	
l1id	BITSTRING [2]	
uil1	BITSTRING [5]	
extb5a	EXTB	
sb	BITSTRING [1]	
neg	BITSTRING [1]	
ur	BITSTRING [5]	
extb5b1	EXTB	
ir	BITSTRING [2]	
nictx	BITSTRING [1]	
nicrx	BITSTRING [1]	
fctx	BITSTRING [1]	
fcrx	BITSTRING [1]	
spb5b1	SPB	
extb5b2	EXTB	
hdrb	BITSTRING[1]	
mfs	BITSTRING[1]	
mode	BITSTRING[1]	
llineg	BITSTRING[1]	
ass	BITSTRING[1]	
ibob	BITSTRING[1]	
spb5b2	SPB	
extb5c	EXTB	
nsb	BITSTRING [2]	
ndb	BITSTRING [2]	
pi	BITSTRING [3]	
extb5d	EXTB	
dplxm	BITSTRING [1]	
modemt	BITSTRING [6]	
extb6	EXTB	
l2id	BITSTRING [2]	
uil2	BITSTRING [5]	
extb6a1	EXTB	
modl2	BITSTRING[2]	
spb6a1	SP3B	
q933	BITSTRING [2]	
extb6a2	EXTB	
usl2pi	BITSTRING[7]	
extb6b	EXTB	
ws	BITSTRING[7]	
extb7	EXTB	
l3id	BITSTRING [2]	
uil3	BITSTRING [5]	
extb7a1	EXTB	
modl3	BITSTRING[2]	
spb7a1	SP5B	

extb7a2	EXTB	
usl3pi	BITSTRING[7]	
extb7b	EXTB	
spb7b	SP3B	
dps	BITSTRING [4]	
extb7c	EXTB	
pws	BITSTRING[7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	LUT	
Comments:	Location updating type GSM 04.08, 10.5.3.5	
Element Name	**Type Definition**	**Comments**
foreq	BITSTRING [1]	
sprb	BITSTRING [1]	
lut	BITSTRING [2]	
Detailed Comments:		

Structured Type Definition		
Type Name:	MA	
Comments:	Mobile allocation GSM 04.08, 10.5.2.21	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
mac_8n	BITSTRING [8]	
mac_7n	BITSTRING [8]	
mac_6n	BITSTRING [8]	
mac_5n	BITSTRING [8]	
mac_4n	BITSTRING [8]	
mac_3n	BITSTRING [8]	
mac_2n	BITSTRING [8]	
mac_1n	BITSTRING [8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	MI	
Comments:	Mobile identity GSM 04.08, 10.5.1.4	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
idigit_1	BITSTRING [4]	
oei	BITSTRING [1]	
toi	BITSTRING [3]	
idigits_other	OCTETSTRING [0..7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	MODEMTYPE	
Comments:		
Element Name	**Type Definition**	**Comments**
V22bisOrV26ter	B_5	
Detailed Comments:		

Structured Type Definition		
Type Name:	MSCLM1	
Comments:	mobile station classmark 1 GSM 04.08, 10.5.1.5	
Element Name	**Type Definition**	**Comments**
spr1	BITSTRING [1]	
rl	BITSTRING [2]	
spr2	BITSTRING [1]	
a5_1	BITSTRING [1]	
rfpc	BITSTRING [3]	
Detailed Comments:	2 octets long	

Structured Type Definition		
Type Name:	MSCLM2	
Comments:	mobile station classmark 2 GSM 04.08, 10.5.1.6	
Element Name	**Type Definition**	**Comments**
iel	LENGTH	
spr1	BITSTRING [1]	
rl	BITSTRING [2]	
spr2	BITSTRING [1]	
a5_1	BITSTRING [1]	
rfpc	BITSTRING [3]	
spr3	BITSTRING [1]	
psc	BITSTRING [1]	
sssi	BITSTRING [2]	
smc	BITSTRING [1]	
spr4	BITSTRING [2]	
fc	BITSTRING [1]	
cm3	BITSTRING [1]	
spr5	BITSTRING [5]	
a5_3	BITSTRING [1]	
a5_2	BITSTRING [1]	
Detailed Comments:	4 octets long	

Structured Type Definition		
Type Name:	MSCLM3	
Comments:	Mobile station classmark 3 GSM 04.08, 10.5.1.7	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
spr1	BITSTRING [4]	
a5_7	BITSTRING [1]	
a5_6	BITSTRING [1]	
a5_5	BITSTRING [1]	
a5_4	BITSTRING [1]	
spr2	OCTETSTRING [11]	
Detailed Comments:	The info element has 14 octets	

Structured Type Definition		
Type Name:	MSRR	
Comments:	measurement results GSM 04.08, 10.5.2.20	
Element Name	**Type Definition**	**Comments**
ba_used	BITSTRING [1]	
dtx_used	BITSTRING [1]	
rxlev_fsc	RXLEV	
spr1	BITSTRING [1]	
meas_valid	BITSTRING [1]	
rxlev_ssc	RXLEV	
spr2	BITSTRING [1]	
rxqual_fsc	BITSTRING [3]	
rxqual_ssc	BITSTRING [3]	
no_nc	BITSTRING [3]	
rxlev_nc1	RXLEV	
bcchfrq_nc1	BCCHFRQ	
bsic_nc1	BSIC	
rxlev_nc2	RXLEV	
bcchfrq_nc2	BCCHFRQ	
bsic_nc2	BSIC	
rxlev_nc3	RXLEV	
bcchfrq_nc3	BCCHFRQ	
bsic_nc3	BSIC	
rxlev_nc4	RXLEV	
bcchfrq_nc4	BCCHFRQ	
bsic_nc4	BSIC	
rxlev_nc5	RXLEV	
bcchfrq_nc5	BCCHFRQ	
bsic_nc5	BSIC	
rxlev_nc6	RXLEV	
bcchfrq_nc6	BCCHFRQ	
bsic_nc6	BSIC	
Detailed Comments:	The info element has a fixed length of 16 octets.	

Structured Type Definition		
Type Name:	MTDIF	
Comments:	Mobile time difference GSM 04.08, 10.5.1.21a	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
value	OCTETSTRING [3]	
Detailed Comments:		

Structured Type Definition		
Type Name:	NCD	
Comments:	Neighbour cells description GSM 04.08, 10.5.2.22	
Element Name	**Type Definition**	**Comments**
rfl2	BITSTRING [2]	
extind	BITSTRING [1]	
baind	BITSTRING [1]	
rfl4	BITSTRING [4]	
rfl	OCTETSTRING [15]	
Detailed Comments:		

Structured Type Definition		
Type Name:	NCD2	
Comments:	Neighbour cells description2 GSM 04.08, 10.5.2.22b	
Element Name	**Type Definition**	**Comments**
b128	BITSTRING [1]	
mbrpt	BITSTRING [2]	
baind	BITSTRING [1]	
b122_124	BITSTRING [3]	
rfl1	BITSTRING [1]	
rfl	OCTETSTRING [15]	
Detailed Comments:		

Structured Type Definition		
Type Name:	PCMD	
Comments:	Power command and access type GSM 04.08, 10.5.2.28, 10.5.2.28a	
Element Name	**Type Definition**	**Comments**
sprb	BITSTRING [3]	
pl	BITSTRING [5]	
Detailed Comments:		

Structured Type Definition		
Type Name:	PI	
Comments:	Progress indicator GSM 04.08, 10.5.4.21	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
cs	BITSTRING [2]	
spb	SPB	
loc	BITSTRING [4]	
extb4	EXTB	
prd	BITSTRING [7]	
Detailed Comments:	total 4 octets	

Structured Type Definition		
Type Name:	PI_SI	
Comments:	Presentation indicator & screening indicator GSM 04.08, 10.5.4.9, 10.5.4.13	
Element Name	**Type Definition**	**Comments**
extb	EXTB	
pi	BITSTRING [2]	
sp3b	SP3B	
si	BITSTRING [2]	
Detailed Comments:		

Structured Type Definition		
Type Name:	PM	
Comments:	Page mode GSM 04.08, 10.5.2.26	
Element Name	**Type Definition**	**Comments**
sprb	B_2	
pgm	B_2	
Detailed Comments:		

Structured Type Definition		
Type Name:	RACHCP	
Comments:	Rach control parameters GSM 04.08, 10.5.2.29	
Element Name	**Type Definition**	**Comments**
maxrtx	BITSTRING [2]	
txint	BITSTRING [4]	
cba	BITSTRING [1]	
re	BITSTRING [1]	
acc_2	BITSTRING [5]	
ec	BITSTRING [1]	
acc_1	BITSTRING [10]	
Detailed Comments:	The info element has a fixed length of 3 octets.	

Structured Type Definition		
Type Name:	RPACK	
Comments:	SMS RP ACKNOWLEDGEMENT ms <-> n GSM 04.11, 7.3.3	
Element Name	**Type Definition**	**Comments**
sprb	SP5B	
rpmti	MTI	
rpmr	MR	
Detailed Comments:		

Structured Type Definition		
Type Name:	RPCAU	
Comments:	RP user data element GSM 04.11, 8.2.5.4	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb2	EXTB	
rpcau_class	BITSTRING [3]	
rpcau_va	BITSTRING [4]	
rpcau_di	OCTETSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	RPDATA	
Comments:	SMS RP DATA ms <-> n GSM 04.11, 7.3.1	
Element Name	**Type Definition**	**Comments**
sprb	SP5B	
rpmti	MTI	
rpmr	MR	
rpOaddr	CDPN	
rpDaddr	CDPN	
rpusrdat	RPUSRDAT	
Detailed Comments:		

Structured Type Definition		
Type Name:	RPERR	
Comments:	SMS RP ERROR ms <-> n GSM 04.11, 7.3.4	
Element Name	**Type Definition**	**Comments**
sprb	SP5B	
rpmti	MTI	
rpmr	MR	
rpcau	RPCAU	
rpusrdat	RPUSRDAT	
Detailed Comments:		

Structured Type Definition		
Type Name:	RPSMMA	
Comments:	SMS RP SMMA ms -> n GSM 04.11, 7.3.2	
Element Name	**Type Definition**	**Comments**
sprb	SP5B	
rpmti	MTI	
rpmr	MR	
Detailed Comments:	SM memory available	

Structured Type Definition		
Type Name:	RPUSRDAT	
Comments:	RP user data element GSM 04.11, 8.2.5.3, GSM 03.40, 9.2.2	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
tpdeliver	SMDLVR	
tpsubmit	SMSBMT	
tpstatus_rpt	SMST_RPT	
tpcommand	SMCMD	
tpdlvr_sbmt_rpt	SMDLVR_RPT	
Detailed Comments:	One of the six tpdu is contained in the RPUSRDATA. Since the structures of the messages SMS-Deliver-Report and SMS-Submit-Report are identical, they have been combined to tpdlvr_sbmt_rpt and therefore only five tp message types exist in this type definition.	

Structured Type Definition		
Type Name:	RQR	
Comments:	Request reference GSM 04.08, 10.5.2.30	
Element Name	**Type Definition**	**Comments**
ra	BITSTRING [8]	
fn	FN	
Detailed Comments:		

Structured Type Definition		
Type Name:	SI3RO	
Comments:	SI3 Rest Octets GSM 04.08, 10.5.2.34	
Element Name	**Type Definition**	**Comments**
pi	BITSTRING [1]	
cbq	BITSTRING [1]	
cro	BITSTRING [6]	
to	BITSTRING [3]	
pt	BITSTRING [5]	
poi	BITSTRING [1]	
po	BITSTRING [2]	
ti	BITSTRING [1]	
ecsc	BITSTRING [1]	
spr1	BITSTRING [3]	
spr2	BITSTRING [8]	
Detailed Comments:	The info element has a fixed length of 4 octets.	

Structured Type Definition		
Type Name:	SI4RO	
Comments:	SI4 Rest Octets GSM 04.08, 10.5.2.35	
Element Name	**Type Definition**	**Comments**
pi	BITSTRING [1]	
cbq	BITSTRING [1]	
cro	BITSTRING [6]	
to	BITSTRING [3]	
pt	BITSTRING [5]	
poi	BITSTRING [1]	
po	BITSTRING [2]	
spr1	BITSTRING [5]	
spr2	BITSTRING [8]	
spr3	BITSTRING [8]	
spr4	BITSTRING [8]	
spr5	BITSTRING [8]	
spr6	BITSTRING [8]	
spr7	BITSTRING [8]	
spr8	BITSTRING [8]	
Detailed Comments:	The info element has a fixed length of 10 octets.	

Structured Type Definition		
Type Name:	SI7RO	
Comments:	SI7 Rest Octets GSM 04.08, 10.5.2.36	
Element Name	**Type Definition**	**Comments**
pi	BITSTRING [1]	
cbq	BITSTRING [1]	
cro	BITSTRING [6]	
to	BITSTRING [3]	
pt	BITSTRING [5]	
poi	BITSTRING [1]	
po	BITSTRING [2]	
spr1	BITSTRING [5]	
spr2	BITSTRING [8]	
spr3	BITSTRING [8]	
spr4	BITSTRING [8]	
spr5	BITSTRING [8]	
spr6	BITSTRING [8]	
spr7	BITSTRING [8]	
spr8	BITSTRING [8]	
spr9	BITSTRING [8]	
spr10	BITSTRING [8]	
spr11	BITSTRING [8]	
spr12	BITSTRING [8]	
spr13	BITSTRING [8]	
spr14	BITSTRING [8]	
spr15	BITSTRING [8]	
spr16	BITSTRING [8]	
spr17	BITSTRING [8]	
spr18	BITSTRING [8]	
Detailed Comments:	The info element has a fixed length of 20 octets.	

Structured Type Definition		
Type Name:	SIGNAL	
Comments:	Signal (CC information element) GSM 04.08, 10.5.4.23	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
sigv	BITSTRING [8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	SERIAL_NUMBER	
Comments:	Serial number for SMSCB, GSM 3.41, 9.3.2	
Element Name	**Type Definition**	**Comments**
gs	BITSTRING[2]	
message_code	BITSTRING[10]	
update_number	BITSTRING[4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMCMD	
Comments:	SMS COMMAND, ms - > n. GSM 03.40, 9.2.2.4	
Element Name	**Type Definition**	**Comments**
sprb1	SP2B	
srr	BITSTRING[1]	
sprb2	SP3B	
mti	BITSTRING [2]	
mr	MR	
pid	TPPID	
ct	OCTETSTRING [1]	
mn	OCTETSTRING [1]	
da	TPA	
cdl	LENGTH	
cd	TPCD	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMDLVR	
Comments:	SMS DELIVER, n - > ms. GSM 03.40, 9.2.2.1	
Element Name	**Type Definition**	**Comments**
rp	BITSTRING [1]	
udhi	BITSTRING [1]	
sri	BITSTRING [1]	
sprb2	SP2B	
mms	BITSTRING [1]	
mti	BITSTRING [2]	
oa	TPA	
pid	TPPID	
dcs	TPDCS	
scts	TPSCTS	
udl	LENGTH	
ud	TPUD	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMDLVR_RPT	
Comments:	SMS DELIVER or SUBMIT REPORT contained in the RP ERROR PDU, n <-> ms. GSM 03.40, 9.2.2.1a, 9.2.2.2a.	
Element Name	**Type Definition**	**Comments**
sprb1	SP6B	
mti	BITSTRING [2]	
fcs	FCS	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMSBMT	
Comments:	SMS SUBMIT, ms -> n. GSM 03.40, 9.2.2.2	
Element Name	**Type Definition**	**Comments**
rp	BITSTRING [1]	
udhi	BITSTRING [1]	
srr	BITSTRING [1]	
vpf	BITSTRING [2]	
rd	BITSTRING [1]	
mti	BITSTRING [2]	
mr	MR	
da	TPA	
pid	TPPID	
dcs	TPDCS	
vp1	OCTETSTRING [1]	
vp7	TPSCTS	
udl	LENGTH	
ud	TPUD	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMSCBpack	
Comments:	SMS cell broadcasting packing data n -> ms. GSM 03.38, 6.1.2.2.1	
Element Name	**Type Definition**	**Comments**
block1	OCTETSTRING [16]	
block2	OCTETSTRING [22]	
block3	OCTETSTRING [22]	
block4	OCTETSTRING [22]	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMST_RPT	
Comments:	SMS STATUS REPORT, n - > ms. GSM 03.40, 9.2.2.3	
Element Name	**Type Definition**	**Comments**
sprb1	SP5B	
mms	BITSTRING [1]	
mti	BITSTRING [2]	
mr	MR	
ra	TPA	
scts	TPSCTS	
dt	TPSCTS	
st	TPST	
Detailed Comments:		

Structured Type Definition		
Type Name:	SSVI	
Comments:	SS version indicator GSM 04.08, 10.5.4.24	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
ssv	OCTETSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	STRT	
Comments:	Starting time GSM 04.08, 10.5.2.38	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
fn	FN	
Detailed Comments:		

Structured Type Definition		
Type Name:	SUBAD	
Comments:	Subaddress GSM 04.08, 10.5.4.8, 10.5.4.10, 10.5.4.14	
Element Name	**Type Definition**	**Comments**
extb	EXTB	
tos	BITSTRING [3]	
oei	BITSTRING [1]	
sp3b	SP3B	
si	OCTETSTRING [2..23]	
Detailed Comments:		

Structured Type Definition		
Type Name:	SYNCHI	
Comments:	Synchronization indication GSM 04.08, 10.5.2.39	
Element Name	**Type Definition**	**Comments**
iei	IEI_4	
nci	B_1	
rot	B_1	
si	B_2	
Detailed Comments:		

Structured Type Definition		
Type Name:	TA	
Comments:	Timing advance GSM 04.08, 10.5.2.40	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
sprb	BITSTRING [2]	
value	BITSTRING [6]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TDIF	
Comments:	Time difference GSM 04.08, 10.5.1.41	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
value	OCTETSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TI	
Comments:	Transaction identifier GSM 04.08, 10.1	
Element Name	**Type Definition**	**Comments**
ti_f	BITSTRING [1]	
ti_v	TI_V	
Detailed Comments:		

Structured Type Definition		
Type Name:	TMSI	
Comments:	Temporary GSM 04.08, 10.5.2.42	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
tmsi_val	OCTETSTRING [4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TON_NPI	
Comments:	Type of number and numbering plan GSM 04.08, 10.5.4.7, 10.5.4.9, 10.5.4.13, GSM 03.40, 9.1.2.5	
Element Name	**Type Definition**	**Comments**
extb	EXTB	
ton	BITSTRING [3]	
npi	BITSTRING [4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TPA	
Comments:	TP address GSM 03.40, 9.1.2.5	
Element Name	**Type Definition**	**Comments**
iel	LENGTH	
tonnpi	TON_NPI	
digits	BCDN	
Detailed Comments:		

Structured Type Definition		
Type Name:	TPDCS	
Comments:	SMS data coding scheme GSM 03.38, 4, 5	
Element Name	**Type Definition**	**Comments**
cg	BITSTRING [4]	
code	BITSTRING [4]	
Detailed Comments:	Identifying the coding scheme within the TP user data.	

Structured Type Definition		
Type Name:	TPPID	
Comments:	TP protocol identifier GSM 03.40, 9.2.3.9	
Element Name	**Type Definition**	**Comments**
type	BITSTRING [2]	
value	BITSTRING [6]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TPST	
Comments:	TP status GSM 03.40, 9.2.3.15	
Element Name	**Type Definition**	**Comments**
sprb1	SPB	
value	BITSTRING [7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	UNKWN	
Comments:	unknown IE	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
contents	OCTETSTRING	
Detailed Comments:		

Structured Type Definition		
Type Name:	UU	
Comments:	User-user GSM 04.08, 10.5.4.25	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
uupd	BITSTRING [8]	
uui	OCTETSTRING [1..128]	
Detailed Comments:	In SETUP, ALERTING, CONNECT, DISCONNECT, RELEASE and RELEASE COMPLETE messages the uui length is of 0 - 32 bytes. In USER INFORMATION messages the uui length is of 1 - 128.	

ASN1 type definitions

ASN.1 Type Definition	
Type Name:	Component
Comments:	ASN1_Encoding: BER
Type Definition	

```
CHOICE {
   registerSSComponents                          RegisterSS_Components,
   eraseSSComponents                              EraseSS_Components,
   activateSSComponents                          ActivateSS_Components,
   deactivateSSComponents                      DeactivateSS_Components,
   interrogateSSComponents                     InterrogateSS_Components,
   notifySSComponents                             NotifySS_Components,
   registerPasswordComponents                RegisterPassword_Components,
   getPasswordComponents                       GetPassword_Components,
   processUnstructuredSSDataComponents     ProcessUnstructuredSSData_Components,
   forwardCheckSSIndicationComponents        ForwardCheckSSIndication_Components,
   processUnstructuredSSRequestComponents ProcessUnstructuredSSRequest_Components,
   unstructuredSSRequestComponents           UnstructuredSSRequest_Components,
   unstructuredSSNotifyComponents              UnstructuredSSNotify_Components,
   forwardCUGInfoComponents                       ForwardCUGInfo_Components,
   splitMPTYComponents                                 SplitMPTY_Components,
   retrieveMPTYComponents                           RetrieveMPTY_Components,
   holdMPTYComponents                                 HoldMPTY_Components,
   buildMPTYComponents                                BuildMPTY_Components,
   forwardChargeAdviceComponents             ForwardChargeAdvice_Components,
   generalComponents                                     General_Components
}
```

Detailed Comments:	Plural components as each type represents invoke, return result, return error etc.

ASN.1 Type Definition	
Type Name:	Components
Comments:	GSM 04.80, 3.6
Type Definition	

```
SET OF Component
```

Detailed Comments:	ASN.1 transfer encoding rules: BER is not wholy used for the type Components. The contents of Components, without the octets encoding the tag and the length of SET OF, is carried as the Components value.

ASN.1 Type Definition

Type Name: ActivateSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       activateSS_InvokeComp          [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (12),
                                ss_ForBS                    SS_ForBS_Code },

       activateSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                        localValue                          INTEGER (12),
                                        ss_Info              SS_Info        }      } ,

       activateSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| dataMissing

| unexpectedDataValue

| negativePW_Check

| numberOfPW_AttemptsViolation) },
              callBarredErr                        [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_ErrorStatusErr             [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_SubscriptionViolationeErr [3] IMPLICIT SEQUENCE { invokeID       InvokeIDType,

errorCode     ErrorLocalValues (ss_SubscriptionViolation),

parameter      SS_SubscriptionOption },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                 [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       activateSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m (15 - 30 s)

ASN.1 Type Definition

Type Name: BuildMPTY_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       buildMPTY_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (124)  },

       buildMPTY_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                         localValue                          INTEGER (124) }
} ,

       buildMPTY_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSS_Operation

| ss_NotAvailable

| maxNumberOfMPTY_ParticipantsExceeded

| resourcesNotAvailable ) },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       buildMPTY_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_buildMPTY (5 - 30 s)

ASN.1 Type Definition

Type Name: DeactivateSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       deactivateSS_InvokeComp          [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (13),
                                ss_ForBS                    SS_ForBS_Code },

       deactivateSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                   InvokeIDType,
                                 result                        SEQUENCE {
                                        localValue                        INTEGER (13),
                                        ss_Info              SS_Info       }      } ,

       deactivateSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| dataMissing

| unexpectedDataValue

| negativePW_Check

| numberOfPW_AttemptsViolation) },
              callBarredErr                       [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_ErrorStatusErr             [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_SubscriptionViolationeErr [3] IMPLICIT SEQUENCE { invokeID       InvokeIDType,

errorCode     ErrorLocalValues (ss_SubscriptionViolation),

parameter      SS_SubscriptionOption },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       deactivateSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: EraseSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       eraseSS_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (11),
                                ss_ForBS                    SS_ForBS_Code },

       eraseSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                        localValue                        INTEGER (11),
                                        ss_Info              SS_Info    }             } ,

       eraseSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues (  bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| dataMissing

| unexpectedDataValue ) },
              callBarredErr                       [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              systemFailureErr                 [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },


       eraseSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: ForwardChargeAdvice_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       forwardChargeAdvice_InvokeComp           [1] IMPLICIT SEQUENCE {
                           invokeID                                        InvokeIDType,
                           localValue                                       INTEGER (125),
                           forwardChargeAdviceArg              SEQUENCE {
                                                ss_Code                       [0] SS_Code,
                                                chargingInformation     [1]  SEQUENCE{
                                                      e1       [1] INTEGER
(0..max10TimesUnitsPerTime)                        OPTIONAL,
                                                      e2       [2]  INTEGER
(0..max10TimesTimeInterval)                         OPTIONAL,
                                                      e3       [3]  INTEGER
(0..max100TimesScalingFactor)                     OPTIONAL,
                                                      e4       [4] INTEGER
(0..max10TimesIncrement)                             OPTIONAL,
                                                      e5       [5]  INTEGER
(0..max10TimesIncrementPerDataInterval)    OPTIONAL,
                                                      e6       [6] INTEGER
(0..maxNumberOfSegmentsPerDataInterval) OPTIONAL,
                                                      e7       [7]  INTEGER
(0..max10TimesInitialTime)                           OPTIONAL } } },

       forwardChargeAdvice_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                           invokeID                    InvokeIDType,
                           result                         SEQUENCE {
                           localValue                   INTEGER (125)  }    } ,

       forwardChargeAdvice_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_aoc = 1 - 40 s

ASN.1 Type Definition

Type Name: ForwardCheckSSIndication_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       forwardCheckSSIndication_InvokeComp           [1] IMPLICIT SEQUENCE {
                              invokeID                      InvokeIDType,
                              localValue                     INTEGER (38)    },

      forwardCheckSSIndication_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments:

ASN.1 Type Definition

Type Name: ForwardCUGInfo_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       forwardCUGInfo_InvokeComp          [1] IMPLICIT SEQUENCE {
                              invokeID                     InvokeIDType,
                              localValue                    INTEGER (120),
                              forwardCUGInfo_Arg             SEQUENCE{
                                       cug_Index              [0]  IMPLICIT CUG_Index
OPTIONAL,
                                       suppressPrefCUG   [1]  IMPLICIT NULL           OPTIONAL,
                                       suppressOA             [2]  IMPLICIT NULL
OPTIONAL } },

        forwardCUGInfo_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	General_Components
Comments:	Non specified components must match this type definition.

Type Definition

```
CHOICE {
  general_InvokeComp          [1] IMPLICIT General_InvokeComponent,
  general_ReturnResultComp [2] IMPLICIT General_ReturnResultComponent,
  general_ReturnErrorComp  [3] IMPLICIT General_ReturnErrorComponent,
  general_RejectComp           [4] IMPLICIT RejectComponent }


-- This is the General InvokeComponent --
General_InvokeComponent ::= SEQUENCE {
  invokeID                     InvokeIDType,
  linked_ID       [0] IMPLICIT InvokeIDType OPTIONAL,
  operation_value              Operation,
  argument                     ANY     OPTIONAL }

-- This is the General ReturnResultComponent --
General_ReturnResultComponent ::= SEQUENCE {
  invokeID       InvokeIDType,
  valueAndResult SEQUENCE {
                    operation_value Operation,
                    result          ANY     } OPTIONAL }

-- This is the General ReturnErrorComponent --
General_ReturnErrorComponent ::= SEQUENCE {
  invokeID InvokeIDType,
  error    ANY }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	GetPassword_Components
Comments:	GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       getPassword_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                linkedID                       [0] IMPLICIT InvokeIDType,
                                localValue                     INTEGER (18),
                                guidanceInfo                ENUMERATED {
                                                                    enterPW  (0),
                                                                    enterNewPW  (1),
                                                                    enterNewPW_Again  (2) }
},

       getPassword_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                        localValue                        INTEGER (18),
                                        currentPassword              Password }            } ,


       getPassword_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: HoldMPTY_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       holdMPTY_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (123)  },

       holdMPTY_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                         localValue                          INTEGER (123) }
} ,

       holdMPTY_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSS_Operation

| facilityNotSupported ) },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       holdMPTY_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_holdMPTY (5 - 30 s)

ASN.1 Type Definition

Type Name: InterrogateSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       interrogateSS_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (14),
                                ss_ForBS                    SS_ForBS_Code },

       interrogateSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                   InvokeIDType,
                                 result                        SEQUENCE {
                                        localValue                        INTEGER (14),
                                        interrogateSS_Res             CHOICE {
                                             ss_Status                     [0] IMPLICIT
SS_Status,
                                             basicServiceGroupList   [2] IMPLICIT
BasicServiceGroupList,
                                             forwardingFeatureList   [3] IMPLICIT
ForwardingFeatureList,
                                             cli_RestrictionInfo         [4] IMPLICIT SEQUENCE {
                                                                      ss_Status
SS_Status,
                                                                      cliRestrictionOption
CliRestrictionOption    OPTIONAL } } } } ,

       interrogateSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| ss_NotAvailable

| dataMissing

| unexpectedDataValue) },
              callBarredErr                       [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              systemFailureErr                 [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },


       interrogateSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: NotifySS_Components
Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       notifySS_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (16),
                                notifySS_Arg              SEQUENCE{
                                        ss_Code                        [1] IMPLICIT SS_Code
OPTIONAL,
                                        ss_Status                      [4] IMPLICIT SS_Status
OPTIONAL,
                                        ss_Notification              [5] IMPLICIT OCTET STRING
(SIZE (1)) OPTIONAL,
                                        callIsWaiting_Indicator   [14] IMPLICIT NULL
OPTIONAL,
                                        callOnHold_Indicator     [15] IMPLICIT ENUMERATED {

callRetrieved (0),

callOnHold (1) }         OPTIONAL,
                                        mpty_Indicator               [16] IMPLICIT NULL
OPTIONAL,
                                        cug_Index                      [17]  IMPLICIT CUG_Index
OPTIONAL,
                                        clirSuppressionRejected [18]  IMPLICIT NULL
OPTIONAL }    },

      notifySS_RejectComp             [4] IMPLICIT RejectComponent        }
```

Detailed Comments:

ASN.1 Type Definition

Type Name: ProcessUnstructuredSSData_Components
Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       processUnstructuredSSData_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (19),
                                ss_UserData               SS_UserData },

       processUnstructuredSSData_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                        SEQUENCE {
                                        localValue                          INTEGER (19),
                                        ss_UserData                   SS_UserData   }  } ,

       processUnstructuredSSData_ReturnErrorComp   CHOICE {
              errorCodes                         [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues (unexpectedDataValue) },
              systemFailureErr                 [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       processUnstructuredSSData_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: ProcessUnstructuredSSRequest_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       processUnstructuredSSRequest_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                     InvokeIDType,
                                localValue                    INTEGER (59),
                                ussd_Arg                   USSD_Arg },

       processUnstructuredSSRequest_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                         localValue                          INTEGER (59),
                                         ussd_Res                         USSD_Res}            }
,

       processUnstructuredSSRequest_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues (dataMissing

| unexpectedDataValue

| unknownAlphabet) },
              callBarredErr                        [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       processUnstructuredSSRequest_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: RegisterPassword_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       registerPassword_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (17),
                                ss_Code                      SS_Code      },

       registerPassword_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                        localValue                          INTEGER (17),
                                        newPassword                 Password    }         } ,

       registerPassword_ReturnErrorComp   CHOICE {
              errorCodes                              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( dataMissing

| unexpectedDataValue

| negativePW_Check

| numberOfPW_AttemptsViolation) },
              callBarredErr                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode      ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_SubscriptionViolationeErr  [3] IMPLICIT SEQUENCE { invokeID    InvokeIDType,

errorCode      ErrorLocalValues (ss_SubscriptionViolation),

parameter      SS_SubscriptionOption },
              systemFailureErr                    [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter     NetworkResource},
              pw_RegistrationFailureErr       [3] IMPLICIT SEQUENCE { invokeID    InvokeIDType,

errorCode     ErrorLocalValues ( pw_RegistrationFailure ),

parameter     PW_RegistrationFailureCause }   },

       registerPassword_RejectComp      [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: RegisterSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       registerSS_InvokeComp          [1] IMPLICIT SEQUENCE {
                                invokeID                     InvokeIDType,
                                localValue                    INTEGER (10),
                                registerSS_Arg           SEQUENCE{
                                        ss_Code                         SS_Code,
                                        basicService                    BasicServiceCode
OPTIONAL,
                                        forwardedToNumber         [4] IMPLICIT AddressString
OPTIONAL,
                                        forwardedToSubaddress   [6] IMPLICIT
ISDN_SubaddressString       OPTIONAL,
                                        noReplyConditionTime      [5] IMPLICIT
NoReplyConditionTime        OPTIONAL     }   },

       registerSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                     InvokeIDType,
                                 result                          SEQUENCE {
                                         localValue                        INTEGER (10),
                                         ss_Info              SS_Info          }         } ,

       registerSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| dataMissing

| unexpectedDataValue ) },
              callBarredErr                       [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },


       registerSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: RejectComponent

Comments: ITU-T Q.767

Type Definition

```
SEQUENCE {
  invokedID CHOICE {
                      derivable            InvokeIDType,
                      notDerivable      NULL },
  problem    CHOICE {
                      generalProblem         [0] IMPLICIT GeneralProblem,
                      invokeProblem          [1] IMPLICIT InvokeProblem,
                      returnResultProblem [2] IMPLICIT ReturnResultProblem,
                      returnErrorProblem  [3] IMPLICIT ReturnErrorProblem } }
```

Detailed Comments: Reject Component is not specific to any particular operation. The invokeID may be used to identify a specific operation.

ASN.1 Type Definition

Type Name: RetrieveMPTY_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       retrieveMPTY_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (122)  },

       retrieveMPTY_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                     InvokeIDType,
                                 result                          SEQUENCE {
                                         localValue                          INTEGER (122) }
} ,

       retrieveMPTY_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSS_Operation

| facilityNotSupported ) },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       retrieveMPTY_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_retrieveMPTY (5 - 30 s)

ASN.1 Type Definition

Type Name: SplitMPTY_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       splitMPTY_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (121)  },

       splitMPTY_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                         localValue                          INTEGER (121) }
} ,

       splitMPTY_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSS_Operation

| facilityNotSupported ) },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       splitMPTY_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_splitMPTY (5 - 30 s)

ASN.1 Type Definition

Type Name: UnstructuredSSNotify_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       unstructuredSSNotify_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (61),
                                ussd_Arg                    USSD_Arg                  },

       unstructuredSSNotify_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                   InvokeIDType,
                                 result                        SEQUENCE {
                                         localValue                          INTEGER (61) }  } ,

       unstructuredSSNotify_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSubscriber

| illegalEquipment

| absentSubscriber

| dataMissing

| unexpectedDataValue

| unknownAlphabet

| ussd_Busy) },
              systemFailureErr                 [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       unstructuredSSNotify_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition	
Type Name:	UnstructuredSSRequest_Components
Comments:	GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       unstructuredSSRequest_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (60),
                                ussd_Arg                     USSD_Arg                  },

       unstructuredSSRequest_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                        SEQUENCE {
                                         localValue                          INTEGER (60),
                                        ussd_Res                          USSD_Res       }  } ,

       unstructuredSSRequest_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSubscriber

| illegalEquipment

| absentSubscriber

| dataMissing

| unexpectedDataValue

| unknownAlphabet

| ussd_Busy) },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       unstructuredSSRequest_RejectComp             [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition	
Type Name:	AddressString
Comments:	GSM 09.02, 14.7.8

Type Definition

```
OCTET STRING (SIZE ( 1 .. maxAddressLength ))
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	Asn1Integer
Comments:	INTEGER of ASN.1 type

Type Definition

```
INTEGER
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	BasicServiceCode
Comments:	GSM 09.02, 14.7.8

Type Definition

```
CHOICE {
        bearerService   [2] IMPLICIT BearerServiceCode,
        teleservice        [3] IMPLICIT TeleserviceCode       }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	BasicServiceGroupList
Comments:	GSM 09.02, 14.7.8
Type Definition	
`SEQUENCE SIZE ( 1 .. maxNumOfBasicServiceGroups ) OF BasicServiceCode`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	BearerServiceCode
Comments:	GSM 09.02, 14.7.10
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CallBarringCause
Comments:	GSM 09.02, 14.7.7
Type Definition	

`ENUMERATED {`
`  barringServiceActive  (0),`
`  operatorBarring  (1) }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CallBarringFeature
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`  basicService  BasicServiceCode  OPTIONAL,`
`  ss_Status  [4] IMPLICIT SS_Status  OPTIONAL }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CallBarringFeatureList
Comments:	GSM 09.02, 14.7.4
Type Definition	
`SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF CallBarringFeature`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CallBarringInfo
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`  ss_Code  SS_Code  OPTIONAL,`
`  callBarringFeatureList  CallBarringFeatureList  }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CliRestrictionOption
Comments:	GSM 09.02, 14.7.4
Type Definition	
ENUMERATED { permanent (0), temporaryDefaultRestricted (1), temporaryDefaultAllowed (2) }	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_Feature
Comments:	GSM 09.02, 14.7.4
Type Definition	
SEQUENCE { basicService BasicServiceCode OPTIONAL, preferentialCUG_Indicator CUG_Index OPTIONAL, interCUG_Restrictions InterCUG_Restrictions }	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_FeatureList
Comments:	GSM 09.02, 14.7.4
Type Definition	
SEQUENCE SIZE (1.. maxNumOfBasicServiceGroups) OF CUG_Feature	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_Index
Comments:	GSM 09.02, 14.7.4
Type Definition	
INTEGER (0..32767)	
Detailed Comments:	The internal structure is defined in ETS 300 138.

ASN.1 Type Definition	
Type Name:	CUG_Interlock
Comments:	GSM 09.02, 14.7.4
Type Definition	
OCTET STRING (SIZE (4))	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_Info
Comments:	GSM 09.02, 14.7.4
Type Definition	
SEQUENCE { cug_SubscriptionList CUG_SubscriptionList, cug_FeatureList CUG_FeatureList OPTIONAL }	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_Subscription
Comments:	GSM 09.02, 14.7.4

Type Definition

```
SEQUENCE {
        cug_Index                       CUG_Index,
        cug_Interlock                  CUG_Interlock,
        intraCUG_Options         IntraCUG_Options,
        basicServiceGroupList    BasicServiceGroupList    OPTIONAL }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	CUG_SubscriptionList
Comments:	GSM 09.02, 14.7.4

Type Definition

```
SEQUENCE SIZE (1.. maxNumOfCUG ) OF CUG_Subscription
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	ErrorLocalValues
Comments:	GSM 04.80, 4.5

Type Definition

```
INTEGER {
  unknownSubscriber                                 (1),
  illegalSubscriber                                    (9),
  bearerServiceNotProvisioned                    (10),
  teleserviceNotProvisioned                        (11),
  illegalEquipment                                    (12),
  callBarred                                             (13),
  illegalSS_Operation                                (16),
  ss_ErrorStatus                                      (17),
  ss_NotAvailable                                     (18),
  ss_SubscriptionViolation                         (19),
  ss_Incompatibility                                  (20),
  facilityNotSupported                             (21),
  absentSubscriber                                    (27),
  systemFailure                                          (34),
  dataMissing                                            (35),
  unexpectedDataValue                              (36),
  pw_RegistrationFailure                            (37),
  negativePW_Check                                 (38),
  numberOfPW_AttemptsViolation                (43),
  unknownAlphabet                                    (71),
  ussd_Busy                                              (72),
  maxNumberOfMPTY_ParticipantsExceeded  (126),
  resourcesNotAvailable                             (127)
 }
```

Detailed Comments: Elements of INTEGER are global for the ATS.

ASN.1 Type Definition	
Type Name:	ForwardingFeature
Comments:	GSM 09.02, 14.7.4

Type Definition

```
SEQUENCE {
        basicService                    BasicServiceCode                    OPTIONAL,
        ss_Status                        [4] IMPLICIT SS_Status
OPTIONAL,
        forwardedToNumber         [5] IMPLICIT ISDN_AddressString          OPTIONAL,
        forwardedToSubaddress   [8] IMPLICIT ISDN_SubaddressString     OPTIONAL,
        forwardingOptions           [6] IMPLICIT ForwardingOptions             OPTIONAL,
        noReplyConditionTime      [7] IMPLICIT NoReplyConditionTime         OPTIONAL }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	ForwardingFeatureList
Comments:	GSM 09.02, 14.7.4
Type Definition	
`SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF ForwardingFeature`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	ForwardingInfo
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`  ss_Code  SS_Code  OPTIONAL,`
`  forwardingFeatureList  ForwardingFeatureList }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	ForwardingOptions
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	GeneralProblem
Comments:	ITU-T Q.767
Type Definition	

`TCAP_Problems ( unrecognizedComponent`
`  | mistypedComponent`
`  | badlyStructuredComponent )`	

Detailed Comments:	Type restricted to these three.

ASN.1 Type Definition	
Type Name:	InterCUG_Restrictions
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	bits: 876543: 000000 (unused) bits 21: 00 CUG only facilities 01 CUG with outgoing access 10 CUG with incoming access 11 CUG with both outgoing and incoming access

ASN.1 Type Definition	
Type Name:	IntraCUG_Options
Comments:	GSM 09.02, 14.7.4
Type Definition	

`ENUMERATED {`
`  noCUG_Restrictions  (0),`
`  cugIC_CallBarred  (1),`
`  cugOG_CallBarred  (2) }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	InvokeIDType
Comments:	ITU-T, Q.773
Type Definition	
`INTEGER (-128 .. 127)`	
Detailed Comments:	Values: Sending Components: If it is an invoke component then use Test Case Variable (with default) to set value. If another invoke component is sent the TCV should be incremented beforehand. If it is a return result, error or reject component in response to a received invoke component then use TCV also, making sure the value is set to the value of the received component beforehand. Receiving Components: If it is an invoke comp then use '?'. If it is a return result, error or reject component in response to a sent invoke component then use TCV value (as used in sent invoke component).

ASN.1 Type Definition	
Type Name:	InvokeProblem
Comments:	ITU-T Q.767
Type Definition	

`TCAP_Problems ( duplicateInvokeID`
`| unrecognizedOperation`
`| mistypedArgument`
`| resourceLimitation`
`| initiatingRelease`
`| unrecognizedLinkedID`
`| linkedResponseUnexpected`
`| unexpectedLinkedOperation )`	

Detailed Comments:	Type restricted to these 8.

ASN.1 Type Definition	
Type Name:	ISDN_AddressString
Comments:	GSM 09.02, 14.7.8
Type Definition	
`AddressString (SIZE ( 1 .. maxISDN_AddressLength ))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	ISDN_SubaddressString
Comments:	GSM 09.02, 14.7.8
Type Definition	
`OCTET STRING (SIZE ( 1 .. maxISDN_SubaddressLength ))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	NetworkResource
Comments:	GSM 09.02, 14.7.8,
Type Definition	

`ENUMERATED {`
`plmn (0),`
`hlr (1),`
`vlr (2),`
`pvlr (3),`
`controllingMSC (4),`
`vmsc (5),`
`eir (6),`
`rss (7) }`	

Detailed Comments:	Elements of INTEGER are global for the ATS.

ASN.1 Type Definition	
Type Name:	NoReplyConditionTime
Comments:	GSM 09.02, 14.7.4
Type Definition	
`INTEGER (5 .. 30 )`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	Operation
Comments:	ITU-T Q.767
Type Definition	

`CHOICE {`
`  localValue  INTEGER,`
`  globalValue OBJECT IDENTIFIER}`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	OverrideCategory
Comments:	GSM 09.02, 14.7.4
Type Definition	

`ENUMERATED {`
`  overrideEnabled  (0),`
`  overrideDisabled  (1) }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	Password
Comments:	GSM 09.02, 14.7.4
Type Definition	
`NumericString (FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"))  (SIZE (4))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	PW_RegistrationFailureCause
Comments:	GSM 09.02, 14.7.7
Type Definition	

`ENUMERATED {`
`  undetermined  (0),`
`  invalidFormat  (1),`
`  newPasswordsMismatch  (2) }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	ReturnErrorProblem
Comments:	ITU-T Q.767
Type Definition	

`TCAP_Problems ( unrecognizedInvokeID`
`| returnErrorUnexpected`
`| unrecognizedError`
`| unexpectedError`
`| mistypedParameter )`	

Detailed Comments:	Type restricted to these 5.

ASN.1 Type Definition	
Type Name:	ReturnResultProblem
Comments:	ITU-T Q.767
Type Definition	

`TCAP_Problems ( unrecognizedInvokeID`
`| returnResultUnexpected`
`| mistypedResult )`	

Detailed Comments:	Type restricted to these three.

ASN.1 Type Definition	
Type Name:	SS_Code
Comments:	GSM 09.02, 14.7.5
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	group (bits 8765), and specific service (bits 4321)

ASN.1 Type Definition	
Type Name:	SS_Data
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`ss_Code SS_Code OPTIONAL,`
`ss_Status [4] IMPLICIT SS_Status OPTIONAL,`
`ss_SubscriptionOption SS_SubscriptionOption OPTIONAL,`
`basicServiceGroupList BasicServiceGroupList OPTIONAL }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_ForBS_Code
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`ss_Code SS_Code,`
`basicService BasicServiceCode OPTIONAL }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_IncompatibilityCause
Comments:	GSM 09.02, 14.7.8
Type Definition	

`SEQUENCE {`
`ss_Code [1] IMPLICIT SS_Code OPTIONAL,`
`basicService BasicServiceCode OPTIONAL,`
`ss_Status [4] SS_Status OPTIONAL }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_Info
Comments:	GSM 09.02, 14.7.4
Type Definition	

`CHOICE {`
`forwardingInfo [0] IMPLICIT ForwardingInfo,`
`callBarringInfo [1] IMPLICIT CallBarringInfo,`
`cug_Info [2] IMPLICIT CUG_Info,`
`ss_Data [3] IMPLICIT SS_Data }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_Status
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	bits 8765: 0000 unused, bits 4: SS state information Q bit, bits 3: SS state information P bit, bits 2: SS state information R bit, bits 1: SS state information A bit.

ASN.1 Type Definition	
Type Name:	SS_SubscriptionOption
Comments:	GSM 09.02, 14.7.4
Type Definition	

`CHOICE {`
`  overrideCategory  [1] IMPLICIT OverrideCategory,`
`  cliRestrictionOption  [2] IMPLICIT CliRestrictionOption }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_UserData
Comments:	GSM 09.02, 14.7.4
Type Definition	
`IA5String (SIZE (1  .. maxSignalInfoLength))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	TCAP_Problems
Comments:	ITU-T Q.767
Type Definition	

`INTEGER {`
`  unrecognizedComponent  (0),  -- GeneralProblem`
`  unrecognizedInvokeID  (0),  -- ReturnResultProblem, ReturnErrorProblem`
`  duplicateInvokeID  (0),  -- InvokeProblem`
`  mistypedComponent  (1),  -- GeneralProblem`
`  returnErrorUnexpected  (1),  -- ReturnErrorProblem`
`  returnResultUnexpected  (1),  -- ReturnResultProblem`
`  unrecognizedOperation  (1),  -- InvokeProblem`
`  badlyStructuredComponent  (2),  -- GeneralProblem`
`  unrecognizedError  (2),  -- ReturnErrorProblem`
`  mistypedArgument  (2),  -- InvokeProblem, originally called mistypedParameter in TCAP`
`  mistypedResult  (2),  -- ReturnResultProblem`
`  resourceLimitation  (3),  -- InvokeProblem`
`  unexpectedError  (3),  -- ReturnErrorProblem`
`  mistypedParameter  (4),  -- ReturnErrorProblem`
`  initiatingRelease  (4),  -- InvokeProblem`
`  unrecognizedLinkedID  (5),  -- InvokeProblem`
`  linkedResponseUnexpected  (6),  -- InvokeProblem`
`  unexpectedLinkedOperation  (7)  -- InvokeProblem`
`  }`	

Detailed Comments:	Errors of the same integer value are distinguished by their different parent types (General, Invoke, ReturnResult, ReturnError).

ASN.1 Type Definition	
Type Name:	TeleserviceCode
Comments:	GSM 09.02, 14.7.9
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	USSD_Arg
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`  ussd_DataCodingScheme  USSD_DataCodingScheme,`
`  ussd_String  USSD_String }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	USSD_Res
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`  ussd_DataCodingScheme  USSD_DataCodingScheme,`
`  ussd_String  USSD_String }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	USSD_DataCodingScheme
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	USSD_String
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1..maxUSSD_StringLength))`	
Detailed Comments:	The structure of the contents of the USSD-String is dependent on the USSD-DataCodingScheme as described in TS GSM03.38.

Test suite operation definitions

Test Suite Operation Definition	
Operation Name:	OC_Asn1intToOct(n:Asn1Integer; l: INTEGER)
Result Type:	OCTETSTRING
Comments:	
Description	
OC_Asn1intToOct converts the ASN.1 INTEGER `n` into OCTETSTRING with length = `l`. for example: OC_Asn1intToOct(14,1) = '0E'O; OC_Asn1intToOct(18,1) = '12'O; OC_Asn1intToOct(18,2) = '0012'O; OC_Asn1intToOct(-128,1) = '80'O (MSB (position p out of 1...p) represents -2exp(p-1)); OC_Asn1intToOct(-32768,2) = '8000'O (MSB (position p out of 1...p) represents -2exp(p-1)).	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_Bcap(setup:SETUP_MO_PDU; callproc:CALL_PROC_PDU; mem:INTEGER)
Result Type:	BCAP
Comments:	

Description

OC_Bcap operation returns a bearer capability IE according the following rule:

- for `mem` = 1:
  - if bearer capability IE were presented in `callproc` the returned BC is the bearer capability 1 of the `callproc`;
  - otherwise, the returned bearer capability is the bearer capability 1 of the `setup`.

- for `mem` = 2:
  - if bearer capability IE were presented in `callproc` the returned BC is the bearer capability 2 of the `callproc`;
  - otherwise, the returned bearer capability is the bearer capability 2 of the `setup`.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_BinAdd(bitstr1:BITSTRING; bitstr2:BITSTRING)
Result Type:	BITSTRING
Comments:	

Description

OC_BinAdd operation performs binary addition of two input parameters `bitstr1` and `bitstr2`, then returns the result of the addition. These two input parameters shall have the same length, the result of the operation has the same length as the input parameters.

for example:

OC_BinAdd('01000'B, '00110'B) = '01110'B;
OC_BinAdd('01000'B, '00011'B) = '01011'B;
OC_BinAdd('00100'B, '00010'B) = '00110'B.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_Bit7(bstring: B_8)
Result Type:	BITSTRING
Comments:	The input parameter bstring is of type BITSTRING[8].

Description

OC_Bit7(bstring) returns the value of bit 7 in the `bstring`.

for example:

OC_Bit7('01010101'B) = '1'B,
OC_Bit7('10101010'B) = '0'B

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_BCDtoInt(bcdstring:HEXSTRING; relevant_digits: INTEGER)
Result Type:	INTEGER
Comments:	

Description

The operation OC_BCDtoInt converts last 'relevant_digits' of an HEXSTRING containing BCD coded digits to an integer representation of these relevant digits.

Example: OC_BCDtoInt('12345'H, 3) := 345

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_CalledNumCHK(callednum:OCTETSTRING; dialnum:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
the operation OC_CalledNumCHK to check whether the called party number `callednum`, which is represented by OCTETSTRING, is the same as the dialed number 'dialnum`, which is represented by IA5String. It returns TRUE if they are the same, otherwise FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_ChkSRES(sres:OCTETSTRING; ki: BITSTRING; rand:BITSTRING)
Result Type:	BOOLEAN
Comments:	sres is 32 bits value.
Description	
OC_ChkSRES checks the input parameter `sres` according to the authentication algorithm defined in the following procedure. It returns TRUE if the `sres` is correct, otherwise it returns FALSE. - firstly the `ki` XOR to the `rand` results in RES1; - then compare the most significant 32 bits of the RES1 with the `sres`; - if they are equal, the `sres` is correct and the operation returns TRUE; - if they are not equal, the `sres` is wrong and the operation returns FALSE. NOTE: this procedure is the test algorithm for authentication defined by GSM.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_CnvtMax(max:INTEGER)
Result Type:	B_2
Comments:	
Description	
The allowed integer values for max are 1, 2, 4, 7 (maximum number of retransmissions). The result BITSTRING is 2 bits long. OC_CnvtMax converts the input integer `max` into a BITSTRING according to the following rule: 1. the result is '00'B if the `max` = 1, 2. the result is '01'B if the `max` = 2, 3. the result is '10'B if the `max` = 4, 4. the result is '11'B if the `max` = 7. for example: OC_CnvtMax(1) = '00'B, OC_CvntMax(7) = '11'B.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_CnvtTx(tx:INTEGER)
Result Type:	B_4
Comments:	The result BITSTRING is 4 bits long.

Description

OC_CnvtTx converts the input integer `tx` into a BITSTRING according the following rule:

1. the result is '0000'B if the `tx` = 3,
2. the result is '0001'B if the `tx` = 4,
3. the result is '0010'B if the `tx` = 5,
4. the result is '0011'B if the `tx` = 6,
5. the result is '0100'B if the `tx` = 7,
6. the result is '0101'B if the `tx` = 8,
7. the result is '0110'B if the `tx` = 9,
8. the result is '0111'B if the `tx` = 10,
9. the result is '1000'B if the `tx` = 11,
10. the result is '1001'B if the `tx` = 12,
11. the result is '1010'B if the `tx` = 14,
12. the result is '1011'B if the `tx` = 16,
13. the result is '1100'B if the `tx` = 20,
14. the result is '1101'B if the `tx` = 25,
15. the result is '1110'B if the `tx` = 32,
16. the result is '1111'B if the `tx` = 50.

for example:

OC_CnvtTx(3) = '0000'B,
OC_CvntTx(5) = '0010'B.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_CodeSMSCBMessage(firstoct: INTEGER; lastoct: INTEGER)
Result Type:	OCTETSTRING
Comments:	

Description

The operation codes a part of the contents for a cell broadcast short message. The cell broadcast short message, is 82 octets long, i.e. 93 characters, each represented by 7 bits. As many as possible different characters are sent, the characters are those corresponding to the 7-bit representation of the integers 0, 1, 2, ..., 92. The bits are arranged acc. to GSM 03.38, clause 6.1.2.1.1. The result of this operation is the octetstring of the octets 'firstoct' to 'lastoct' (16 octets for the first message block, 22 octets for the 2nd, 3rd and 4th blocks), with the octets of the cell broadcast short message being numbered from 1 to 82.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_CodingOfUssdString(text: IA5String)
Result Type:	OCTETSTRING
Comments:	

Description

This operation provides the coding for a USSD String 'text' in the default alpabet, each character represented by 7 bits. The bits are arranged acc. to GSM 03.38, clause 6.1.2.2.1.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_ComputeSMContents(NumberOfCharacters: INTEGER)
Result Type:	OCTETSTRING
Comments:	max. 160 characters, i.e. 140 octets.
Description	
This operation provides a short message's contents with a specified number of characters 'NumberOfCharacters', each represented by 7 bits. As possible different characters are sent, the characters are those corresponding to the 7-bit representation of 0, 1, 2, ... up to ('NumberOfCharacters' - 1). If more than 128 characters are sent, the rest of the characters is the corresponding to 0, 1, ... up to (NumberOfCharacters - 128 - 1), e.g. for 160 characters: 0, 1, ..., 127, 0, 1, ..., 31. The bits are arranged acc. to GSM 03.38, clause 6.1.2.1.1.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_ComputeSMContentsSpecText(NumbOfIA5chara: INTEGER; text: IA5String)
Result Type:	OCTETSTRING
Comments:	max. 160 characters, i.e. 140 octets.
Description	
This operation provides a short message's contents with a specified number of characters 'NumbOfIA5chara', each represented by 7 bits. 'text' is used as contents of the short message. If "text' contains less than 'NumberOfCharacters' characters, 'text' is repeated until the short message reaches the 'NumberOfCharacters' characters long. The bits packing is arranged according. to GSM 03.38, clause 6.1.2.1.1.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_CphKeyGen(ki:BITSTRING; rand:BITSTRING)
Result Type:	BITSTRING
Comments:	both ki and rand are 128 bits values, the result of the operation is 64 bits value
Description	
OC_CphKeyGen generates the ciphering key from the input parameters according to the following procedure: - firstly the `ki` XOR to the `rand` results in RES1; - then discard the most significant 32 bits of the RES; - the next 64 bits of RES1 are the ciphering key, the operation returns this value. - the 32 least singnificant bits are not used.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_FirstDigi(bcddigits:HEXSTRING)
Result Type:	B_4
Comments:	
Description	
The input parameter bcddigits shall be a BCD string (subset of HEXSTRING), the resut is a BITSTRING[4] of a binary representation of one BCD digit. The function of the OC_FirstDigi is to return the first (most significant) digit of the input parameter `bcddigits`. for example: OC_FirstDigi('12345') = '0001'B, OC_FirstDigi('012345678') = '0000'B.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_FnInc(fn:FN; inc:INTEGER)
Result Type:	FN
Comments:	

Description

The function of OC_FnInc operation to increase frame number with `inc`. The frame number to be incremented is the input parameter `fn` in FN type and the increment `inc` is in INTEGER type, the incremented frame number is returned in FN type.

```
OC_FnInc(fn, inc)
FN     fn;
INTEGER  inc;
{
int     fmrin;
FN     frmout;

   frmin = 51 * ((fn.t3 - fn.t2) MOD 26) + fn.t3 + 1326 * fn.t1_ ;

   frmout.t1_ = ((frmin + inc) DIV 1326) MOD 32;
   frmout.t2   = (frmin + inc) MOD 26;
   frmout.t3   = (frmin + inc) MOD 51;
   return (frmout);
}
```

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_GetSCTimeStamp(timezone:TZONES)
Result Type:	TPSCTS
Comments:	TPSCTS is HEXSTRING[14]

Description

This Operation provides the hexstring containing the service center time stamp (SCTS) according to GSM 03.40, clauses 9.2.2.1 and 9.2.3.11. The TSO reads the current time of the test systems clock and transforms the time in conbination with the input parameter 'timezone' into a service center time stamp.

Example:

1996 April 18, 15:32:46, timezone=4

OC_GetSCTimeStamp returns 69408151236440

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_IncTmsi(tmsi:TMSI_V; inc:OCTETSTRING)
Result Type:	TMSI_V
Comments:	the `tmsi` is 4 OCTETs long

Description

OC_IncTmsi operation adds the two input parameters and returns the result. An overflow of addition is allowed.

For example :

OC_IncTmsi('33542140'O + '01'O) = '33542141'O;
OC_IncTmsi('21322140'O + '08'O) = '21322148'O.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_InRang(tx: INTEGER; maxret:INTEGER; m: INTEGER)
Result Type:	BOOLEAN
Comments:	
Description	
The operation returns TRUE if (m DIV (maxret * ((230+maxret -1)/maxret))) is inside the following interval: [0.8 - ((tx+1)/2) DIV tx; 1.2 - ((tx+1)/2) DIV tx] where / is integer division, DIV is float division, m, tx and maxret are input parameters of the operation.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_IntToOct(n:INTEGER; l: INTEGER)
Result Type:	OCTETSTRING
Comments:	
Description	
OC_IntToOct converts the INTEGER `n` into OCTETSTRING with length = `l`. for example: OC_IntToOct(14,1) = '0E'O; OC_IntToOct(18,1) = '12'O; OC_IntToOct(18,2) = '0012'O.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LeastBits(bstring:BITSTRING; lg:INTEGER)
Result Type:	BITSTRING
Comments:	
Description	
OC_LeastBits operation returns the `lg` least significant bits of the original `bstring`. for example: OC_LeastBits('110011000101010'B, 3) = '010'B, OC_LeastBits('110011000101010'B, 6) = '101010'B.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LengthOf(identity_field :MI)
Result Type:	INTEGER
Comments:	
Description	
OC_LengthOf operation returns the actual length of the `identity_field`, the unit of length is in OCTET.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LengthOfBCDN(bcdn :BCDN)
Result Type:	INTEGER
Comments:	
Description	
OC_LengthOfBCDN operation returns the actual length of an IE of type BCDN 'bcdn', the unit of length is in OCTET.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LengthOfComp(comp: Component_T)
Result Type:	LENGTH
Comments:	
Description	
OC_LengthOfComp operation returns the actual length of the Component_T `comp`, the unit of length is in OCTET.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LengthOfString(strg: IA5String)
Result Type:	INTEGER
Comments:	
Description	
OC_LengthOfString operation returns the actual length (number of characters) of the string `strg`.	
Detailed Comments:	

Test Suite Operation Definition

Operation Name:	OC_LookupS(t: INTEGER; combined :BOOLEAN)
Result Type:	INTEGER
Comments:	the algorithm is derived from Table 3.1/GSM 04.08 for values of parameter S.

Description

This operation returns an INTEGER according the following algorithm:

```
OC_LookupS( t, comb)
INTEGER t;
BOOLEAN combined;
{
    switch (t)
        {  case 3:
           case 8:
           case 14:
           case 50:
              if (combined) then
                      return(41)
                 else   return(55);
              break;
           case 4:
           case 9:
           case 16:
              if (combined) then
                      return(52)
                 else  return(76);
              break;
           case 5:
           case 10:
           case 20:
              if (combined) then
                      return(58)
                 else  return(109);
              break;
           case 6:
           case 11:
           case 25:
              if (combined) then
                      return(86)
                 else  return(163);
              break;
           case 7:
           case 12:
           case 32:
              if (combined) then
                      return(115)
                 else  return(217);
              break;
     }
}
```

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_MostBits(bstring:BITSTRING; lg :INTEGER)
Result Type:	BITSTRING
Comments:	
Description	
OC_MostBits operation returns the `lg` most significant bits of the original `bstring`. for example: OC_LeastBits('110011000101010'B, 3) = '110'B, OC_LeastBits('110011000101010'B, 6) = '110011'B.	
Detailed Comments:	

Test Suite Operation Definition

Operation Name:	OC_MsrReptChk(msrres: MSRR; index:INTEGER)
Result Type:	BOOLEAN
Comments:	parameter index is used for selecting different match pattern

Description

OC_MsrReptChk operation checks whether the received measurement report contains correct values:

1. when the `index` = 1:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00000'B, '001011'B}, {'01111'B, '001001'B},
   {'10101'B, '001011'B}, {'10110'B, '001101'B},
   {'11100'B, '001111'B}, {'11111'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

2. when the `index` = 2:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00000'B, '001011'B}, {'00100'B, '001001'B},
   {'00110'B, '001011'B}, {'00101'B, '001101'B},
   {'00010'B, '001111'B}, {'00111'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

3. when the `index` = 3:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00001'B, '001011'B}, {'00101'B, '001001'B},
   {'00111'B, '001011'B}, {'00110'B, '001101'B},
   {'00011'B, '001111'B}, {'01000'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

4. when the `index` = 4 :
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   }
   equals to
   set { {'00000'B, '001011'B}, {'00010'B, '001111'B},
   {'00001'B, '001111'B}, {'00011'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

5. when the `index` = 5 :
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   }
   equals to
   set { {'00000'B, '001011'B}, {'00010'B, '001111'B},
   {'00001'B, '001111'B}, {'00100'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

6. when the `index` = 6:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00000'B, '001011'B}, {'00011'B, '001001'B},
   {'00100'B, '001011'B}, {'00101'B, '001101'B},
   {'00110'B, '001111'B}, {'00111'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

7. when the `index` = 7:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00000'B, '001011'B}, {'00100'B, '001001'B},
   {'00110'B, '001011'B}, {'00101'B, '001101'B},
   {'00010'B, '001111'B}, {'00111'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

8. when the `index` = 8 :
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2}
   }
   equals to
   set { {'00000'B, '001011'B}, {'00010'B, '001101'B}
   }
   then the operation returns TRUE otherwise FALSE.

9. when the `index` = 9 :
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2}
   }
   equals to
   set { {'00110'B, '001011'B}, {'00111'B, '001101'B}
   }
   then the operation returns TRUE otherwise FALSE.

10. when the `index` = 10 :
    if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2}
    }
    equals to
    set { {'00011'B, '001011'B}, {'00101'B, '001101'B}
    }
    then the operation returns TRUE otherwise FALSE.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_OeBit(bcddigits:HEXSTRING)
Result Type:	BITSTRING
Comments:	The input parameter `bcddigits` is really BCD string (subset of HEXSTRING), the result is BITSTRING[1].
Description	
The function of the OC_OeBit is as the following: 1. it returns '1'B, if the length of the `bcddigits` is odd, 2. it returns '0'B, if the length of the `bcddigits` is even. for example: OC_OeBit('12583') = '1'B, OC_OeBit('87259957') ='0'B.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_OctToInt(ostr: OCTETSTRING)
Result Type:	INTEGER
Comments:	
Description	
OC_OctToInt converts the OCTETSTRING `ostr` into INTEGER.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_OctToInvokeIDType(o:OCTETSTRING)
Result Type:	InvokeIDType
Comments:	
Description	
OC_OctToInvokeIDType converts the OCTETSTRING `o` into InvokeIDType, with the MSB of 'o' representing the negative value -2exp(n-1), for example '80'O->'10000000'B-> -128. The rest of the bits can code positive values up to +2exp(n-2)-1, for example '4F'O->'01111111'B-> +127. for example: OC_OctToInvokeIDType('80'O) = -128; OC_OctToInvokeIDType('81'O) = -128+1 = -127; OC_OctToInvokeIDType('40'O) = +64. OC_OctToInvokeIDType('7F'O) = +127. OC_OctToInvokeIDType('C0'O) = -128+64 = -64.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_OtherDigi(bcddigits:HEXSTRING)
Result Type:	OCTETSTRING
Comments:	The input parameter` bcddigits` is really BCD string (subset of HEXSTRING), the result is an even BCD digits except that the next to last may either be 'F'H or a BCD digit.

Description

The function of the OC_OtherDigi is as the following:

1. If the number of the `bcddigits` is odd, the operation removes the most significant digit, and then reverses the order of each pair of digits;
2. If the number of the `bcddigits` is even, first the operation suffixes the `bcddigits` with 'F'H, then removes the most significant digit, and then reverses the order of each pair of digits.

for example:

OC_OtherDigi('12345') = '3254',
OC_OtherDigi('12345678') ='325476F8'.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_PosinSet(set: Components; comp: Component)
Result Type:	INTEGER
Comments:	

Description

The operation OC_PosinSet returns the position of component `comp` within the SET `set`.

for example:

if the set = { registerSSComponents, eraseSSComponents, activateSSComponents, deactivateSSComponents }

OC_PosinSet(set, registerSSComponents) = 0,
OC_PosinSet(set, activateSSComponents) = 2.

Detailed Comments:

Test Suite Operation Definition

Operation Name: OC_RachSlots(fn1:FN; fn2:FN; cmbn: BOOLEAN; mode:INTEGER)

Result Type: INTEGER

Comments:

Description

OC_RachSlots calculates the number of RACH slots between frame number `fn1` and `fn2` excluding the slots in `fn1` and `fn2`, then return the result of the calculation.

If mode=0, 'fn1' denotes the (first) frame number for sending Paging Request, while 'fn2' denotes the (first) frame number for the received Channel Request. If mode=1 'fn1' and 'fn2' denote the (first) frame number for the received two subsequent Channel Requests. Any other values for mode is not allowed.

Since the number of RACH slots depends on the RACH being combined or not with dedicated channels, the parameter 'cmbn' is needed: TRUE --> combined, FALSE--> not combined.

It is assumed that the distance between fn1 and fn2 is not more than one group of 42432 TDMA frames (modulo 42432 is used to calculate t1' of fn1 and fn2). This is equivalent to approximately 196 seconds. The 51 TDMA frames in a multiframe are numbered 0 to 50.

fn1' := fn1 mod 51

fn2' := fn2 mod 51

CASE 1: cmbn is FALSE ('not combined')

When 'not combined', all slots are RACH slots.

1.1 mode=0 (sending paging message at fn1)

The paging uses the numbered frames 6 - 9, 12 -19, 22-29, 32-39 and 42-49.

1.1.1 If fn1 is not in the paging TDMA frame mapping, it leads testing to a test system error.

IF (fn1' < 6) OR (9 < fn1' < 12) OR (19 < fn1' < 22) OR (29 < fn1' < 32) OR (39 < fn1' < 42) OR (fn1' >49),

A test system error! The calling test case shall re-run again.

OTHERWISE

1.1.2 The number of RACH slots is equal to the number of frames between fn1 + 3 and fn2. The fn1 is added by 3 because a page message occupies 4 slots. The fn1 indicates the first slot of the paging message.

IF (fn1 + 3) < fn2, OC_RachSlots := fn2 - (fn1 + 3) -1 = fn2 - fn1 - 4,

1.1.3 If fn1+3 is equal to or greater than fn2, then this is due to fn2 being in the next group of 42432 frames. In this case 42432 frames have to be added.

IF fn2 <= (fn1 + 3), OC_RachSlots := fn2 -fn1 - 3 + 42432 -1 = fn2 -fn1 + 42428

1.2 mode=1 (receiving channel request at fn1)

The number of RACH slots is equal to the number of frames between fn1 and fn2

1.2.1 IF fn1 < fn2, OC_RachSlots := fn2 - fn1 - 1,

1.2.2 IF fn2 <= fn1, OC_RachSlots := fn2 -fn1 + 42432 -1 = fn2 -fn1 + 42431

CASE 2: cmbn is TRUE ('combined')

When combined only the slots of the numbered frames 4, 5, 14 to 36, 45 and 46 in each multiframe are RACH slots, i.e.total 27 frames per multiframe.

2.1 mode=0 (sending paging message at fn1)

The paging uses the numbered frames 6 - 9 and 12 -19.

2.1.1 If fn1 is not in the paging TDMA frame mapping, it leads testing to a test system error. If fn2 is not in the RACH TDMA frame mapping, it leads testing to fail.

IF (fn2` < 4) OR (5 < fn2` < 14) OR (36 < fn2` < 45) OR (fn2' > 46), OC_RachSlots := -9999

IF (fn1' < 6) OR (9 < fn1' < 12) OR (fn1' >19), A test system error! The calling test case shall re-run again.

OTHERWISE

2.1.2 Calculation of the number 'c' multiframes between fn1 + 3 and fn2. If fn1 + 3 is equal to or greater than fn2, then this is due to fn2 being in the next group of 42432 frames. In this case 42432 frames have to be added. '/' shall be the integer division, i.e. the result is also integer. Fractions are discarded.

2.1.2.1 IF (fn1 + 3) < fn2, c := fn2 / 51 - (fn1 + 3) / 51

2.1.2.2 IF fn2 <= (fn1 + 3), c := fn2 / 51 - (fn1 + 3) / 51 + 42432 / 51 = fn2 / 51 - (fn1 + 3) / 51 + 832

2.1.3 Calculation of the number of frames 'a' to be subtracted according to the position of fn1' within the multiframe

IF (5 < fn1` < 9) , a := 2

IF (fn1` =9), a := 3
IF (11<fn1` <20), a := fn1'-8

2.2 mode=1 (receiving channel request at fn1)

2.2.1 If fn1 or fn2 are not in the RACH TDMA frame mapping, it leads testing to fail.
IF (fn2` < 4) OR (5 < fn2` < 14) OR (36 < fn2` < 45) OR (fn2' > 46)
OR (fn1' < 6) OR (9 < fn1' < 12) OR (fn1' >19), OC_RachSlots := -9999

2.2.2 Calculation of the number 'c' multiframes between fn1 and fn2
2.2.2.1 IF fn1 < fn2, c := fn2 / 51 - fn1 / 51
2.2.2.2 IF fn2 <= fn1 c := fn2 / 51 - fn1 / 51 + 42432 / 51 = fn2 / 51 - fn1/ 51 + 832

2.2.3 Calculation of the number of frames 'a' to be subtracted according to the position of fn1' within the multiframe
IF (3 < fn1' < 6:), a:= fn1' - 3
IF (13 < fn1' < 37), a:= fn1' - 11
IF (44 < fn1' < 47), a:= fn1' - 19

2.3 Calculation of the number of slots 'b' to be added according to the position of fn2' within the multiframe
IF (3 < fn2' < 6:), b:= fn2' - 4
IF (13 < fn2' < 37), b:= fn2' - 12
IF (44 < fn2' < 47), b:= fn2' - 20

2.4 Calculation of the number of RACH slots. There are 27 RACH slots in each multiframe.
OC_RachSlots := 27 * c + b - a

Detailed Comments: TC_26_2_1_1 uses mode=0. TC_26_2_1_2 uses mode=1.

Test Suite Operation Definition

Operation Name: OC_Random(n1:INTEGER; n2:INTEGER)
Result Type: INTEGER
Comments:

Description

This operation randomly returns one number from the following candidates:
'n1', 'n1'+1, ..., 'n2'

Detailed Comments:

Test Suite Operation Definition

Operation Name: OC_RcsdPresent(msg:MODIFY_PDU)
Result Type: BOOLEAN
Comments: To check if RCSD IE is present or not in Modify PDU

Description

IF RCSD IE is present in the PDU passed on input paremeter return TRUE ELSE return FALSE.

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OC_SaveAndProc1(val:INTEGER; mode :INTEGER; cnt:INTEGER; cmbnd:BOOLEAN)
Result Type:	BOOLEAN
Comments:	This operation is used for storing and analysing the CHANNEL REQUEST message distribution.

Description

The function of this operation is defined as an pseudo C code:

```
#define SAVE 0;
#define PROC 1;

OC_SaveAndProc(val, mode,cnt)
INTEGER       val, mode,cnt;
BOOLEAN     combined;
{
  static INTEGER   buf[200];
   int i, j, n;
   if (mode == SAVE)
       { cnt = cnt mod 200;
         if (cmbnd == C_NotCombined) AND (val<(151 + 8)) then
                 {  buf[cnt] = val;
                   return (TRUE);
                  }
         if (cmbnd == C_Combined) AND (val<(81 + 8)) then
                 {  buf[cnt] = val;
                   return (TRUE);
                  }
         else return (FALSE);
        }
   if (mode == PROC)
        {
          for (i=0,  i<200, i++)
               {  n=0;
                  for (j=0, j<200, j++)
                      if (buf[j] == buf[i])
                          n = n+1;
                  if (n > 41) return (FALSE);
               }
            return(TRUE);
         }
}
```

Detailed Comments:	This test suite operation has two operation modes : - when the mode is SAVE and the val is less than 89 for combined or 159 for non-combined, it saves the value val into internal buffer and returns TRUE, otherwise returns FALSE. 200 values will be stored in the internal buffer when the operation is invoked 200 times. - When the mode is PROC, it analyses the values stored in the internal buffer, if no more than 41 of them are equal the operation returns TRUE, otherwise returns FALSE. (i.e. for all n, CARD{ k|f(k) = n } <= 41)

Test Suite Operation Definition	
Operation Name:	OC_SaveAndProc3(val:BITSTRING; mode :INTEGER; cnt:INTEGER)
Result Type:	BOOLEAN
Comments:	This operation is used for storing and analysing the random reference.

Description

The function of this operation is defined as an pseudo C code:

```
#define SAVE 0;
#define PROC 1;

OC_SaveAndProc(val, mode,cnt)
BITSTRING[8] val;
INTEGER        mode,cnt;
{
  static BITSTRING[8]   buf[7];
   int i, j, n=7;
   if (mode == SAVE)
       { cnt = cnt mod 7;
         buf[cnt] = val;
         return(TRUE);
        }
   if (mode == PROC)
        {
          for (i=0,  i<6, i++)
             { for (j=i+1, j<7, j++)
                  if (buf[j] == buf[i])
                    { n = n-1;
                      break;
                    }
             }
          if (n>=4) then
                  return(TRUE)
          else    return (FALSE);
         }

}
```

Detailed Comments:	The function of this test suite operation is : - it saves the value val into internal buffer when the mode = SAVE. 7 values will be stored in the internal buffer when the operation is invoked 7 times. - it compares the values stored in the internal buffer when the mode = PROC, if 4 or more than 4 of them are different the operation returns TRUE, otherwise returns FALSE.

Test Suite Operation Definition

Operation Name:	OC_SaveAndRetrv(val:RQR; mode, idx, ex: INTEGER)
Result Type:	RQR
Comments:	This operation is used for storing and retrieving the random reference.

Description

The function of this test suite operation is (in prose):

- it saves the value val into internal buffer buf[idx] when the mode = SAVE and ex = 0. Maximum 9 values can be stored. The idx is ranged between 1-9.
- it returns the value stored in the internal buffer buf[idx] when the mode = RETRV and ex = 0.
- it returns a value which is different from any value stored in the buf and also different from any values which have been returned by the consecutive invocation of this operation with ex <> 0, if ex <> 0. The idx is ranged between 10-13.

The function of this operation is defined as an pseudo C code:

```
#define SAVE 0;
#define RETRV 1;

OC_SaveAndRetrv(val, mode, idx, ex)
RQR             val;
INTEGER         mode, idx, ex;
{
  int i, j;
  static RQR   buf[14];
   if ((mode == SAVE) AND (ex == 0))
       {
         buf[idx] = val;
         buf[10].ra = '00000000'B;
         buf[10].fn = '00'O;
         buf[11].ra = '00000000'B;
         buf[11].fn = '00'O;
         buf[12].ra = '00000000'B;
         buf[12].fn = '00'O;
         buf[13].ra = '00000000'B;
         buf[13].fn = '00'O;
         return (buf[10];
        }
   if ((mode == RETRV) AND (ex == 0))
         return (buf[idx]);
   if (ex != 0)
         {
           buf[0].ra = buf[1].ra + '00000001'B;
           buf[0].fn = buf[1].fn;
           for (j=1, j <13, j++)
              {
                for (i=1, i<14, i++)
                  if( buf[0] == buf[i])
                      { buf[0].ra = buf[i].ra + '00000001'B;
                        break;
                       }
                if (i == 14)
                 {
                   buf[idx].ra = buf[0].ra:
                   return (buf[0]);
                 }
              }
          buf[idx].ra = buf[i].ra+'00000001'B;
          return (buf[idx]);
         }
}
```

Detailed Comments:	0. The pseudo C code gives one of the possible implementations for the OC description in prose. 1. '00'O should be understood as equivalence of Fn_01, a TTCN structured type constraint. 2. The OC can save maximum 13 RQR values. In SAVE mode, idx is ranged between 1-9.

	3. For ex<>0, there are maximum 13 scans. In each scan a new value is assigned to buf[0].ra. If there is a 'non-match' in one of the 13 scans the buf[0] is returned. In case of 'all-match' for the 13 scans, the pair of (buf[i].ra + '00000001' at the last scan, buf[idx].fn is returned (idx is ranged between 10 - 13).

Test Suite Operation Definition

Operation Name: OC_StartTime(frmn:FN; t, i:INTEGER)
Result Type: STRT
Comments:

Description

OC_StartTime operation generates the STARTING TIME IE according to the input parameters.

```
(* frmn is the current frame number,
   t is the "delay" in applying the new frequencies,
   i is the contents of the starting time IE. *)
OC_StartTime(frmn, t, i)
INTEGER  t, i;
FN          frmn;
{
int tmp;
STRT strt;

tmp = 51 * ((frmn.t3 - frmn.t2) MOD 26) + frmn.t3 + 1326 * frmn.t1_ ;
tmp = (tmp + t) MOD 42432;
strt.fn.t1_ = (tmp DIV 1326) MOD 32;
strt.fn.t2   = tmp MOD 26;
strt.fn.t3   = tmp MOD 51;
strt.iei = OMIT;
if (i == 1) strt.iei := '01111100'B  /* if i=1 the information element identifier shall be included \
```

Detailed Comments:

Test Suite Operation Definition

Operation Name: OC_SubOctet(src:OCTETSTRING; len :INTEGER)
Result Type: OCTETSTRING
Comments:

Description

OC_SubOctet(src, len) is the octetstring of length `len` starting from the leftmost position of the source octetstring `src`.

For example : OC_SubOctet('123456789ABCDEF'O, 4) = '12345678'O

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OC_SubchOfFacchh(subch: B_1; cell:CellID; inst:INTEGER)
Result Type:	LOGICCH
Comments:	subch is of type BITSTRING[1]

Description

OC_SubchOfFacchh operation returns a logic channel identifier for FACCHH subchannel indicated by the input parameters `subch`, `cell` and `inst`, where 'subch' is the subchannel number, `cell` is the cell identifier which the channel belongs to, `inst` is the instance of the channel.

for example:

OC_SubchOfFacchh('0'B, C_CellA, 1) = C_FACCHH0_A_1;
OC_SubchOfFacchh('1'B, C_CellA, 1) = C_FACCHH1_A_1;
OC_SubchOfFacchh('0'B, C_CellB, 1) = C_FACCHH0_B_1;
OC_SubchOfFacchh('1'B, C_CellB, 1) = C_FACCHH1_B_1.

pseudo C code definition for the operation as following :

```
OC_SubchOfFacchh(subch, cell, inst)
BITSTRING   subch;
CellID     cell;
INTEGER   inst;
{
   LOGICCH    logch;
   if (subch=='0'B && cell == C_CellA)
      switch (inst)
        {
          case 1 : logch = C_FACCHH0_A_1; break;
          case 2 : logch = C_FACCHH0_A_2; break;
          case 3 : logch = C_FACCHH0_A_3; break;
        }
  if (subch=='0'B && cell == C_CellB)
      switch (inst)
        {
          case 1 : logch = C_FACCHH0_B_1; break;
          case 2 : logch = C_FACCHH0_B_2; break;
          case 3 : logch = C_FACCHH0_B_3; break;
        }
  if (subch=='0'B && cell == C_CellC)
      switch (inst)
        {
          case 1 : logch = C_FACCHH0_C_1; break;
          case 2 : logch = C_FACCHH0_C_2; break;
          case 3 : logch = C_FACCHH0_C_3; break;
        }
   if (subch=='1'B && cell == C_CellA)
      switch (inst)
        {
          case 1 : logch = C_FACCHH1_A_1; break;
          case 2 : logch = C_FACCHH1_A_2; break;
          case 3 : logch = C_FACCHH1_A_3; break;
        }
  if (subch=='1'B && cell == C_CellB)
      switch (inst)
        {
          case 1 : logch = C_FACCHH1_B_1; break;
          case 2 : logch = C_FACCHH1_B_2; break;
          case 3 : logch = C_FACCHH1_B_3; break;
        }
  if (subch=='1'B && cell == C_CellC)
      switch (inst)
        {
          case 1 : logch = C_FACCHH1_C_1; break;
          case 2 : logch = C_FACCHH1_C_2; break;
```

```
            case 3 : logch = C_FACCHH1_C_3; break;
          }
  return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OC_SubchOfSacchh(subch: B_1; cell:CellID; inst:INTEGER)
Result Type:	LOGICCH
Comments:	subch is of type BITSTRING[1]

Description

OC_SubchOfSacchh operation returns a logic channel identifier for SACCHH subchannel indicated by the input parameters `subch`, `cell` and `inst`, where 'subch' is the subchannel number, `cell` is the cell identifier which the channel belongs to, `inst` is the instance of the channel.

for example:

OC_SubchOfSacchh('0'B, C_CellA, 1) = C_SACCHH0_A_1;
OC_SubchOfSacchh('1'B, C_CellA, 1) = C_SACCHH1_A_1;
OC_SubchOfSacchh('0'B, C_CellB, 1) = C_SACCHH0_B_1;
OC_SubchOfSacchh('1'B, C_CellB, 1) = C_SACCHH1_B_1.

pseudo C code definition for the operation as following :

```
OC_SubchOfSacchh(subch, cell, inst)
BITSTRING   subch;
CellID     cell;
INTEGER   inst;
{
   LOGICCH    logch;
   if (subch=='0'B && cell == C_CellA)
      switch (inst)
        {
          case 1 : logch = C_SACCHH0_A_1; break;
          case 2 : logch = C_SACCHH0_A_2; break;
          case 3 : logch = C_SACCHH0_A_3; break;
        }
  if (subch=='0'B && cell == C_CellB)
      switch (inst)
        {
          case 1 : logch = C_SACCHH0_B_1; break;
          case 2 : logch = C_SACCHH0_B_2; break;
          case 3 : logch = C_SACCHH0_B_3; break;
        }
  if (subch=='0'B && cell == C_CellC)
      switch (inst)
        {
          case 1 : logch = C_SACCHH0_C_1; break;
          case 2 : logch = C_SACCHH0_C_2; break;
          case 3 : logch = C_SACCHH0_C_3; break;
        }
   if (subch=='1'B && cell == C_CellA)
      switch (inst)
        {
          case 1 : logch = C_SACCHH1_A_1; break;
          case 2 : logch = C_SACCHH1_A_2; break;
          case 3 : logch = C_SACCHH1_A_3; break;
        }
  if (subch=='1'B && cell == C_CellB)
      switch (inst)
        {
          case 1 : logch = C_SACCHH1_B_1; break;
          case 2 : logch = C_SACCHH1_B_2; break;
          case 3 : logch = C_SACCHH1_B_3; break;
        }
  if (subch=='1'B && cell == C_CellC)
      switch (inst)
        {
          case 1 : logch = C_SACCHH1_C_1; break;
          case 2 : logch = C_SACCHH1_C_2; break;
```

```
          case 3 : logch = C_SACCHH1_C_3; break;
        }
  return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name: OC_SubchOfSacch4(subch: B_2; cell:CellID)
Result Type: LOGICCH
Comments:

Description

OC_SubchOfSacch4 operation returns a logic channel identifier for SACCHC4 subchannel indicated by the input parameters `subch ` and `cell`, where `subch` is the TDMA offset, `cell` is the cell identifier which the channel belongs to.

for example:

OC_SubchOfSacch4('00'B, C_CellA) = C_SACCHC40_A;
OC_SubchOfSacch4('01'B, C_CellA) = C_SACCHC41_A;
OC_SubchOfSacch4('10'B, C_CellA) = C_SACCHC42_A;
OC_SubchOfSacch4('11'B, C_CellA) = C_SACCHC43_A;
OC_SubchOfSacch4('00'B, C_CellB) = C_SACCHC40_B;
OC_SubchOfSacch4('01'B, C_CellB) = C_SACCHC41_B.

pseudo C code definition for the operation as following :

```
OC_SubchOfSacch4(subch, cell)
BITSTRING   subch;
CellID     cell;
{
   LOGICCH    logch;
   if (cell == C_CellA)
      switch (subch)
        {
          case '00'B : logch = C_SACCHC40_A; break;
          case '01'B : logch = C_SACCHC41_A; break;
          case '10'B : logch = C_SACCHC42_A; break;
          case '11'B : logch = C_SACCHC43_A; break;
        }
   if (cell == C_CellB)
      switch (subch)
        {
          case '00'B : logch = C_SACCHC40_B; break;
          case '01'B : logch = C_SACCHC41_B; break;
          case '10'B : logch = C_SACCHC42_B; break;
          case '11'B : logch = C_SACCHC43_B; break;
        }
   if (cell == C_CellC)
      switch (subch)
        {
          case '00'B : logch = C_SACCHC40_C; break;
          case '01'B : logch = C_SACCHC41_C; break;
          case '10'B : logch = C_SACCHC42_C; break;
          case '11'B : logch = C_SACCHC43_C; break;
        }
  return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OC_SubchOfSacch8(subch: B_3; cell:CellID; inst:INTEGER)
Result Type:	LOGICCH
Comments:	offset is of type BITSTRING[3]

Description

OC_SubchOfSacch8 operation returns a logic channel identifier for SACCHC8 subchannel indicated by the input parameters `subch`, `cell` and `inst`, where `subch` is the TDMA offset, `cell` is the cell identifier which the channel belongs to, `inst` is the instance of the channel.

for example:

OC_SubchOfSacch8('000'B, C_CellA, 1) = C_SACCHC80_A_1;
OC_SubchOfSacch8('001'B, C_CellA, 1) = C_SACCHC81_A_1;
OC_SubchOfSacch8('010'B, C_CellA, 1) = C_SACCHC82_A_1;
OC_SubchOfSacch8('011'B, C_CellA, 1) = C_SACCHC83_A_1;
OC_SubchOfSacch8('100'B, C_CellA, 1) = C_SACCHC84_A_1;
OC_SubchOfSacch8('101'B, C_CellA, 1) = C_SACCHC85_A_1;
OC_SubchOfSacch8('110'B, C_CellA, 1) = C_SACCHC86_A_1;
OC_SubchOfSacch8('111'B, C_CellA, 1) = C_SACCHC87_A_1;
OC_SubchOfSacch8('000'B, C_CellB, 1) = C_SACCHC80_B_1;
OC_SubchOfSacch8('001'B, C_CellB, 1) = C_SACCHC81_B_1;
OC_SubchOfSacch8('000'B, C_CellC, 2) = C_SACCHC80_C_2;
OC_SubchOfSacch8('001'B, C_CellC, 2) = C_SACCHC81_C_2.

pseudo C code definition for the operation as following :

```
OC_SubchOfSacch8(subch, cell, inst)
BITSTRING   subch;
CellID     cell;
INTEGER   inst;
{
  LOGICCH    logch;
  if (inst==1 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SACCHC80_A_1; break;
         case '001'B : logch = C_SACCHC81_A_1; break;
         case '010'B : logch = C_SACCHC82_A_1; break;
         case '011'B : logch = C_SACCHC83_A_1; break;
         case '100'B : logch = C_SACCHC84_A_1; break;
         case '101'B : logch = C_SACCHC85_A_1; break;
         case '110'B : logch = C_SACCHC86_A_1; break;
         case '111'B : logch = C_SACCHC87_A_1; break;
       }
  if (inst==2 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SACCHC80_A_2; break;
         case '001'B : logch = C_SACCHC81_A_2; break;
         case '010'B : logch = C_SACCHC82_A_2; break;
         case '011'B : logch = C_SACCHC83_A_2; break;
         case '100'B : logch = C_SACCHC84_A_2; break;
         case '101'B : logch = C_SACCHC85_A_2; break;
         case '110'B : logch = C_SACCHC86_A_2; break;
         case '111'B : logch = C_SACCHC87_A_2; break;
       }
  if (inst==3 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SACCHC80_A_3; break;
         case '001'B : logch = C_SACCHC81_A_3; break;
         case '010'B : logch = C_SACCHC82_A_3; break;
         case '011'B : logch = C_SACCHC83_A_3; break;
         case '100'B : logch = C_SACCHC84_A_3; break;
```

```
        case '101'B : logch = C_SACCHC85_A_3; break;
        case '110'B : logch = C_SACCHC86_A_3; break;
        case '111'B : logch = C_SACCHC87_A_3; break;
       }
if (inst==1 && cell == C_CellB)
   switch (subch)
     {
      case '000'B : logch = C_SACCHC80_B_1; break;
      case '001'B : logch = C_SACCHC81_B_1; break;
      case '010'B : logch = C_SACCHC82_B_1; break;
      case '011'B : logch = C_SACCHC83_B_1; break;
      case '100'B : logch = C_SACCHC84_B_1; break;
      case '101'B : logch = C_SACCHC85_B_1; break;
      case '110'B : logch = C_SACCHC86_B_1; break;
      case '111'B : logch = C_SACCHC87_B_1; break;
     }
if (inst==2 && cell == C_CellB)
   switch (subch)
     {
      case '000'B : logch = C_SACCHC80_B_2; break;
      case '001'B : logch = C_SACCHC81_B_2; break;
      case '010'B : logch = C_SACCHC82_B_2; break;
      case '011'B : logch = C_SACCHC83_B_2; break;
      case '100'B : logch = C_SACCHC84_B_2; break;
      case '101'B : logch = C_SACCHC85_B_2; break;
      case '110'B : logch = C_SACCHC86_B_2; break;
      case '111'B : logch = C_SACCHC87_B_2; break;
     }
if (inst==3 && cell == C_CellB)
   switch (subch)
     {
      case '000'B : logch = C_SACCHC80_B_3; break;
      case '001'B : logch = C_SACCHC81_B_3; break;
      case '010'B : logch = C_SACCHC82_B_3; break;
      case '011'B : logch = C_SACCHC83_B_3; break;
      case '100'B : logch = C_SACCHC84_B_3; break;
      case '101'B : logch = C_SACCHC85_B_3; break;
      case '110'B : logch = C_SACCHC86_B_3; break;
      case '111'B : logch = C_SACCHC87_B_3; break;
     }
if (inst==1 && cell == C_CellC)
   switch (subch)
     {
      case '000'B : logch = C_SACCHC80_C_1; break;
      case '001'B : logch = C_SACCHC81_C_1; break;
      case '010'B : logch = C_SACCHC82_C_1; break;
      case '011'B : logch = C_SACCHC83_C_1; break;
      case '100'B : logch = C_SACCHC84_C_1; break;
      case '101'B : logch = C_SACCHC85_C_1; break;
      case '110'B : logch = C_SACCHC86_C_1; break;
      case '111'B : logch = C_SACCHC87_C_1; break;
     }
if (inst==2 && cell == C_CellC)
   switch (subch)
     {
      case '000'B : logch = C_SACCHC80_C_2; break;
      case '001'B : logch = C_SACCHC81_C_2; break;
      case '010'B : logch = C_SACCHC82_C_2; break;
      case '011'B : logch = C_SACCHC83_C_2; break;
      case '100'B : logch = C_SACCHC84_C_2; break;
      case '101'B : logch = C_SACCHC85_C_2; break;
      case '110'B : logch = C_SACCHC86_C_2; break;
      case '111'B : logch = C_SACCHC87_C_2; break;
     }
if (inst==3 && cell == C_CellC)
```

```
      switch (subch)
        {
          case '000'B : logch = C_SACCHC80_C_3; break;
          case '001'B : logch = C_SACCHC81_C_3; break;
          case '010'B : logch = C_SACCHC82_C_3; break;
          case '011'B : logch = C_SACCHC83_C_3; break;
          case '100'B : logch = C_SACCHC84_C_3; break;
          case '101'B : logch = C_SACCHC85_C_3; break;
          case '110'B : logch = C_SACCHC86_C_3; break;
          case '111'B : logch = C_SACCHC87_C_3; break;
        }
return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name: OC_SubchOfSdcch4(subch: B_2; cell:CellID)
Result Type: LOGICCH
Comments:

Description

OC_SubchOfSdcch4 operation returns a logic channel identifier for SDCCH4 subchannel indicated by the input parameters `subch ` and `cell`, where `subch` is the TDMA offset, `cell` is the cell identifier which the channel belongs to.

for example:

OC_SubchOfSdcch4('00'B, C_CellA) = C_SDCCH40_A;
OC_SubchOfSdcch4('01'B, C_CellA) = C_SDCCH41_A;
OC_SubchOfSdcch4('10'B, C_CellA) = C_SDCCH42_A;
OC_SubchOfSdcch4('11'B, C_CellA) = C_SDCCH43_A;
OC_SubchOfSdcch4('00'B, C_CellB) = C_SDCCH40_B;
OC_SubchOfSdcch4('01'B, C_CellB) = C_SDCCH41_B.

pseudo C code definition for the operation as following :

```
OC_SubchOfSdcch4(subch, cell)
BITSTRING   subch;
CellID     cell;
{
   LOGICCH    logch;
   if (cell == C_CellA)
      switch (subch)
         {
          case '00'B : logch = C_SDCCH40_A; break;
          case '01'B : logch = C_SDCCH41_A; break;
          case '10'B : logch = C_SDCCH42_A; break;
          case '11'B : logch = C_SDCCH43_A; break;
         }
   if (cell == C_CellB)
      switch (subch)
         {
          case '00'B : logch = C_SDCCH40_B; break;
          case '01'B : logch = C_SDCCH41_B; break;
          case '10'B : logch = C_SDCCH42_B; break;
          case '11'B : logch = C_SDCCH43_B; break;
         }
   if (cell == C_CellC)
      switch (subch)
         {
          case '00'B : logch = C_SDCCH40_C; break;
          case '01'B : logch = C_SDCCH41_C; break;
          case '10'B : logch = C_SDCCH42_C; break;
          case '11'B : logch = C_SDCCH43_C; break;
         }
  return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OC_SubchOfSdcch8(subch: B_3; cell:CellID; inst:INTEGER)
Result Type:	LOGICCH
Comments:	offset is of type BITSTRING[3]

Description

OC_SubchOfSdcch8 operation returns a logic channel identifier for SDCCH8 subchannel indicated by the input parameters `subch`, `cell` and `inst`, where `subch` is the TDMA offset, `cell` is the cell identifier which the channel belongs to, `inst` is the instance of the channel.

for example:

OC_SubchOfSdcch8('000'B, C_CellA, 1) = C_SDCCH80_A_1;
OC_SubchOfSdcch8('001'B, C_CellA, 1) = C_SDCCH81_A_1;
OC_SubchOfSdcch8('010'B, C_CellA, 1) = C_SDCCH82_A_1;
OC_SubchOfSdcch8('011'B, C_CellA, 1) = C_SDCCH83_A_1;
OC_SubchOfSdcch8('100'B, C_CellA, 1) = C_SDCCH84_A_1;
OC_SubchOfSdcch8('101'B, C_CellA, 1) = C_SDCCH85_A_1;
OC_SubchOfSdcch8('110'B, C_CellA, 1) = C_SDCCH86_A_1;
OC_SubchOfSdcch8('111'B, C_CellA, 1) = C_SDCCH87_A_1;
OC_SubchOfSdcch8('000'B, C_CellB, 1) = C_SDCCH80_B_1;
OC_SubchOfSdcch8('001'B, C_CellB, 1) = C_SDCCH81_B_1;
OC_SubchOfSdcch8('000'B, C_CellC, 2) = C_SDCCH80_C_2;
OC_SubchOfSdcch8('001'B, C_CellC, 2) = C_SDCCH81_C_2.

pseudo C code definition for the operation as following :

```
OC_SubchOfSdcch8(subch, cell, inst)
BITSTRING   subch;
CellID     cell;
INTEGER   inst;
{
  LOGICCH    logch;
  if (inst==1 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SDCCH80_A_1; break;
         case '001'B : logch = C_SDCCH81_A_1; break;
         case '010'B : logch = C_SDCCH82_A_1; break;
         case '011'B : logch = C_SDCCH83_A_1; break;
         case '100'B : logch = C_SDCCH84_A_1; break;
         case '101'B : logch = C_SDCCH85_A_1; break;
         case '110'B : logch = C_SDCCH86_A_1; break;
         case '111'B : logch = C_SDCCH87_A_1; break;
       }
  if (inst==2 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SDCCH80_A_2; break;
         case '001'B : logch = C_SDCCH81_A_2; break;
         case '010'B : logch = C_SDCCH82_A_2; break;
         case '011'B : logch = C_SDCCH83_A_2; break;
         case '100'B : logch = C_SDCCH84_A_2; break;
         case '101'B : logch = C_SDCCH85_A_2; break;
         case '110'B : logch = C_SDCCH86_A_2; break;
         case '111'B : logch = C_SDCCH87_A_2; break;
       }
  if (inst==3 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SDCCH80_A_3; break;
         case '001'B : logch = C_SDCCH81_A_3; break;
         case '010'B : logch = C_SDCCH82_A_3; break;
         case '011'B : logch = C_SDCCH83_A_3; break;
         case '100'B : logch = C_SDCCH84_A_3; break;
```

```
        case '101'B : logch = C_SDCCH85_A_3; break;
        case '110'B : logch = C_SDCCH86_A_3; break;
        case '111'B : logch = C_SDCCH87_A_3; break;
       }
if (inst==1 && cell == C_CellB)
   switch (subch)
     {
      case '000'B : logch = C_SDCCH80_B_1; break;
      case '001'B : logch = C_SDCCH81_B_1; break;
      case '010'B : logch = C_SDCCH82_B_1; break;
      case '011'B : logch = C_SDCCH83_B_1; break;
      case '100'B : logch = C_SDCCH84_B_1; break;
      case '101'B : logch = C_SDCCH85_B_1; break;
      case '110'B : logch = C_SDCCH86_B_1; break;
      case '111'B : logch = C_SDCCH87_B_1; break;
     }
if (inst==2 && cell == C_CellB)
   switch (subch)
     {
      case '000'B : logch = C_SDCCH80_B_2; break;
      case '001'B : logch = C_SDCCH81_B_2; break;
      case '010'B : logch = C_SDCCH82_B_2; break;
      case '011'B : logch = C_SDCCH83_B_2; break;
      case '100'B : logch = C_SDCCH84_B_2; break;
      case '101'B : logch = C_SDCCH85_B_2; break;
      case '110'B : logch = C_SDCCH86_B_2; break;
      case '111'B : logch = C_SDCCH87_B_2; break;
     }
if (inst==3 && cell == C_CellB)
   switch (subch)
     {
      case '000'B : logch = C_SDCCH80_B_3; break;
      case '001'B : logch = C_SDCCH81_B_3; break;
      case '010'B : logch = C_SDCCH82_B_3; break;
      case '011'B : logch = C_SDCCH83_B_3; break;
      case '100'B : logch = C_SDCCH84_B_3; break;
      case '101'B : logch = C_SDCCH85_B_3; break;
      case '110'B : logch = C_SDCCH86_B_3; break;
      case '111'B : logch = C_SDCCH87_B_3; break;
     }
if (inst==1 && cell == C_CellC)
   switch (subch)
     {
      case '000'B : logch = C_SDCCH80_C_1; break;
      case '001'B : logch = C_SDCCH81_C_1; break;
      case '010'B : logch = C_SDCCH82_C_1; break;
      case '011'B : logch = C_SDCCH83_C_1; break;
      case '100'B : logch = C_SDCCH84_C_1; break;
      case '101'B : logch = C_SDCCH85_C_1; break;
      case '110'B : logch = C_SDCCH86_C_1; break;
      case '111'B : logch = C_SDCCH87_C_1; break;
     }
if (inst==2 && cell == C_CellC)
   switch (subch)
     {
      case '000'B : logch = C_SDCCH80_C_2; break;
      case '001'B : logch = C_SDCCH81_C_2; break;
      case '010'B : logch = C_SDCCH82_C_2; break;
      case '011'B : logch = C_SDCCH83_C_2; break;
      case '100'B : logch = C_SDCCH84_C_2; break;
      case '101'B : logch = C_SDCCH85_C_2; break;
      case '110'B : logch = C_SDCCH86_C_2; break;
      case '111'B : logch = C_SDCCH87_C_2; break;
     }
if (inst==3 && cell == C_CellC)
```

```
    switch (subch)
      {
        case '000'B : logch = C_SDCCH80_C_3; break;
        case '001'B : logch = C_SDCCH81_C_3; break;
        case '010'B : logch = C_SDCCH82_C_3; break;
        case '011'B : logch = C_SDCCH83_C_3; break;
        case '100'B : logch = C_SDCCH84_C_3; break;
        case '101'B : logch = C_SDCCH85_C_3; break;
        case '110'B : logch = C_SDCCH86_C_3; break;
        case '111'B : logch = C_SDCCH87_C_3; break;
      }
return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OM_2Msgs(ch:LOGICCH; pgg:PGG; mode:SENDINGMODE)
Result Type:	BOOLEAN
Comments:	

Description

OM_2Msgs puts the layer 2 emulator into a special operation mode then returns. In this special mode the layer 2 emulator sends the next two consecutive messages in the following way:

- send the first message on the paging subchannel indicated by the `pgg` and 'ch';

- if the `mode` = C_NxtButOne, send the second message in the next but one paging sub block;
- if the `mode` = C_FmrAGB, send the second message in a former access grant block;
- if the `mode` = C_BfReOcc, send the second message before the MS's original paging subchannel re-occurs but later than the next paging block of that CCCH (paging block not belong to the MS);
- if the `mode` = C_NxtButOneNxt, nothing is sent in the next but one paging sub block, then send the second message in the next paging subblock of the MS's paging subchannel.

Detailed Comments:

Test Suite Operation Definition

Operation Name: OM_Assoc(lgch1:LOGICCH; sublgch1:LOGICCH; sublgch2:LOGICCH; sublgch3:LOGICCH; sublgch4:LOGICCH; sublgch5:LOGICCH; sublgch6:LOGICCH; sublgch7: LOGICCH; sublgch8:LOGICCH; lgch2:LOGICCH; sublgch9:LOGICCH; sublgch10:LOGICCH; sublgch11:LOGICCH; sublgch12:LOGICCH; sublgch13:LOGICCH; sublgch14:LOGICCH; sublgch15: LOGICCH; sublgch16:LOGICCH)

Result Type: BOOLEAN

Comments:

Description

OM_Assoc operation associates the sub logic channel identifiers `sublgch1` to `sublgch8` and `sublgch9` to `sublgch16` with the generic "parent" channel identifier ` lgch1` and `lgch2` respectively, therefore the subchannel identifiers can refer to the corresponding channels configured by OM_ChConf operation.

The `lgch1` can be an identifier for SDCCH4, SDCCH8, FACCHH and the `lgch2` can be SACCHC4, SACCHC8, and SACCHH.

"dummy" in the actual parameter list means the corresponding parameter is not used.

Naming convention for logical channel constants:

C_<channel type>[<rate>][<subchannel>][_<cell>][_<instance>]

e.g.: C_FACCH H 0 _C _3 => C_FACCHH0_C_3

channel type: FCCH, SCH, BCCH, CBCH, PCH, RACH, FACCH, SACCH, SDCCH, DCCH

rate: F(ull), H(alf), 4 (fourths, C4 for SACCHC4), 8 (eighths, C8 for SACCHC8).

subchannel: H: 0, 1
(C)4: 0, ..., 3
(C)8: 0, ..., 7

cell: A, ..., H

instance: 1, ..., 3 When several channels of the same type can exist in one cell simultaneously (in different time slots and / or on different frequencies) they are identified by their instance. The parameters of ChConf are used to specify the frequency and slot of this instance of the channel type.

Not all possible combinations of this naming syntax have been used. According to the elements that are used / omitted the name can have a special meaning.

Examples:	Comments:
C_FCCH_A	FCCH of cell A.
C_SCH_A	SCH of cell A.
C_BCCH_A_1	First instance of a BCCH in cell A.
C_CBCH_A	CBCH of cell A.
C_PCH_A_1	First instance of a PCH in cell A (there is one instance of the PCH per BCCH in the cell).
C_AGCH_A_1	First instance of a AGCH in cell A (there is one instance of the AGCH per BCCH in the cell).
C_RACH_A_1	First instance of a RACH in cell A.
C_FACCHF_A_1	First instance of a full rate FACCH in cell A.
C_FACCHH_A_1	First instance of a half rate FACCH in cell A (all subchannels).
C_FACCHH0_A_1	Subchannel 0 of the first instance of a half rate FACCH in cell A.
C_SACCHF_A_1	First instance of a full rate SACCH in cell A.
C_SACCHH_A_1	First instance of a half rate SACCH in cell A (all subchannels).
C_SACCHH1_A_1	Subchannel 1 of the first instance of a half rate SACCH in cell A.
C_SACCHC4_A	SACCH/4 of cell A (all subchannels).
C_SACCHC43_A	Subchannel 3 of the SACCH/4 of cell A.
C_SACCHC8_A_1	First instance of a SACCH/8 in cell A (all subchannels).
C_SACCHC87_A_1	Subchannel 7 of the first instance of a SACCH/8 in cell A.
C_SDCCH4_A	SDCCH/4 of cell A (all subchannels).
C_SDCCH42_A	Subchannel 2 of the SDCCH/4 of cell A.
C_SDCCH87_A_1	First instance of a SDCCH/8 in cell A (all subchannels).
C_SDCCH87_A_1	Subchannel 7 of the first instance of a SDCCH/8 in cell A.

This list covers all types of combinations used in the TTCN for the time being.

When the meaning refers to 'all' subchannels of a channel (eg. C_FACCHH_A_1), it means that all active subchannels of this channel activated previously by OM_ChConf, are associated with a logical channel identifier in the OM_Assoc.

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OM_BmInfo(ch:LOGICCH; mode:B_8)
Result Type:	BOOLEAN
Comments:	mode is of type BITSTRING[8], imode is a part of the channel mode IE

Description

OM_BmInfo operation checks whether the MS starts transmiting Bm channel information according to the new channel mode `mode'. The operation returns TRUE if the result of the check is correct otherwise the TSO returns FALSE.

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OM_ChangeRFOf2Cells(cellid1:CellID; bspwr1:INTEGER; cellid2:CellID; bspwr2:INTEGER)
Result Type:	BOOLEAN
Comments:	

Description

This TSO is used to change the RF level of the cell 'cellid1' to the RF level 'bspwr1' and the RF level of the cell 'cellid2' to the RF level 'bspwr2' [dBuVemf], then the TSO returns TRUE.

The main use of this TSO is to force the Mobile Station(MS) to select a new cell according to the GSM 05.08 and GSM 03.22.

For the MS to select the new cell then the path loss criterion parameter (C1) of the old will have to be less than zero (C1 < 0).
This criterion is defined as:

C1 = (A - max(B, 0))

when

A - Received Level Average - (RXLEV_ACCESS_MIN)
B - MS_TXPWR_MAX_CCH - P {non DCS 1800}
MS_TXPWR_MAX_CCH + POWEROFFSET - P {DCS 1800}

RXLEV_ACCESS_MIN - Minimum received level at the MS required for access to the system.
MS_TXPWR_MAX_CCH - Maximum TX power level an MS may use when accessing the system until otherwise commanded.
POWER OFFSET - The power offset to be used in conjunction with the MS TXPWR MAX CCH
parameter by the class 3 DCS 1 800 MS.
P - Maximum RF output power of the MS.

All values are expressed in dBm.

In most test cases MS_TXPWR_MAX_CHH and the POWER_OFFSET will be set at their minimum values thus C1 = A (B always being negative). Suitable levels of cells will be either 63 dBuV(C_E_default) ,53 dBuV(C_E_neighbourdefault) or 45 dBuV(C_E_suitable). Assuming that the MS uses a RXLEV_ACCESS_MIN between -110 and -115 dbM then to ensure that C1 < 0 the test system's output needs to be set to ensure the level input detected by the MS is less than -123 dBm or lower (-115 plus an 8dB for discrimination between C1 or C2 values and 0, GSM 11.10-1 section 20). This value of -123 dBm corresponds to -10 dBuV (C_E_notsuitable)

Detailed Comments:

Test Suite Operation Definition

Operation Name: OM_ChConf(bspwr: INTEGER; mspwr:INTEGER; acttype:BITSTRING; chmod:CHMOD; ta:TA; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; td:INTEGER; fn:INTEGER; babr, cch_con, bpm:B_3; pgfil:PG1_RQ_PDU; dtxu:BITSTRING; dtxd:BITSTRING; cell:CellID; lgch1:LOGICCH; lgch2:LOGICCH; lgch3:LOGICCH; lgch4:LOGICCH; lgch5:LOGICCH; lgch6:LOGICCH; lgch7:LOGICCH; lgch8:LOGICCH; lgch9:LOGICCH)

Result Type: BOOLEAN

Comments:

Description

The operation sets the configuration of a basic physical channel according to the input parameters and map logic channel combination onto it:

- bspwr: base station power level in dBuv;
- mspwr: mobile station power level in "power control level";
- acttype: type of activation:
  - '000'B activation for intra-cell channel change ---- immediate assignment;
  - '001'B activation for intra-cell channel change ---- assignment;
  - '010'B activation for inter-cell channel change ---- asynchronous handover;
  - '011'B activation for inter-cell channel change ---- synchronous handover;
  - '100'B activation for additional assignment;
    NOTE: the types defined above are all the same, no real difference
  - '101'B activation for receiving only;
- chmod: channel mode;
- ta: timing advance ---- receive timing ta bit periods in advance of the normal receive timing if the channel is a dedicated channel ;
- slot: time slot;
- tsc: Training sequence code for normal bursts;
- rf: ARFCN or hopping parameters;
- chcmbn: logic channel combination mapping to the physical channel;
- td: timing difference between the cell and conceptual timing reference ---- to make 2 cells with different timebase;
- fn: initial frame number offset to the timing base counter ---- to make 2 cells with different frame numbers;
- babr number of blocks reserved for access grant
- cch-con CCCH/SDCCHs configuration
- bpm multiframe period for transmission of PAGING REQUEST
- pgfil: paging filling contents;
- dtxu: mobile station discontinuous transmission:
  - '0'B mobile station discontinuous transmission is not applied;
  - '1'B mobile station discontinuous transmission is applied;
- dtxd: base station discontinuous transmission:
  - '0'B base station discontinuous transmission is not applied;
  - '1'B base station discontinuous transmission is applied;
- cell: cell identifier;
- lgch1: logic channel identifier for the channel FCCH;
- lgch2: logic channel identifier for the channel SCH;
- lgch3: logic channel identifier for the channel BCCH;
- lgch4: logic channel identifier for the channel PCH;
- lgch5: logic channel identifier for the channel AGCH;
- lgch6: logic channel identifier for the channel RACH;
- lgch7: logic channel identifier for the channel FACCH or SDCCH;
- lgch8: logic channel identifier for the channel SACCH;
- lgch9: logic channel identifier for the channel CBCH;

For parameters lgch1 to lgch9 "dummy" in the actual parameter list means that the corresponding parameters are not used.

In the test cases the logic channel identifiers lgch1 to lgch9 are used to refer the logic channels configured by the operation. There are only generic identifiers for SDCCH4 or SDCCH8 or FACCHH channel and SACCHC4 or SACCHC8 or SACCHH, identifiers for subchannels of SDCCH4 or SDCCH8 or FACCHH and SACCHC4 or SACCHC8 or SACCHH are linked to the generic identifiers by OM_Assoc operation and in turn refer to the sub logic channels configured by the operation.

Channel re-use whithiin OM_ChConf:

When an OM_ChConf calls is made then the following rules can be applied for deciding whether or not to re-use existing physical channel.

Decode the 'chcmbn' (channel combination) parameter to determine the new channel type.

If new channel = "C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4" (combined) or

if new channel = "C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4" (combined + SMS) then re-use any existing control channel configuration with same CellID.

If new channel = "C_FCCH_SCH_BCCH_CCCH" (non-combined) then use any combined, combined + SMS or non-combined control channel configuration with the same CellID.

If new channel = "C_BCCH_CCCH_2", "C_BCCH_CCCH_3" or "C_BCCH_CCCH_4" (ccch-group) then re-use any existing ccch-group control channel configuration with the same CellID and the slot, or re-use any existing combined, combined + SMS with the same CellID.

If new channel = "C_TCHF_ACCHF_1", "C_TCHF_ACCHF_2", "C_TCHH_ACCHH_1", "C_TCHH_ACCHH_2" (traffic) or "C_SDCCH8_SACCHC8_1", "C_SDCCH8_SACCHC8_2", "C_SDCCH8_SACCHC8_3" (dedicated) then re-use any existing traffic or dedicated channel with the same channel description.

Detailed Comments:

Test Suite Operation Definition

Operation Name: OM_ChMdModi(ch:LOGICCH; chmod: CHMOD)

Result Type: BOOLEAN

Comments:

Description

OM_ChMdModi operation passes channel mode IE for channel `ch` to lower layer emulator and requests the emulator to set the `ch` to the mode `chmod`. After the mode changed, it returns.

Detailed Comments:

Test Suite Operation Definition

Operation Name: OM_CphMdChg(ch:LOGICCH; cphmod: CPHMS; key:BITSTRING)

Result Type: BOOLEAN

Comments:

Description

OM_CphMdChg operation passes ciphering parameters for channel `ch` to lower layer and sets the lower layer emulator into a special operation mode in which lower layer starts the 3-step ciphering mode setting sequence when the next L3 message on the channel `ch` arrives.

Detailed Comments:

Test Suite Operation Definition

Operation Name: OM_CphMd(ch:LOGICCH; cphmod: CPHMS; key:BITSTRING)

Result Type: BOOLEAN

Comments:

Description

OM_CphMd operation passes ciphering parameters for channel `ch` to lower layer and sets the channel `ch` in the ciphering mode `cphmod`. This operation puts the channel `ch` into specified ciphering mode `cphmod` immediately without the 3-step ciphering mode setting sequence.

Detailed Comments:

Test Suite Operation Definition

Operation Name: OM_ComingFn(ch:LOGICCH)

Result Type: FN

Comments:

Description

OM_ComingFn operation returns the frame number (FN modulo 42432) which is about 5 seconds later than current frame number and is able to carry L3 message on the channel `ch`.

Detailed Comments: The delay of about 5 seconds ensures that there is still enough time left for higher layer controller to prepare next TTCN send event after the OM_ComingFn returns. the exact delay value is up to the implementor.

Test Suite Operation Definition	
Operation Name:	OM_Deactivate(ch1, ch2:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
OM_Deactivate operation deactivates the logical channels `ch1` and 'ch2', then returns TRUE. The ch1 can be a DCCH or a TCH, the ch2 is the associated SACCH. Deactivation of an active SDCCH channel can be done by stopping transmission; this means that for the bursts on the BCCH carriers, dummy bursts shall be transmitted, for other bursts nothing shall be transmitted. Alternatively the deactivation of a SDCCH can be done by transmitting bad blocks on the SACCH with the parity bits set incorrectly. This will lead to a radio link failure in the MS. Deactivation of a TCH channel can be done by stopping transmission if it is non-BCCH. Deactivation of a SACCH_T can be done by transmitting bad blocks on the SACCH_T with the parity bits set incorrectly. The wrong channel coding on the SACCH will lead to a radio link failure in the MS after several seconds depending on the system information messages..	
Detailed Comments:	The TSO is called in TC_26_7_3_2, TC_26_7_4_3_2, TC_26_7_4_3_3, TC_26_7_4_3_4 on a SDCCH/4/SACCH/C4 channel in downlink, TC_26_2_4_1, TC_26_2_4_2, TC_26_8_2_1, TC_26_8_2_2, TC_26_8_2_3 on TCH/SACCH_T channel in both directions.

Test Suite Operation Definition	
Operation Name:	OM_FHCHK(ch:LOGICCH; ca:CCHD; ma:MA; cd:CHD; t :INTEGER; fn:FN)
Result Type:	BOOLEAN
Comments:	
Description	
After the SS has been configured to start / change frequency hopping at a particular start time, this TSO instructs the SS to check whether the MS correctly uses the hopping frequencies on and after the expected start time according to the frequency parameters. If `t` = 0 or 32024 <= `t` <= 42431 the operation checks that the new frequencies starts "without delay". If 0 < `t` <= 31000 the operation checks that the new frequencies starts "on" the frame number of (X + t) mod 42432, where x = 51 * ((fn.t3 - fn.t2) MOD 26) + fn.t3 + 1326 * fn.t1. If 31000 < `t` < 32024, or `t` < 0 or `t` > 42431 then the test has been configured incorrectly and should be re-run with a permitted value for `t`. This check is performed at the uplink RF burst level by sampling for a period of 3 seconds from the start time and should therefore be called immediately after the SS has been configured to start / change hopping. The operation returns TRUE if the MS behaviour is correct, otherwise it returns FALSE. Note: before this TSO is called, a layer 3 message must be sent to the MS to command the MS to start / change hopping using the same frequency hopping parameters.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_FreeResource(ch :LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
OM_FreeResource operation instructs the lower layer emulator to release the resources which have been allocated to the logical channel "ch". After the resources are released the operation returns TRUE.	
Detailed Comments:	In some test cases (most of them are hanhover test cases) diffirent channels are needed while the test goes on. the purpose of the operation is to reuse lower layer resources (e.g. tranceivers), which have been allocated to the previous channel, for configing another channel required in the next testing step.

Test Suite Operation Definition	
Operation Name:	OM_FreqDef(st:STRT; ma:MA; ch :LOGICCH; chd: CHD; ca:CCHD; frmn: FN)
Result Type:	BOOLEAN
Comments:	
Description	
OM_FreqDef operation passes the frequency hopping parameters for channel `ch` to lower layer emulator. The emulator starts using these frequency parameters at the frame number indicated by starting time `st`. The parameter 'frmn' indicates a reference frame number at which the first burst of the message, FREQUENCY REDEFINITION, ASSIGNMENT COMMAND or HANDOVER COMMAND, containing starting time is sent in the concerned cell. If 0 <= (st - frmn) mod 42432 <= 31623, the indicated time is the next time when frame number mod 42432 is equal to st. If 32024 <= (st - frmn) mod 42432 <= 42431, the indicated time has already elapsed. After completely sending out the message containing the parameter starting time the tester starts frequency hopping.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_GetHoaccPara(ch:LOGICCH)
Result Type:	HOACC_PARA
Comments:	
Description	
OM_GetHoaccPara instructs the lower layer emulator to return the total number of the handover access bursts received on channel `ch`, the observed minimum and maximum power levels in dBm modulo 256 and the observed minimum and maximum timing advance in bits modulo 256.The lower layer emulator starts the observation of handover access bursts as soon as sending out a HANDOVER COMMAND message. The TSO returns an accumulative result of the observation when it is called.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_GetL1Hd(ch:LOGICCH)
Result Type:	L1HD
Comments:	
Description	
OM_GetL1Hd instructs the L 2 emulator to read the L 1 header of the next SACCH frame received on channel `ch` and return this value.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_LowerLayerFail(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
This operation is to force the lower layer failure, then it returns. The lower layer failure can be any one of layer 1 failures or layer 2 failures. - layer 1 failure, see GSM 05.08; - layer 2 failure, see GSM 04.06.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_NoL2Ack(i:INTEGER; ch: LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
When the parameter i>0 the layer 2 emulator is put into a special operation mode then the TSO returns. In this special mode the layer 2 emulator shall not aknowledge the 'i'th occurence of the L2 I frame which has the more data bit "M" = 0 on the channel 'ch'. The layer 2 emulator resumes normal operation when - either the TSO is called again with i = 0 on the same 'ch', - or a data link SABM message is received on the 'ch'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_NotAckSetup(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
This operation puts the layer 2 emulator into a special operation mode then returns. In this special mode the layer 2 emulator will not acknowledge the last L 2 frame which carries the SETUP message, and the layer 2 emulator resumes normal operation after this L 2 frame.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_NoUAforDISC(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
OM_NoUAforDISC and OM_ResumUAforDISC are a pair of TSOs. OM_NoUAforDISC forces the L2 emulator of the tester to enter a specific mode. OM_ResumUAforDISC resumes the L2 emulator to a normal L2 operational mode. In the specific mode on receiving a DISC on the channel 'ch', the L2 emulator - does not respond with a UA, - remains in the multiple-frame established state, - indicates that the DISC has been received (by means of subsequent DL_RELEASE_INDICATION ASPs). Then the TSO returns TRUE.	
Detailed Comments:	The TSO is called in TC_26_6_12_2 and TC_26_6_12_4.

Test Suite Operation Definition	
Operation Name:	OM_PgFill(cellid:CellID; msg:PG1_RQ_PDU)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
OM_PgFill operation sets up the contents of paging filling frame to be sent on all paging subchannels continuously. If the contents of paging filling frame in cell `cellid` have not been set up, the operation sets the contents to `msg`. If paging filling frame in cell `cellid` has been set up, the operation changes the paging filling message to `msg`. After the required action has been taken it returns.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_Reactivate(ch1, ch2:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
The TSO OM_Reactivate reactivates the logical channels `ch1` and 'ch2' which were previously deactivated by the OM_Deactivate. Then the TSO returns TRUE. The ch1 is an SDCCH, the ch2 is the associated SACCH. Reactivation of deactivated SDCCH/SACCH channels means to restart normal transmissions both on the SDCCH and the associated SACCH as specified in the ETS 300 574 (GSM 05.02).	
Detailed Comments:	The TSO is called in TC_26_7_3_2, TC_26_7_4_3_2, TC_26_7_4_3_3, TC_26_7_4_3_4 on the SDCCH/4 / SACCH/C4 channels.

Test Suite Operation Definition	
Operation Name:	OM_ReturnFn(ch:LOGICCH)
Result Type:	FN
Comments:	
Description	
OM_ReturnFn operation returns the frame number (FN modulo 42432) on which the last L3 message was sent on logic channel `ch`.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_ResumUAforDISC(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
OM_NoUAforDISC and OM_ResumUAforDISC are a pair of TSOs. OM_NoUAforDISC forces the L2 emulator to enter a specific mode. OM_ResumUAforDISC resumes the L2 emulator to a normal L2 operational mode. OM_ResumeUAforDISC cancels the effect of OM_NoUAforDISC, returning L2 on the channel 'ch' to normal operation in the multiple-frame established state. Then the TSO returns TRUE.	
Detailed Comments:	The TSO is called in TC_26_6_12_2 and TC_26_6_12_4.

Test Suite Operation Definition	
Operation Name:	OM_SendSMSCBWhilePaging(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The TSO makes the lower layers send the second SMSCB message at the same time as the MS is paged. This shall be achieved by paging the MS immediately after the first block of the CB message has been sent. The SS shall ensure that the page is transmitted on the radio interface prior to the transmission of the 4th block of the CB message.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_SendNextOn(ch:LOGICCH; fn:FN)
Result Type:	BOOLEAN
Comments:	
Description	
OM_SendNextOn operation sets the lower layer emulator into a special operation mode then returns. In the special mode the lower layer emulator sends the next L3 message on the indicated frame number `fn` on the 'ch' channel. After the next L3 message is sent, the lower layer resumes normal operation automatically.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_SendNextOnHO(ch:LOGICCH; cnt:INTEGER)
Result Type:	BOOLEAN
Comments:	Used in non-synchronised handover tests
Description	
OM_SendNextOnHO operation sets the lower layer emulator into a special operation mode then returns. In the special mode the lower layer emulator waits for the number of Handover bursts specified by the parameter 'cnt' on channel 'ch' before sending the next L3 message. After the message is sent, the lower layer resumes normal operation automatically.The TSO does not affect any TTCN receiving event.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_StartMsrReport(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The operation starts the reporting of received MEASUREMENT REPORT message to layer 3 emulator, then it returns with value TRUE. Measurement Reports shall be disable by default. By default means that the tester filters them out from the TTCN in-buffer. This operation enables the reports entering the TTCN in-buffer. The OM_StopMsrReport operation disables the reports entering the TTCN in-buffer. When starting each test case, the measurement reports are filtered out by default.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_StopAllBCCH(cell1, cell2, cell3, cell4, cell5, cell6, cell7, cell8:CellID)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The TSO stops the RF transmission on all BCCH channels including dummy bursts in the indicated cells, but keeps the uplink CCCH reception active. After all transmission have been stopped the TSO returns TRUE. If a cell in the parameter list is not in use a "dummy" value needs to be assigned to it by the calling test case. The TSO is called in TC_26_3_3.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_StopCell(cell:CellID)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The TSO stops completely all RF transmission of the specific cell `cell`, including the dummy bursts on other slots. After all RF transmission have been stopped the TSO returns TRUE. The cell shall be deleted from the lower layer tester and therefore if the cell 'cell' is required again by the calling test case it needs to be re-configured. The TSO is called in TC_26_3_4, TC_26_6_6_1, TC_26_7_4_2_2_1, TC_26_7_4_2_2_2, TC_26_7_4_6, TC_31_6_2_3.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_StopMsrReport(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The operation stops the reporting of received MEASUREMENT REPORT message to layer 3 emulator, after measurement report stops it returns with value TRUE. Measurement Reports shall be disable by default. By default means that the tester filters them out from the TTCN in-buffer. This operation disables the reports entering the TTCN in-buffer. The OM_StartMsrReport operation enables the reports entering the TTCN in-buffer. When starting each test case, the measurement reports are filtered out by default.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ACMIncCHK(para:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
the OO_ACMIncCHK operation requests (e.g. from the control console of the test system) the test operator to read the value of ACM on SIM and check the increment of the value. Then the operator informs the test system whether the increment is as description `para`, if it is as the description the operation returns TRUE, otherwise returns FALSE. The value of ACM can be read either via MMI or by removing the SIM and using SIM reader.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ACMReading
Result Type:	BOOLEAN
Comments:	
Description	
the OO_ACMReading operation requests (e.g. from the control console of the test system) the test operator to read and note the value of ACM on SIM. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE. The value of ACM can be read either via MMI or by removing the SIM and using SIM reader.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ACMSetting
Result Type:	BOOLEAN
Comments:	
Description	
the OO_ACMSetting operation requests (e.g. from the control console of the test system) the test operator to reset the ACM to zero and set the ACMmax to 2 units. After the operator finishes the action and informs the test system it returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_AddPwrAmp
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to add power amplification at the Mobile Station under test. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_AltIndCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the alerting indication is generated. If the operator informs (e.g. from the control console of the test system) the test system that the alerting indication is generated, the operation returns TRUE. If the operator informs the test system that the alerting indication is not generated, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CalledNumCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the called party number is correctly displayed on the mobile station. If the operator informs (e.g. from the control console of the test system) the test system that the display is correct, the operation returns TRUE. If the operator informs the test system that the display is wrong, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CalledPtyNumCHK(num:CDPN)
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the called party number `num` is the one that was entered into the MS. If the operator informs (e.g. from the control console of the test system) the test system that the number is correct, the operation returns TRUE. If the operator informs the test system that the number is wrong, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckAllSMPresentBut4th
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the mobile station under test has stored all short messages during the test case but not the 4th one which should have been replaced by the 5th(cf. GSM 11.10, clause 34.2.7.4). The messages are displayed. Then the operation returns. TRUE: All SM present but not the 4th FALSE: 4th SM present or 1st, 2nd, 3rd or 5th missing.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckCBSMReceived(NumbOfIA5chara: INTEGER; msg:SMSCBpack)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check whether the MS under test has correctly received the cell broadcast short message represented by 'msg' . Then the TSO returns TRUE: if the CBSM has been correctly received, FALSE: if non or an incorrect CBSM has been received. The TSO needs to convert the 'msg' into IA5 string according to the packing character rules specified in the section 6.1.2 of ETS 300 628 (GSM 03.38). The 'NumbOfIA5chara' indicates the number of IA5 characters in the 'msg'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckMCEFOnSIM
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the Memory Capacity Exceeded Flag has been set on the SIM simulator and to inform the test system of the result of the checking. If the checking succeds the operation returns TRUE, FALSE otherwise.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckMCEFOnSIMUnset
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the Memory Capacity Exceeded Flag has been unset on the SIM simulator and to inform the test system of the result of the checking. If the checking succeds the operation returns TRUE, FALSE otherwise.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckMessageDisplayed(NumbOfIA5chara:INTEGER; msg:OCTETSTRING)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the mobile station under test indicates that an SM has arrived. If the MS provides the functionality to display MT messages, it is checked that the message contents represented by 'msg' is correctly displayed. Then the operation returns TRUE: if the MS indicates SM reception (and displays correct message), FALSE: if the MS does not indicate SM reception (or displays incorrect message). The TSO needs to convert the 'msg' into IA5 string according to the packing character rules specified in the section 6.1.2 of ETS 300 628 (GSM 03.38). The 'NumbOfIA5chara' indicates the number of IA5 characters in the 'msg'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckUssdStringDisplayed(strg: IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the mobile station under test displays the Ussd String 'strg'. Then the operation returns. TRUE: MS displays the correct Ussd String. FALSE: MS does not display the correct Ussd String.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ConnectSIMSimulator
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to connect the SIM simulator to the mobile station under test. For the contents of the SIM simulator the operator refers to initial conditions of the test case in GSM 11.10-1. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_DepressEndKey
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to depress the 'END' key of the Mobile Stations keypad, then returns.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_DialCalledNum
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to dial the called party number at the Mobile Station under test but not press the "SEND" key (i.e. not initiate the call setup). After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_DisplaySMAndSendReplySM(n, NumbOfIA5chara: INTEGER; msg:OCTETSTRING)
Result Type:	OCTETSTRING
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to display the 'n'-th short message received in the arriving order and to check whether the short message contents represented by 'msg' is correctly displayed. Then the TSO asks the operator to send a reply short message from the mobile. After the operator finishes the action, the operation returns with the message contents sent in OCTETSTRING. The TSO needs to convert the 'msg' into IA5 string and the IA5 text entered by the operator into OCTETSTRING according to the packing character rules specified in the section 6.1.2 of ETS 300 628 (GSM 03.38). The 'NumbOfIA5chara' indicates the number of IA5 characters in the 'msg'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_DTMFIndCHK(character:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the `character` is indicated by the DTMF indicator of the MS. If the operator informs (e.g. from the control console of the test system) the test system that the MS does not give the indication of `character`, the operation returns FALSE. If the operator informs the test system that the MS gives the indication, the operation returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_EmptyMessageStorage
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to empty the message storage of the mobile station under test. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_EnterPswd(pswd:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This TSO displays a message on the control console to inform the operator that he is to enter a given password (the actual password is given by 'pswd') on the mobile. The operation waits for the operator to acknowledge the request, which may be done before or after entering the password into the mobile, and then returns the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_HookOff
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to answer the mobile terminating call at the Mobile Station under test. This operation waits for the operator to acknowledge the request which may be done before or after the call has been answered and then returns the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_Telephony
Result Type:	BOOLEAN
Comments:	
Description	
This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or configuration of the MS in such a way that the MS is able to successfully receive the call for telephony service. If the MS supports both telephony without immediate connection mode and telephony with immediate connection mode, set the MS in telephony without immediate connection mode. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE. This TSO is used in MO call test cases except TC_11_2, where this TSO is call for single-numbering scheme MT testing.	
Detailed Comments:	used for TS11 and TS12

Test Suite Operation Definition

Operation Name: OO_IFsetup_BS2xorBS3x_MO(srv:SERVICES; itc, sacp:B_3; ce:B_2; rate:IA5String)

Result Type: BOOLEAN

Comments:

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS, when trigged, is able to initiate the call for the BS2x or BS3x service `srv` with the specified itc, sacp, ce values. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

This TSO is used in MO call test cases except TC_11_2, where this TSO is call for single-numbering scheme MT testing.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT paramters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:

1) srv ----- basic service

- C_Async300 Basic Service 21, Asynchronous 300 bps
- C_Async1200 Basic Service 22, Asynchronous 1.2 kbps
- C_Async120075 Basic Service 23, Asynchronous 1200/75 bps
- C_Async2400 Basic Service 24, Asynchronous 2.4 kbps
- C_Async4800 Basic Service 25, Asynchronous 4.8 kbps
- C_Async9600 Basic Service 26, Asynchronous 9.6 kbps
- C_Sync1200 Basic Service 31, Synchronous 1.2 kbps
- C_Sync2400 Basic Service 32, Synchronous 2.4 kbps
- C_Sync4800 Basic Service 33, Synchronous 4.8 kbps
- C_Sync9600 Basic Service 34, Synchronous 9.6 kbps

2) itc ------ information transfer capability

- '001'B unrestricted digital information
- '010'B 3.1 kHz audio, ex PLMN

3) sacp ---- signalling access protocol

- '001'B I.440/450
- '010'B X.21
- '101'B X.28 - non dedicated PAD
- '110'B X.32

4) ce ------ connection element

- '00'B transparent
- '01'B non transparent

5) rate ----- channel rate

- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: Used for BS21, BS22, BS23, BS24, BS25, BS26, BS31, BS32, BS33, BS34

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS4x_MO(srv:SERVICES; ce:B_2; rate:IA5String)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, orthe MS configuration in such a way that the MS, when trigged, is able to initiate the call for the BS4x service `srv` with specified ur and ce values. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:

1) srv ----- basic service

- C_PAD300 Basic Service 41, PAD access 300 bps
- C_PAD1200 Basic Service 42, PAD access 1.2 kbps
- C_PAD120075 Basic Service 43, PAD access 1200/75 bps
- C_PAD2400 Basic Service 44, PAD access 2.4 kbps
- C_PAD4800 Basic Service 45, PAD access 4.8 kbps
- C_PAD9600 Basic Service 46, PAD access 9.6 kbps

2) ce ------ connection element

- '00'B transparent
- '01'B non transparent

3) rate ----- channel rate

- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: Used for BS41, BS42, BS43, BS44, BS45, BS46

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS5x_MO(srv:SERVICES; ur:B_4; rate:IA5String)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, orthe MS configuration in such a way that the MS, when trigged, is able to initiate the call for the BS5x service `srv` with the specifyied ur value. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:

1) srv ----- basic service

- C_Packet2400 Basic Service 51, Packet access 2.4 kbps
- C_Packet4800 Basic Service 52, Packet access 4.8 kbps
- C_Packet9600 Basic Service 53, Packet access 9.6 kbps

2) ur ------ user rate

- '0011'B 2400 bit/s
- '0100'B 4800 bit/s
- '0101'B 9600 bit/s

3) rate ----- channel rate

- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: Used for BS51, BS52, BS53

Test Suite Operation Definition

Operation Name: OO_IFsetup_BS61orBS81_MO(srv:SERVICES; ur:B_4; sa:B_1; ce:B_2; rate:IA5String)

Result Type: BOOLEAN

Comments:

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS, when trigged, is able to initiate the call for BS61 or BS81 with the specified ur, ce and sa values, and the first bearer capability is speech. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

This TSO is used in MO call test cases except TC_11_2, where this TSO is call for single-numbering scheme MT testing.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:

1) srv ----- basic service

- C_AltSpchData Basic Service 61, Alternatie Speech/data
- C_SpchData Basic Service 81, Speech followed by data

2) ur ------ user rate

- '0001'B 300 bit/s
- '0010'B 1200 bit/s
- '0111'B 1200/75 bit/s
- '0011'B 2400 bit/s
- '0100'B 4800 bit/s
- '0101'B 9600 bit/s

3) sa ------ synchronous/asynchronous

- '0'B synchronous
- '1'B asynchronous

4) ce ------ connection element

- '00'B transparent
- '01'B non transparent

5) rate ----- channel rate

- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: used for BS61 and BS81

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_TS6x_MO(srv:SERVICES; ur:B_4; ce:B_2; rate:IA5String)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS, when trigged, is able to initiate the call for TS61 and TS62 with the specified ur and ce values, and the first bearer capability is speech. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

This TSO is used in MO call test cases except TC_11_2, where this TSO is call for single-numbering scheme MT testing.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:
1) srv ----- basic service
- C_AltSpchFax Teleservice 61, Alternatie Speech and Facsimile group 3
- C_AutoFax Teleservice 62, Automatic Facsimile group 3

2) ur ------ user rate
- '0011'B 2400 bit/s
- '0100'B 4800 bit/s
- '0101'B 9600 bit/s

3) ce ------ connection element
- '00'B transparent
- '01'B non transparent

4) rate ----- channel rate
- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: used for TS61 and TS62

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS2x_MT(srv:SERVICES; itc:B_3)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS is able to successfully receive the call for the BS2x service `srv` with the specified itc value. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT paramters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

Detailed Comments: Used for BS21, BS22, BS24, BS25, BS26

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS3x_MT(srv:SERVICES; itc, sacp:B_3)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, orthe MS configuration in such a way that the MS is able to successfully receive the call for the BS3x service `srv` with specified itc and sacp values. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

Detailed Comments: Used for BS31, BS32, BS33, BS34

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS61orBS81_MT(srv:SERVICES; ur:B_4; sa:B_1)
Result Type:	BOOLEAN
Comments:	
Description	
This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS is able to successfully receive the call for BS61 or BS81 with the specified ur and sa values. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE. Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.	
Detailed Comments:	used for BS61 and BS81

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_TS6x_MT(srv:SERVICES; ur:B_4)
Result Type:	BOOLEAN
Comments:	
Description	
This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS is able to successfully receive the call for TS61 and TS62 with the specified ur value. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE. Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.	
Detailed Comments:	used for TS61 and TS62

Test Suite Operation Definition	
Operation Name:	OO_InCallModi(srv:SERVICES)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to initiate a in-call modification at the Mobile Station under test, which is triggered by the calling tone identification (CNG) received by the MS, for the basic service 'srv', then returns TRUE. The calling test step can pass the 'srv' value either as TSPX or constant to the TSO. The possible service values include (see also test suite constant declarations): 1) Teleservices - C_Telephony - C_EmgCall - C_AltSpchFax - C_AutoFax 2) Bearer services - C_Async300, C_Async1200, C_Async120075, C_Async2400, C_Async4800, C_Async9600 - C_Sync1200, C_Sync2400, C_Sync4800, C_Sync9600 - C_PAD300, C_PAD1200, C_PAD120075, C_PAD2400, C_PAD4800, C_PAD9600 - C_Packet2400, C_Packet4800, C_Packet9600 - C_AltSpchData - C_SpchData	
Detailed Comments:	

Test Suite Operation Definition

Operation Name: OO_InitCall(srv:SERVICES)
Result Type: BOOLEAN
Comments:

Description

This operation requests (e.g. from the control console of the test system) the test operator to initiate an MS originating call for the basic service `srv`, then returns TRUE. The calling test step can pass the 'srv' value either as TSPX or constant to the TSO. The necessary MS configuration has been done in the corresponding TSO OO_IFsetup_xx...xx_MO before this TSO is called.

The possible service values include (see also test suite constant declarations):

1) Teleservices

- C_Telephony ----- TS11
- C_EmgCall ----- TS12
- C_AltSpchFax ----- TS61
- C_AutoFax ----- TS62

2) Bearer services

- C_Async300 ----- BS21
- C_Async1200 ----- BS22
- C_Async120075 ----- BS23
- C_Async2400 ----- BS24
- C_Async4800 ----- BS25
- C_Async9600 ----- BS26
- C_Sync1200 ----- BS31
- C_Sync2400 ----- BS32
- C_Sync4800 ----- BS33
- C_Sync9600 ----- BS34
- C_PAD300 ----- BS41
- C_PAD1200 ----- BS42
- C_PAD120075 ----- BS43
- C_PAD2400 ----- BS44
- C_PAD4800 ----- BS45
- C_PAD9600 ----- BS46
- C_Packet2400 ----- BS51
- C_Packet4800 ----- BS52
- C_Packet9600 ----- BS53
- C_AltSpchData ----- BS61
- C_SpchData ----- BS81

Detailed Comments:

Test Suite Operation Definition

Operation Name: OO_InitNonCallSS
Result Type: BOOLEAN
Comments:

Description

This operation requests (e.g. from the control console of the test system) the test operator to initiate a non call related supplementary service which is supported by the MS.

After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OO_InitSS(action: IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to initiate the required supplementary service by the MMI sequence `action`. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_InServiceCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the Mobile Station under test is in normal service state ("Idle, updated")-- listening to CCCH and BCCH and with U1 UPDATED status. If the operator informs (e.g. from the control console of the test system) the test system that the MS does not give any service indication, the operation returns FALSE. If the operator informs the test system that the MS gives service indication, the operation returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_Key
Result Type:	BOOLEAN
Comments:	
Description	
This operation returns TRUE if a key of the SS' keyboard has been pressed. Otherwise it returns FALSE. After OO_PressKeyWhenInService was called, the TSO is then continuously called until the test operator hits a key or a timer expires.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_MptyCall
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to enter MultiParty MMI command at the Mobile Station under test, then it returns with the value TRUE after the operator finishes the action and informs the test system.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_MSSetupStoreClass1SMInMEMemory
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to setup the MS under test to store class 1 SM in the ME memory (by way of MMI, as described in PICS/PIXIT statement). After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PLMNsCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the Mobile Station presents a list of available PLMNs. The list shall include the MNC and MCC of cells 1 to 7, but not cell 8 (for GSM900) or cells 1 to 6, but not cell 7 (for DCS1800). If the operator informs (e.g. from the control console of the test system) the test system that the MS correctly presents the list, the operation returns TRUE. If the operator informs the test system that the list is incorrect, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PLMNselModeAuto
Result Type:	BOOLEAN
Comments:	
Description	
This operation switches the PLMN selection mode of the MS to automatic selection. After correctly switching of the mode the operation returns TRUE otherwise FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PLMNselModeMan
Result Type:	BOOLEAN
Comments:	
Description	
This operation switches the PLMN selection mode of the MS to manual selection. After correctly switching of the mode the operation returns TRUE otherwise FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PowerUp
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to power up the Mobile Station under test and to inform after the test system, then it returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PowerDown
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to power down the Mobile Station under test and to inform after the test system, then it returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PressKeyWhenInService
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to press any key when the MS shows the service indicator, then returns. The return value is always TRUE. Working with OO_Key together, the TSO displays a prompt to the test operator and then immediately returns.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_RecallAndDisplaySM(NumbOfIA5chara: INTEGER; msg: OCTETSTRING)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to recall and display an SM stored in the ME of the MS under test e.g. by means of the MMI. Then the operator checks whether the short message contents represented by 'msg' is correctly displayed. The TSO returns TRUE: if SM can be recalled and displayed correctly in comparison with the 'msg', FALSE: if SM can not be recalled or displayed, or displayed incorrectly in comparison with the 'msg'. The TSO needs to convert the 'msg' into IA5 string according to the packing character rules specified in the section 6.1.2 of ETS 300 628 (GSM 03.38). The 'NumbOfIA5chara' indicates the number of IA5 characters in the 'msg'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_RemoveOneSM
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to remove one of the SM from the message store and then to inform the test system. It does not matter which short message has been removed. After that the operation returns with value TRUE. The returned value is not used in the ATS.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_RemvPwrAmp
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to remove the added power amplification at the Mobile Station under test and to inform after the test system, then the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_RFoutputCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the Mobile Station produces any RF output. If the operator informs (e.g. from the control console of the test system) the test system that the MS does not produce any RF output, the operation returns FALSE. If the operator informs the test system that the MS produces RF output, the operation returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SelPLMN(par_plmn:OCTETSTRING)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to select the given PLMN in par_plmn manually, and to inform after the test system then the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SendMOShortMessage
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to send a short message from the mobile station under test. The operation waits for the operator to acknowldge the request which may be done before or after sending the MO Short Message, and then returns the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SendSMSCOMMANDDel
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to send a SMS COMMAND message from the mobile station under test containing requiring to delete the previouly submited SM and then to inform the test system. Then, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SendSMSCOMMANDEnq
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to send a SMS COMMAND message from the mobile station under test containing an enquiry about the previouly submited SMand then to inform the test system. Then, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SetRefuseCall
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to enable call refusal on the MS. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ShortKeyDepr(ch:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to cause a DTMF tone to be generated, e.g. by short depression of the key `ch` on the Mobile Station under test. The operation waits for the operator to acknowldge the request which may be done before or after generating the DTMF tone, and then returns the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIMIns
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to insert the SIM card into the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIM2Ins
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to insert the SIM card 2 into the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE. The SIM Card 2 shall contain the following parameter values, which are different from dafault values(SIM Card 1): IMSI= '001011234' HPLMN_search_period=6min.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIM3Ins
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to insert the SIM card 3 into the Mobile Station under test and turn the MS power on, after the test operator finishes the action and informs the test system it returns with value TRUE. The SIM Card 3 shall contain fixed dialling number allocated and activated.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIMRmv
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to remove the SIM card from the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIMSimulAttIndOK
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check whether the SIM simulator indicates an attempt made by the ME to store the short message in the SIM. The SIM simulator returns the status response "OK" ('90 00"). Then returns. Attempt indicated: TRUE Attempt NOT indicated: FALSE	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIMSimulAttIndMemProblem
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check whether the SIM simulator indicates an attempt made by the ME to store the short message in the SIM. The SIM simulator returns the status response "Memory Problem" ('92 40"). Then returns. Attempt indicated: TRUE Attempt NOT indicated: FALSE	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SSresultCHK(svc:INTEGER)
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check the user indication of the result of the supplementary service ` svc`. If the operator informs (e.g. from the control console of the test system) the test system that the indication is correct, the operation returns TRUE. If the operator informs the test system that the indication is wrong, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SwitchOn
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to switch on (or if there is no switch then to restore the power to) the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SwitchOff
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to switch off (or if there is no switch then to remove the power from) the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_TCHThroConnCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the TCH is through connected. The SS has to generate a noise pattern so that the operator can check this. If the operator informs (e.g. from the control console of the test system) the test system that the TCH is through connected, the operation returns TRUE. If the operator informs the test system that the TCH is not through connected, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_TermCall
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to terminate the ongoing call. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, it returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_TguardTimeOut
Result Type:	BOOLEAN
Comments:	
Description	
This operation informs (e.g. from the control console of the test system) the test operator that the T_guard timer has expired during the test case leading to an inconclusive or fail verdict and returns immediately with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ToneStopCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the MS stops the tone generation If the operator informs (e.g. from the control console of the test system) the test system that the MS does not stop tone generation, the operation returns FALSE. If the operator informs the test system that the MS stops tone generation, the operation returns TRUE. Definition of "tone generation": -During MO call: MS generates alerting tone/presents inband information sent by the network. -During MT call: MS generates ringing tone.	
Detailed Comments:	

Test suite parameter declarations

Test Suite Parameter Declarations			
Parameter Name	**Type**	**PICS/PIXIT Ref**	**Comments**
TSPC_24DataF	BOOLEAN	PICS Table A.25 Item 12	2.4 k full rate data mode supported
TSPC_24DataH	BOOLEAN	PICS Table A.25 Item 13	2.4 k half rate data mode supported
TSPC_48DataF	BOOLEAN	PICS Table A.25 Item 14	4.8 k full rate data mode supported
TSPC_48DataH	BOOLEAN	PICS Table A.25 Item 15	4.8 k half rate data mode supported
TSPC_96Data	BOOLEAN	PICS Table A.25 Item 16	9.6 k full rate data mode supported
TSPC_Feat_A51	BOOLEAN	PICS Table A.2 Item 17	ciphering algorithm A5/1 supported
TSPC_Feat_A52	BOOLEAN	PICS Table A.2 Item 18	ciphering algorithm A5/2 supported
TSPC_AddCharSet	BOOLEAN	PICS Table A.25 Item 47	A, B, C, D chars supported
TSPC_AddInfo_PseudoSynch	BOOLEAN	PICS	Pseudo synchronised supported
TSPC_AlertInd	BOOLEAN	PICS Table A.25 Item 49	alerting indication to the user supported
TSPC_SvcOnTCH	BOOLEAN	PICS Table A.25 Item 22	at least one service on traffic channel supported
TSPC_SMS	BOOLEAN	PICS Table A.25 Item 31	at least one short message service supported
TSPC_SS	BOOLEAN	PICS Table A.25 Item 29	at least one supplementary service supported
TSPC_AutoAutoMode	BOOLEAN	PICS Table A.25 Item 48	automatically enter automatic selection of PLMN mode supported
TSPC_BasCharSet	BOOLEAN	PICS Table A.25 Item 46	Chars 0-9, *, # supported
TSPC_CalledNumDisp	BOOLEAN	PICS Table A.2 Item 1	called number display supported
TSPC_DCS	BOOLEAN	PICS Table A.1 Item 3	testing DCS1800
TSPC_DetachOnPwrDn	BOOLEAN	PICS Table A.25 Item 38	detach on power down supported
TSPC_DetachOnSIMRmv	BOOLEAN	PICS Table A.25 Item 39	detach on SIM remove supported
TSPC_DispRcvSMS	BOOLEAN	PICS Table A.25 Item 34	display of received SMS supported
TSPC_DualRate	BOOLEAN	PICS Table A.25 Item 23	dual rate channel types supported
TSPC_EGSM	BOOLEAN	PICS Table A.1 Item 2	both standard and extended GSM band supported
TSPC_EmgOnly	BOOLEAN	PICS Table A.25 Item 27	the only circuit switched basic service is emergency call
TSPC_followOnReq	BOOLEAN	PICS Table A.25 Item 53	follow-on request procedure supported
TSPC_HalfRateData	BOOLEAN	PICS Table A.25 Item 6	half rate data supported
TSPC_HalfRateSpeech	BOOLEAN	PICS Table A.25 Item 3	half rate speech mode supported
TSPC_InCallMod	BOOLEAN	PICS Table A.25 Item 52	In-Call modification supported
TSPC_NoimmConn	BOOLEAN	PICS Table A.25 Item 51	at least one service not support immediate connection
TSPC_Feat_FND	BOOLEAN	PICS Table A.2 Item 21	FND feature supported
TSPC_NonCallSS	BOOLEAN	PICS Table A.25 Item 30	non call related supplementary service supported
TSPC_MTsvc	BOOLEAN	PICS Table A.25 Item 19	at least one MT circuit switched basic service supported
TSPC_MOsvc	BOOLEAN	PICS Table A.25 Item 20	at least one MO circuit switched basic service supported
TSPC_PGSM	BOOLEAN	PICS Table A.1 Item 1	only standard GSM band supported
TSPC_RefusalCall	BOOLEAN	PICS Table A.25 Item 54	refusal of call supported
TSPC_ReplaceSMS	BOOLEAN	PICS Table A.25 Item 33	replace SMS supported
TSPC_ReplyProc	BOOLEAN	PICS Table A.25 Item 32	(SMS) reply procedures supported
TSPC_RFAmp	BOOLEAN	PICS Table A.25 Item 55	RF amplification supported
TSPC_SDCCHOnly	BOOLEAN	PICS Table A.25 Item 21	only SDCCH supported
TSPC_Serv_SS_AoCC	BOOLEAN	PICS Table A.5 Item 14	AOCC SS supported
TSPC_Serv_SS_BAIC	BOOLEAN	PICS Table A.5 Item 18	BAIC SS supported
TSPC_Serv_SS_BI	BOOLEAN	PICS	BI SS supported

TSPC_Serv_SS_BICRoam	BOOLEAN	PICS Table A.5 Item 19	BICRoam SS supported
TSPC_Serv_SS_BAOC	BOOLEAN	PICS Table A.5 Item 15	BAOC SS supported
TSPC_Serv_SS_BOIC	BOOLEAN	PICS Table A.5 Item 16	BOIC SS supported
TSPC_Serv_SS_BOICexHC	BOOLEAN	PICS Table A.5 Item 17	BOICExHC SS supported
TSPC_Serv_SS_CFB	BOOLEAN	PICS Table A.5 Item 6	CFB supported
TSPC_Serv_SS_CFNRy	BOOLEAN	PICS Table A.5 Item 7	CFNry supported
TSPC_Serv_SS_CFNRc	BOOLEAN	PICS Table A.5 Item 8	CFNRc supported
TSPC_Serv_SS_CFU	BOOLEAN	PICS Table A.5 Item 5	CFU supported
TSPC_Serv_SS_HOLD	BOOLEAN	PICS Table A.5 Item 10	Call Hold SS supported
TSPC_Serv_SS_MPTY	BOOLEAN	PICS Table A.5 Item 11	multiparty SS supported
TSPC_Serv_SS_unstruct	BOOLEAN	PICS	USSD supported
TSPC_Serv_TS11	BOOLEAN	PICS Table A.3 Item 1	telephony supported
TSPC_Serv_TS12	BOOLEAN	PICS Table A.3 Item 2	emergency call supported
TSPC_Serv_TS21	BOOLEAN	PICS Table A.3 Item 3	SMS MT/PP supported
TSPC_Serv_TS22	BOOLEAN	PICS Table A.3 Item 4	SMS MO/PP supported
TSPC_Serv_TS23	BOOLEAN	PICS Table A.3 Item 5	SMS cell broadcast supported
TSPC_Serv_TS61	BOOLEAN	PICS	alternate speech and G3 fax (TS61) supported
TSPC_Serv_TS62	BOOLEAN	PICS	automatic G3 fax (TS62) supported
TSPC_Serv_BS21	BOOLEAN	PICS Table A.4 Item 1	data circuit duplex async 300 bit/s supported
TSPC_Serv_BS22	BOOLEAN	PICS Table A.4 Item 2	data circuit duplex async 1200 bit/s supported
TSPC_Serv_BS23	BOOLEAN	PICS Table A.4 Item 3	data circuit duplex async 1200/75 bit/s supported
TSPC_Serv_BS24	BOOLEAN	PICS Table A.4 Item 4	data circuit duplex async 2400 bit/s supported
TSPC_Serv_BS25	BOOLEAN	PICS Table A.4 Item 5	data circuit duplex async 4800 bit/s supported
TSPC_Serv_BS26	BOOLEAN	PICS Table A.4 Item 6	data circuit duplex async 9600 bit/s supported
TSPC_Serv_BS31	BOOLEAN	PICS Table A.4 Item 7	data circuit duplex sync 1200 bit/s supported
TSPC_Serv_BS32	BOOLEAN	PICS Table A.4 Item 8	data circuit duplex sync 2400 bit/s supported
TSPC_Serv_BS33	BOOLEAN	PICS Table A.4 Item 9	data circuit duplex sync 4800 bit/s supported
TSPC_Serv_BS34	BOOLEAN	PICS Table A.4 Item 10	data circuit duplex sync 9600 bit/s supported
TSPC_Serv_BS41	BOOLEAN	PICS Table A.4 Item 11	PAD access 300 bit/s supported
TSPC_Serv_BS42	BOOLEAN	PICS Table A.4 Item 12	PAD access 1200 bit/s supported
TSPC_Serv_BS43	BOOLEAN	PICS Table A.4 Item 13	PAD access 1200/75 bit/s supported
TSPC_Serv_BS44	BOOLEAN	PICS Table A.4 Item 14	PAD access 2400 bit/s supported
TSPC_Serv_BS45	BOOLEAN	PICS Table A.4 Item 15	PAD access 4800 bit/s supported
TSPC_Serv_BS46	BOOLEAN	PICS Table A.4 Item 16	PAD access 9600 bit/s supported
TSPC_Serv_BS51	BOOLEAN	PICS Table A.4 Item 17	packet access 2400 bit/s supported
TSPC_Serv_BS52	BOOLEAN	PICS Table A.4 Item 18	packet access 4800 bit/s supported
TSPC_Serv_BS53	BOOLEAN	PICS Table A.4 Item 19	packet access 9600 bit/s supported
TSPC_Serv_BS61	BOOLEAN	PICS	alternate speech/data (BS61) supported
TSPC_Serv_BS81	BOOLEAN	PICS	speech followed data (BS81) supported
TSPC_SMSStatusRepCap	BOOLEAN	PICS Table A.25 Item 35	SMS status report capabilities supported
TSPC_StoreRcvSMSME	BOOLEAN	PICS Table A.25 Item 37	Storage of received SMS in ME supported
TSPC_StoreRcvSMSSIM	BOOLEAN	PICS Table A.25 Item 36	Storage of received SMS in SIM supported
TSPC_SIMRmv	BOOLEAN	PICS Table A.25 Item 40	SIM removable without power down supported
TSPC_SwitchOnOff	BOOLEAN	PICS Table A.2 Item 15	switch on/off supported
TSPC_TranspDataOnly	BOOLEAN	PICS Table A.25 Item 9	only transparent data service supported
TSPC_CC	BOOLEAN	PICS Table A.25 Item 26	CC protocol for at least one BC

			supported
TSPC_TeleSvc	BOOLEAN	PICS Table A.25 Item 25	at least one teleservice supported
TSPC_EFR	BOOLEAN		Set to TRUE for GSM-900 Phase 2+ (EFR) MS which supports any EFR features, and FALSE otherwise.
TSPC_EFR_Speech_v2	BOOLEAN		Set to TRUE for GSM-900 Phase 2+ (EFR) MS which supports Speech Version 2 and FALSE otherwise.
TSPC_EFR_Speech_v3	BOOLEAN		Set to TRUE for GSM-900 Phase 2+ (EFR) MS which supports Speech Version 3 and FALSE otherwise.
TSPC_EFR_EmgCallBcap	BOOLEAN		Set to TRUE if received ESETUP msg contains bearer capability IE, otherwise FALSE.
TSPC_FullRateSpeech	BOOLEAN	PICS Table A.25 Item 2	full rate speech mode supported
TSPC_FullRateOnly	BOOLEAN	PICS Table A.25 Item 24	only full rate channel type supported
TSPX_AltNb	BOOLEAN	PIXIT	use alternative neighbour cells description. default : FALSE
TSPX_TE_stopbit	B_1	PIXIT	TE conf. number stop bits. default value : '0'B, 1 bit
TSPX_TE_databit	B_1	PIXIT	TE conf. number data bits. default value : '1'B, 8 bits
TSPX_TE_FLCT	FLWCNTL	PIXIT	TE flow control. default : 0, outband flow control
TSPX_TE_parity	B_3	PIXIT	TE conf. parity. default : '001'B, no parity
TSPX_BS_21_itc1	B_3	PIXIT	itc value supported for BS21. default value : '001'B, UDI
TSPX_BS_21_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS21. default value : FALSE, only one itc
TSPX_BS_21_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS21. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_21_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_21_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent mode
TSPX_BS_21_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_21_itc2	B_3	PIXIT	other itc value supported by BS21, it shall be differenet from the previous one if the answer to TSPX_BS_21_more_itc is TRUE. default value :'001'B, UDI
TSPX_BS_22_itc1	B_3	PIXIT	itc value supported by BS22. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_22_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS22. default value : FALSE, only one itc
TSPX_BS_22_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS22. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_22_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_22_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent
TSPX_BS_22_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_22_itc2	B_3	PIXIT	other itc value supported by

			BS22, it shall be differenet from the previous one if the answer to TSPX_BS_22_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_23_itc	B_3	PIXIT	itc value supported by BS23. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_23_T_NT	BOOLEAN	PIXIT	both T and NT supported for BS23. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_23_ce	B_2	PIXIT	ce value used in MO call. default value : '00'B, transparent
TSPX_BS_23_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_24_itc1	B_3	PIXIT	itc value supported by BS24. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_24_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS24. default value : FALSE, only one itc
TSPX_BS_24_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS24. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_24_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_24_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_24_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_24_itc2	B_3	PIXIT	other itc value supported by BS24, it shall be differenet from the previous one if the answer to TSPX_BS_24_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_25_itc1	B_3	PIXIT	itc value supported by BS25. default value : '001'B, UDI
TSPX_BS_25_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS25. default value : FALSE, only one itc
TSPX_BS_25_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS25. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_25_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_25_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_25_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_25_itc2	B_3	PIXIT	other itc value supported by BS25, it shall be differenet from the previous one if the answer to TSPX_BS_25_more_itc is TRUE. default value : '001'B, UDI
TSPX_BS_26_itc1	B_3	PIXIT	itc value supported by BS26. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_26_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS26. default value : FALSE, only one itc
TSPX_BS_26_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS26. default value : FALSE, not support both transparent and non transparent modes

TSPX_BS_26_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_26_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_26_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_26_itc2	B_3	PIXIT	other itc value supported by BS26, it shall be differenet from the previous one if the answer to TSPX_BS_26_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_31_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS31. default value : FALSE, only one itc
TSPX_BS_31_more_sacp	BOOLEAN	PIXIT	more than one sacp value supported by BS31. default value : FALSE, only one sacp
TSPX_BS_31_itc1	B_3	PIXIT	itc value supported by BS31. default value : '001'B, UDI
TSPX_BS_31_sacp1	B_3	PIXIT	sacp value supported by BS31. default value : '001'B, I440/450
TSPX_BS_31_itc2	B_3	PIXIT	other itc value supported by BS31, it shall be differenet from the previous one if the answer to TSPX_BS_31_more_itc is TRUE. default value : '001'B, UDI
TSPX_BS_31_sacp2	B_3	PIXIT	other sacp value supported by BS31, it shall be differenet from the previous one if the answer to TSPX_BS_31_more_sacp is TRUE. default value : '001'B, I440/450
TSPX_BS_32_itc1	B_3	PIXIT	itc value supported by BS32. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_32_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS32. default value : FALSE, only one itc
TSPX_BS_32_more_sacp	BOOLEAN	PIXIT	more than one sacp value supported by BS32. default value : FALSE, only one sacp
TSPX_BS_32_sacp1	B_3	PIXIT	sacp value supported by BS32. default value : '110'B, X.32
TSPX_BS_32_X32_T_NT	BOOLEAN	PIXIT	BS 32 both T and NT supported for X32 3.1kHz. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_32_X32_ce	B_2	PIXIT	if the answer to TSPX_BS_32_X32_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent
TSPX_BS_32_itc2	B_3	PIXIT	other itc value supported by BS32, it shall be differenet from the previous one if the answer to TSPX_BS_32_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_32_sacp2	B_3	PIXIT	other sacp value supported by BS32, it shall be differenet from the previous one if the answer to TSPX_BS_32_more_sacp is TRUE. default value : '110'B, X.32
TSPX_BS_33_itc1	B_3	PIXIT	itc value supported by BS33. default value : '001'B, UDI
TSPX_BS_33_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS33. default value : FALSE, only one itc

TSPX_BS_33_more_sacp	BOOLEAN	PIXIT	more than one sacp value supported by BS33. default value : FALSE, only one sacp
TSPX_BS_33_sacp1	B_3	PIXIT	sacp value supported by BS33. default value : '010'B, X.21
TSPX_BS_33_X32_T_NT	BOOLEAN	PIXIT	BS 33 both T and NT supported for X32 3.1kHz. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_33_X32_ce	B_2	PIXIT	if the answer to TSPX_BS_33_X32_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_33_itc2	B_3	PIXIT	other itc value supported by BS32, it shall be differenet from the previous one if the answer to TSPX_BS_33_more_itc is TRUE. default value : '001'B, UDI
TSPX_BS_33_sacp2	B_3	PIXIT	other sacp value supported by BS33, it shall be differenet from the previous one if the answer to TSPX_BS_33_more_sacp is TRUE. default value : '010'B, X.21
TSPX_BS_34_itc1	B_3	PIXIT	itc value supported by BS34. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_34_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS34. default value : FALSE, only one itc
TSPX_BS_34_more_sacp	BOOLEAN	PIXIT	more than one sacp value supported by BS34. default value : FALSE, only one sacp
TSPX_BS_34_sacp1	B_3	PIXIT	sacp value supported by BS34. default value : '001'B, I440/450
TSPX_BS_34_X32_T_NT	BOOLEAN	PIXIT	BS 34 both T and NT supported for X32 3.1kHz. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_34_X32_ce	B_2	PIXIT	if the answer to TSPX_BS_34_X32_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_34_itc2	B_3	PIXIT	other itc value supported by BS34, it shall be differenet from the previous one if the answer to TSPX_BS_34_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_34_sacp2	B_3	PIXIT	other sacp value supported by BS34, it shall be differenet from the previous one if the answer to TSPX_BS_34_more_sacp is TRUE. default value : '001'B, I440/450
TSPX_BS_41_T_NT	BOOLEAN	PIXIT	BS41 both T and NT supported . default value : FALSE, not support both
TSPX_BS_41_ce	B_2	PIXIT	ce value used in BS41 MO call. default value : '00'B, transparent
TSPX_BS_42_T_NT	BOOLEAN	PIXIT	BS42 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_42_ce	B_2	PIXIT	ce value used in BS42 MO call. default value : '010'B, non transparent
TSPX_BS_43_T_NT	BOOLEAN	PIXIT	BS43 both T and NT supported . default value : FALSE, not

			support both transparent and non transparent modes
TSPX_BS_43_ce	B_2	PIXIT	ce value used in BS43 MO call. default value : '00'B, transparent
TSPX_BS_44_T_NT	BOOLEAN	PIXIT	BS44 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_44_ce	B_2	PIXIT	ce value used in BS44 MO call. default value : '00'B, transparent
TSPX_BS_45_T_NT	BOOLEAN	PIXIT	BS45 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_45_ce	B_2	PIXIT	ce value used in BS45 MO call. default value : '01'B, non transparent
TSPX_BS_46_T_NT	BOOLEAN	PIXIT	BS46 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_46_ce	B_2	PIXIT	ce value used in BS46 MO call. default value : '00'B, transparent
TSPX_BS_61_S	BOOLEAN	PIXIT	synchronous data supported by BS 61 . default value : TRUE, synchronous mode
TSPX_BS_61_S_more_ur	BOOLEAN	PIXIT	more than one user rate for synchronous data service supported by BS61. default value : FALSE, only ine user rate
TSPX_BS_61_S_ur1	B_4	PIXIT	usrer rate value supported for synchronous data service of BS61. default value : '0100'B, 4.8 kbit/s
TSPX_BS_61_S_ur2	B_4	PIXIT	another usrer rate value supported for synchronous data service of BS61. it shall be different from the previous one if the answer to TSPX_BS_61_S_more_ur is TRUE. default value : '0100'B, 4.8 kbit/s
TSPX_BS_61_A	BOOLEAN	PIXIT	asynchronous data service supported by BS61 . default value : TRUE, asynchronous
TSPX_BS_61_A_ur1	B_4	PIXIT	usrer rate value supported for asynchronous data service of BS61. default value : '0101'B, 9.6 kbit/s
TSPX_BS_61_A_ur1_T_NT	BOOLEAN	PIXIT	both T and NT supported for the rate TSPX_BS_61_A_ur1 of BS61. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_61_A_ur1_ce	B_2	PIXIT	if the answer to TSPX_BS_61_A_ur1_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_61_A_more_ur	BOOLEAN	PIXIT	more than one user rate for asynchronous data service supported by BS61. default value : FALSE, only one user rate
TSPX_BS_61_A_ur2	B_4	PIXIT	another usrer rate value supported for asynchronous data service of BS61. it shall be different from the previous one if the answer to TSPX_BS_61_A_more_ur is TRUE. default value : '0101'B, 9.6 kbit/s
TSPX_BS_61_A_ur2_T_NT	BOOLEAN	PIXIT	both T and NT supported for the rate TSPX_BS_61_A_ur2 of BS61. default value : FALSE, not

			support both transparent and non transparent modes
TSPX_BS_81_S	BOOLEAN	PIXIT	synchronous data supported by BS 81. default value : TRUE, synchronous mode
TSPX_BS_81_S_more_ur	BOOLEAN	PIXIT	more than one user rate for synchronous data service supported by BS81. default value : FALSE, only one user rate
TSPX_BS_81_S_ur1	B_4	PIXIT	usrer rate value supported for synchronous data service of BS81. default value : '0101'B, 9.6 kbit/s
TSPX_BS_81_S_ur2	B_4	PIXIT	another usrer rate value supported for synchronous data service of BS81. it shall be different from the previous one if the answer to TSPX_BS_81_S_more_ur is TRUE. default value : '0101'B, 9.6 kbit/s
TSPX_BS_81_A	BOOLEAN	PIXIT	asynchronous data service supported by BS81. default value : TRUE, asynchronous mode
TSPX_BS_81_A_ur1	B_4	PIXIT	usrer rate value supported for asynchronous data service of BS81. default value : '0100'B, 4.8 kbit/s
TSPX_BS_81_A_ur1_T_NT	BOOLEAN	PIXIT	both T and NT supported for the rate TSPX_BS_81_A_ur1 of BS81. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_81_A_ur1_ce	B_2	PIXIT	if the answer to TSPX_BS_81_A_ur1_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent
TSPX_BS_81_A_more_ur	BOOLEAN	PIXIT	more than one user rate for asynchronous data service supported by BS81. default value : FALSE, only one user rate
TSPX_BS_81_A_ur2	B_4	PIXIT	another usrer rate value supported for asynchronous data service of BS81. it shall be different from the previous one if the answer to TSPX_BS_81_A_more_ur is TRUE. default value : '0100'B, 4.8 kbit/s
TSPX_BS_81_A_ur2_T_NT	BOOLEAN	PIXIT	both T and NT supported for the rate TSPX_BS_81_A_ur2 of BS81. default value : FALSE, not support both transparent and non transparent modes
TSPX_TS_61_T_NT	BOOLEAN	PIXIT	both T and NT supported for TS61(fax). default value : FALSE, not support both transparent and non transparent modes
TSPX_TS_61_ce	B_2	PIXIT	if the answer to TSPX_TS_61_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent
TSPX_TS_61_ur1	B_4	PIXIT	user rate for TS61(fax). default value : '0101'B, 9.6 kbit/s
TSPX_TS_61_more_ur	BOOLEAN	PIXIT	more than one user rate supported for TS61. default value : FALSE, only one user rate
TSPX_TS_61_ur2	B_4	PIXIT	another user rate for TS61, it shall be different from the previous one if the answer to TSPX_TS_61_more_ur is TRUE.

			default value : '0101'B, 9.6 kbit/s
TSPX_TS_62_T_NT	BOOLEAN	PIXIT	both T and NT supported for TS62. default value : FALSE, not support both transparent and non transparent modes
TSPX_TS_62_ce	B_2	PIXIT	if the answer to TSPX_TS_62_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_TS_62_ur1	B_4	PIXIT	user rate for TS62. default value : '0100'B, 4.8 kbit/s
TSPX_TS_62_more_ur	BOOLEAN	PIXIT	more than one user rate supported by TS62. default value : FALSE, only one user rate
TSPX_TS_62_ur2	B_4	PIXIT	another user rate for TS62, it shall be different from the previous one if the answer to TSPX_TS_62_more_ur is TRUE. default value : '0100'B, 4.8 kbit/s
TSPX_DTMF	B_1	PIXIT	call control capabilities: value '1'B means MS supports DTMF
TSPX_modF	CHMOD_VAL	PIXIT	any non signalling full rate channel mode for TC_26_6_4_1
TSPX_modH	CHMOD_VAL	PIXIT	any non signalling half channel mode for TC_26_6_4_1
TSPX_anymod	CHMOD_VAL	PIXIT	channel mode for TC_62_10_2_4_2. Any supported value except signalling and Full rate speech. possible values are '00000011'B' 12k data, '00001011'B 6k data, '0010011'B 3.6k data
TSPX_CKSNA	CKSN	PIXIT	cipher key sequence number
TSPX_CKSNB	CKSN	PIXIT	cipher key sequence number
TSPX_CKSNC	CKSN	PIXIT	cipher key sequence number
TSPX_CKSNDef	CKSN	PIXIT	default cipher key sequence number
TSPX_RfPwrCap	B_3	PIXIT	RF power capability
TSPX_RevLevel	B_2	PIXIT	Revision level of classmark 1
TSPX_CiphAlgA5_1	B_1	PIXIT	A5/1 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_2	B_1	PIXIT	A5/2 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_3	B_1	PIXIT	A5/3 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_4	B_1	PIXIT	A5/4 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_5	B_1	PIXIT	A5/5 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_6	B_1	PIXIT	A5/6 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_7	B_1	PIXIT	A5/7 algorithm: '1'B : available, '0'B: not available
TSPX_pSyncCap	B_1	PIXIT	pseudo synchronisation capability, '0'B: not present '1'B: present
TSPX_SSscrnInd	B_2	PIXIT	SS screen indicator
TSPX_SMCap	B_1	PXIIT	short message capability, '0'B: not present '1'B: present
TSPX_frqCap	B_1	PIXIT	frequency capability. DCS: reserved, GSM: '0'B not support extension band, '1'B support extension band
TSPX_ClsMk3	B_1	PIXIT	classmark 3 indicator, '0'B no addition capability info, '1'B additional capability info in class mark 3
TSPX_RfPwrCapAmp	B_3	PIXIT	RF power capability with external RF amplifier
TSPX_CphAlgA	CPHALG	PIXIT	any supported ciphering algorithm

TSPX_CphAlgB	CPHALG	PIXIT	ciphering algorithm supported
TSPX_CphAlgC	CPHALG	PIXIT	ciphering algorithm supported
TSPX_CphAlgD	CPHALG	PIXIT	ciphering algorithm supported
TSPX_CphAlgE	CPHALG	PIXIT	ciphering algorithm supported
TSPX_CphAlgDef	CPHALG	PIXIT	default ciphering algorithm
TSPX_DTMFInd	BOOLEAN	PIXIT	DTMF indication to user supported
TSPX_NoOfHoAccA	INTEGER	PIXIT	number of handover access bursts, value range 10 - 20
TSPX_NoOfHoAccB	INTEGER	PIXIT	number of handover access bursts, value range 10 - 20
TSPX_NoOfHoAccC	INTEGER	PIXIT	number of handover access bursts, value range 10 - 20
TSPX_NoOfHoAccD	INTEGER	PIXIT	number of handover access bursts, value range 5 - 10
TSPX_NoOfHoAccE	INTEGER	PIXIT	number of handover access bursts, value range 5 - 10
TSPX_NoOfHoAccF	INTEGER	PIXIT	number of handover access bursts, value range 5 - 10
TSPX_NoOfHoAccG	INTEGER	PIXIT	number of handover access bursts, value range 2 - 5
TSPX_NoOfHoAccH	INTEGER	PIXIT	number of handover access bursts, value range 2 - 5
TSPX_NoOfHoAccI	INTEGER	PIXIT	number of handover access bursts, value range 2 - 5
TSPX_HoRefA	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefB	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefC	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefD	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefE	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefF	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefG	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefH	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefI	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefJ	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HSN	INTEGER	PIXIT	Hopping sequence number value range: 0 - 63
TSPX_IMSI	HEXSTRING	PIXIT	IMSI of the MS
TSPX_IMEI	HEXSTRING	PIXIT	IMEI of the MS.
TSPX_IMEISV	HEXSTRING	PIXIT	IMEISV of the MS. Used in: TC_26_6_8_5, TC_26_7_3_1
TSPX_Ki	BITSTRING	PIXIT	default authentication key used in testing
TSPX_MAIO	INTEGER	PIXIT	mobile allocation index offset, value range: 0 - 63
TSPX_MaxRetrans	INTEGER	PIXIT	Max-Retrans
TSPX_MSTxpwrMax	MAXTXPOW	PIXIT	maximum output power from MS
TSPX_PwrlvlA	INTEGER	PIXIT	power level, value between 2 to 15
TSPX_PwrlvlB	INTEGER	PIXIT	power level, value between 2 to 15
TSPX_PwrlvlC	INTEGER	PIXIT	power level, value between 2 to 15
TSPX_PwrlvlD	INTEGER	PIXIT	power level, value between 2 to 15
TSPX_RANDA	RAND	PIXIT	challenge RAND
TSPX_RANDB	RAND	PIXIT	challenge RAND

TSPX_RANDC	RAND	PIXIT	challenge RAND
TSPX_RANDDef	RAND	PIXIT	default challenge RAND
TSPX_SDCCH4SubA	B_2	PIXIT	TDMA offset of SDCCH/4 subchannel
TSPX_SDCCH4SubB	B_2	PIXIT	TDMA offset of SDCCH/4 subchannel
TSPX_SDCCH4SubC	B_2	PIXIT	TDMA offset of SDCCH/4 subchannel
TSPX_SDCCH4SubDef	B_2	PIXIT	TDMA offset of default SDCCH/4 subchannel
TSPX_SDCCH8SubA	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubB	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubC	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubD	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubE	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubF	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubG	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubDef	B_3	PIXIT	TDMA offset of default SDCCH/8 subchannel
TSPX_TC1M	INTEGER	PIXIT	timer value for GSM timer TC1M (for SMS)
TSPX_MaxCPDataRetx	INTEGER	PIXIT	max. number of CP data retransmissions for SMS
TSPX_Immconn	BOOLEAN	PIXIT	Immediate connect supported
TSPX_TimadvA	INTEGER	PIXIT	timing advance
TSPX_TimadvB	INTEGER	PIXIT	timing advance
TSPX_TimadvC	INTEGER	PIXIT	timing advance
TSPX_TmSltA	SN	PIXIT	time slot
TSPX_TmSltB	SN	PIXIT	time slot
TSPX_TmSltC	SN	PIXIT	time slot
TSPX_TmSltD	SN	PIXIT	time slot
TSPX_TmSltE	SN	PIXIT	time slot
TSPX_TmSltF	SN	PIXIT	time slot
TSPX_TmSltG	SN	PIXIT	time slot
TSPX_TmSltDef	SN	PIXIT	default time slot
TSPX_TmSltNotZero	SN	PIXIT	time slot, arbitrarily value, but not zero.
TSPX_TmSltNotZero1	SN	PIXIT	time slot, arbitrarily value, but not zero and not TSPX_TmSltNotZero
TSPX_Txint	INTEGER	PIXIT	Tx-Integer
TSPX_TscA	TSC	PIXIT	training sequence code
TSPX_TscB	TSC	PIXIT	training sequence code
TSPX_TscC	TSC	PIXIT	training sequence code
TSPX_TscD	TSC	PIXIT	training sequence code
TSPX_TscE	TSC	PIXIT	training sequence code
TSPX_TscF	TSC	PIXIT	training sequence code
TSPX_TscG	TSC	PIXIT	training sequence code
TSPX_TscDef	TSC	PIXIT	default TSC
TSPX_T3122	WI	PIXIT	value of timer T3122, HEXSTRING[2]
TSPX_TCHcarrierA_ho	INTEGER	PIXIT	the value can be choosen arbitrarily from cell allocation of cell A (GSM), but not BCCH carrier .possible value is : 10, 17, 26, 34, 42, 45, 46, 52, 59, 66, 73, 74, 75, 76, 108, 114
TSPX_TCHcarrierA_hod	INTEGER	PIXIT	the value can be choosen arbitrarily from cell allocation of cell A (DCS), but not BCCH carrier . possible value is : 734,741,754,759,762,766,767,773,775,779,782,791,798,829,832,8

			44
TSPX_TCHcarrierB	INTEGER	PIXIT	TCH and SDCCH channel frequency number of cell B (GSM)
TSPX_TCHcarrierBd	INTEGER	PIXIT	TCH and SDCCH channel frequency number of cell B (DCS)
TSPX_TCHcarrierB_ho	INTEGER	PIXIT	the value can be choosen arbitrarily from cell allocation of cell B (GSM), but not BCCH carrier . possibles value is : 14,18,22,24,30,31,38,60,66,73,74 ,75,76,108,114
TSPX_TCHcarrierB_hod	INTEGER	PIXIT	not BCCH carrier of cell B. the value can be choosen arbitrarily from cell allocation of cell B (DCS), which is (739,743,746,749,756,758,761,77 1,779,782,791,798,829,832,844)
TSPX_TCHcarrierB2_ho	INTEGER	PIXIT	Choosen arbitrarly from cell allocation B for GSM HO cases, but not BCCH carrier!
TSPX_TCHcarrierB2_hod	INTEGER	PIXIT	Choosen arbitrarly from cell allocation B for DCS HO cases, but not BCCH carrier!
TSPX_TCHHSubA	B_1	PIXIT	TDMA offset of half rate subchannel
TSPX_TCHHSubDef	B_1	PIXIT	TDMA offset of default half rate subchannel
TSPX_TMSI	TMSI_V	PIXIT	TMSI of the MS used in test
TSPX_TMSI1	TMSI_V	PIXIT	another TMSI used in test which shall differ from TSPX_TMSI, TSPX_TMSI + '01'O TSPX_TMSI + '02'O TSPX_TMSI + '03'O
TSPX_Uupd	B_8	PIXIT	user-user protocol discriminator
TSPX_UuInfo	OCTETSTRING	PIXIT	user-user information
TSPX_k	INTEGER	PIXIT	timing of cell A before cell B k bit periods for TC_26_6_5_1, TC_26_6_5_2_? TC_26_6_5_3_?, TC_26_6_5_4_?.
TSPX_y	INTEGER	PIXIT	timing advance for TC_26_6_5_3_?, TC_26_6_5_4_?, TC_26_6_5_5_? . The values of TSPX_k (or TSPX_k1) and TSPX_y: 0 < (2*TSPX_k + TSPX_y) MOD 256 < 60.
TSPX_k1	INTEGER	PIXIT	timing of cell A before cell B k bit periods for TC_26_6_5_5_?
TSPX_k2	INTEGER	PIXIT	timing of cell A before cell B k bit periods for TC_26_6_5_6
TSPX_y2	INTEGER	PIXIT	timing advance for TC_26_6_5_6, value range : 11 - 62.
TSPX_k3	INTEGER	PIXIT	timing of cell A before cell B k bit periods for TC_26_6_5_7
TSPX_y3	INTEGER	PIXIT	timing advance for TC_26_6_5_7
TSPX_nPara	INTEGER	PIXIT	the n'th ChReq for TC_26_6_1_2, shall be choose to [1..8], MAXRETRANS for TC_26_6_1_2 is 7 (See 11.10 for more explaination).
TSPX_kPara	INTEGER	PIXIT	the k'th ChReq for TC_26_6_1_2
TSPX_rPara	INTEGER	PIXIT	the r'th ChReq for TC_26_6_1_2
TSPX_i1Para	INTEGER	PIXIT	for TC_26_1_2
TSPX_i2Para	INTEGER	PIXIT	for TC_26_1_2
TSPX_i3Para	INTEGER	PIXIT	the r'th ChReq for TC_26_6_1_2
TSPX_n1Para	INTEGER	PIXIT	the n'th ChReq for TC_26_6_1_3
TSPX_i4Para	INTEGER	PIXIT	for TC_26_6_1_3
TSPX_xPara	INTEGER	PIXIT	t3122 for TC_26_6_1_3
TSPX_AGBLKS1	INTEGER	PIXIT	BS-AG-BLKS-RES for TC_26_6_2_3_1, TC_26_6_2_1_1

TSPX_PAMFRMS1	INTEGER	PIXIT	BS-PA-MFRMS for TC_26_6_2_3_1 (shall not be set to 9), TC_26_6_2_1_1
TSPX_CcchConf1	CCCH_CON	PIXIT	paging subchannel for TC_26_6_2_3_1, TC_26_6_2_1_1
TSPX_PgSubch	INTEGER	PIXIT	paging subchannel for TC_26_6_2_3_1
TSPX_AGBLKS2	INTEGER	PIXIT	BS-AG-BLKS-RES for TC_26_6_2_3_2, TC_26_6_2_1_2
TSPX_PAMFRMS2	INTEGER	PIXIT	BS-PA-MFRMS for TC_26_6_2_3_2, TC_26_6_2_1_2
TSPX_CcchConf2	CCCH_CON	PIXIT	paging subchannel for TC_26_6_2_3_2, TC_26_6_2_1_2
TSPX_AGBLKS3	INTEGER	PIXIT	BS-AG-BLKS-RES for TC_26_6_2_4, TC_26_6_2_2, TC_26_6_2_1_3
TSPX_PAMFRMS3	INTEGER	PIXIT	BS-PA-MFRMS for TC_26_6_2_4, TC_26_6_2_2, TC_26_6_2_1_3
TSPX_CcchConf3	CCCH_CON	PIXIT	paging subchannel for TC_26_6_2_4, TC_26_6_2_2, TC_26_6_2_1_3
TSPX_AGBLKS4	INTEGER	PIXIT	BS-AG-BLKS-RES for TC_26_6_2_5
TSPX_PAMFRMS4	INTEGER	PIXIT	BS-PA-MFRMS for TC_26_6_2_5
TSPX_CcchConf4	CCCH_CON	PIXIT	paging subchannel for TC_26_6_2_5 (shall be in the set ('010', '100', '110'))
TSPX_Chtp1	CH_TDMA	PIXIT	channel type for TC_26_6_13_1, any value supported by the MS
TSPX_ChMod1	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_1, any value for the channel type of TSPX_Chtp1
TSPX_Maio1	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma1, default : 2
TSPX_Ma1	MAC	PIXIT	mobile allocation for TC_26_6_13_1, its value shall indicate number of frequencies between 1 and 17. default : '010101'O
TSPX_Hsn1	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1. default: 1
TSPX_Maio2	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma2, default : 8
TSPX_Ma2	MAC	PIXIT	mobile allocation for TC_26_6_13_1, its value shall indicate number of frequencies between 1 and 17. default :'01362A5'O
TSPX_Hsn2	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, default : 6
TSPX_Maio3	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma3, default : 14
TSPX_Ma3	MAC	PIXIT	mobile allocation for TC_26_6_13_1, its value shall indicate number of frequencies between 1 and 17. default :'01FFFA'O
TSPX_Hsn3	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, default : 0

TSPX_Chtp2	CH_TDMA	PIXIT	channel type for TC_26_6_13_2, any value supported by the MS
TSPX_ChMod2	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_2
TSPX_Maio4	INTEGER	PIXIT	hopping parameter for TC_26_6_13_2, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma4, default : 3
TSPX_Ma4	MAC	PIXIT	mobile allocation for TC_26_6_13_2, its value shall indicate number of frequencies between 1 and 17. default : '001141'O
TSPX_Hsn4	INTEGER	PIXIT	hopping parameter for TC_26_6_13_2, default : 0
TSPX_Maio5	INTEGER	PIXIT	hopping parameter for TC_26_6_13_2, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma5, default : 9
TSPX_Ma5	MAC	PIXIT	mobile allocation for TC_26_6_13_2, its value shall indicate number of frequencies between 1 and 17. default : '01ABCB'O
TSPX_Hsn5	INTEGER	PIXIT	hopping parameter for TC_26_6_13_2, default : 16
TSPX_Chtp3	CH_TDMA	PIXIT	channel type for TC_26_6_13_3, any value supported by the MS
TSPX_Maio6	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma6, default : 4
TSPX_Ma6	MAC	PIXIT	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 1 and 17. default :'006248'O
TSPX_Hsn6	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, default : 4
TSPX_Maio7	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma7, default : 10
TSPX_Ma7	MAC	PIXIT	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 2 and 17. default : '009AFB'O
TSPX_Chtp4	CH_TDMA	PIXIT	channel type for TC_26_6_13_3, any value supported by the MS
TSPX_Maio8	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma8, default : 1
TSPX_Ma8	MAC	PIXIT	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 2 and 17. default : '002800'O
TSPX_Hsn8	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, default : 40
TSPX_Maio9	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma9, default : 15
TSPX_Ma9	MAC	PIXIT	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 2 and 17. default :

			'01FFFB'O
TSPX_Hsn9	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, default : 8
TSPX_Chtp5	CH_TDMA	PIXIT	SDCCH8 subchannel immediately assigned, TC_26_6_13_4
TSPX_Maio10	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma10, default : 5
TSPX_Ma10	MAC	PIXIT	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 1 and 17. default : '01D082'O
TSPX_Hsn10	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, default: 0
TSPX_Maio11	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma11, default : 11
TSPX_Ma11	MAC	PIXIT	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 2 and 17. default : '00CFF3'O
TSPX_Chtp6	CH_TDMA	PIXIT	channel type of a non- existing channel in Assignment Cmd, TC_26_6_13_4
TSPX_Maio12	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma12, default : 1
TSPX_Ma12	MAC	PIXIT	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 1 and 17. default : '010100'O
TSPX_Hsn12	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, default : 39
TSPX_Maio13	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma13, default : 6
TSPX_Ma13	MAC	PIXIT	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 1 and 17. default : '00E690'O
TSPX_Hsn13	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, default: 42
TSPX_Chtp7	CH_TDMA	PIXIT	channel type for TC_26_6_13_5
TSPX_ChMod4	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_5
TSPX_Maio14	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma14, default : 5
TSPX_Ma14	MAC	PIXIT	mobile allocation for TC_26_6_13_5, its value shall indicate number of frequencies between 1 and 17. default : '014A28'O
TSPX_Hsn14	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, default : 50
TSPX_Maio15	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma15, default : 12
TSPX_Ma15	MAC	PIXIT	mobile allocation for

			TC_26_6_13_5, its value shall indicate number of frequencies between 1 and 16. default : '00FF79'O
TSPX_Hsn15	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, default : 33
TSPX_Maio16	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma16, default : 2
TSPX_Ma16	MAC	PIXIT	mobile allocation for TC_26_6_13_5, its value shall indicate number of frequencies between 1 and 16. default : '001110'O
TSPX_Hsn16	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, default : 21
TSPX_Chtp8	CH_TDMA	PIXIT	channel type for TC_26_6_13_6
TSPX_ChMod5	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_6
TSPX_Maio17	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma17, default : 7
TSPX_Ma17	MAC	PIXIT	mobile allocation for TC_26_6_13_6, its value shall indicate number of frequencies between 1 and 17. default : '00E6A1'O
TSPX_Hsn17	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, default : 22
TSPX_Maio18	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma18, default : 13
TSPX_Ma18	MAC	PIXIT	mobile allocation for TC_26_6_13_6, its value shall indicate number of frequencies between 1 and 16. default : '00FFF9'O
TSPX_Hsn18	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, default : 9
TSPX_Maio19	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma19, default : 3
TSPX_Ma19	MAC	PIXIT	mobile allocation for TC_26_6_13_6, its value shall indicate number of frequencies between 1 and 16. default : '001111'O
TSPX_Hsn19	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, default : 44
TSPX_Chtp9	CH_TDMA	PIXIT	channel type for TC_26_6_13_7
TSPX_ChMod6	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_7
TSPX_Maio20	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma20, default : 8
TSPX_Ma20	MAC	PIXIT	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 17. default : '016699'O
TSPX_Hsn20	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, default : 30
TSPX_Maio21	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in

			TSPX_Ma21, default : 14
TSPX_Ma21	MAC	PIXIT	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 17. default : '01FEFB'O
TSPX_Chtp10	CH_TDMA	PIXIT	channel type for TC_26_6_13_7
TSPX_Maio22	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma22, default : 4
TSPX_Ma22	MAC	PIXIT	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 16. default : '006241'O
TSPX_Hsn22	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, default : 11
TSPX_Maio23	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma23, default : 11
TSPX_Ma23	MAC	PIXIT	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 16. default : '00FF1B'O
TSPX_Hsn23	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, default : 60
TSPX_Chtp11	CH_TDMA	PIXIT	SDCCH8 subchannel immediately assigned, TC_26_6_13_8
TSPX_ChMod7	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_8
TSPX_Maio24	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma24, default : 9
TSPX_Ma24	MAC	PIXIT	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 17. default : '00E6E9'O
TSPX_Hsn24	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, default : 7
TSPX_Maio25	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma25, default : 15
TSPX_Ma25	MAC	PIXIT	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 17. default : '01FFFB'O
TSPX_Chtp12	CH_TDMA	PIXIT	channel type of a non- existing channel in Assignment Cmd, TC_26_6_13_8
TSPX_Maio26	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma26, default : 5
TSPX_Ma26	MAC	PIXIT	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 16. default : '009168'O
TSPX_Hsn26	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, default : 9
TSPX_Maio27	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies)

			-1, which is defined in TSPX_Ma27, default : 1
TSPX_Ma27	MAC	PIXIT	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 16. default : '004080'O
TSPX_Hsn27	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, default : 38
TSPX_Chtp13	CH_TDMA	PIXIT	channel type for TC_26_6_13_9
TSPX_Tm3	INTEGER	PIXIT	channel mode for TC_26_6_13_9, value between 60 -100
TSPX_Maio28	INTEGER	PIXIT	hopping parameter for TC_26_6_13_9, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma28, default : 10
TSPX_Ma28	MAC	PIXIT	mobile allocation for TC_26_6_13_9, its value shall indicate number of frequencies between 1 and 17. default : '00EE7A'O
TSPX_Hsn28	INTEGER	PIXIT	hopping parameter for TC_26_6_13_9, default : 11
TSPX_Maio29	INTEGER	PIXIT	hopping parameter for TC_26_6_13_9, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma29, default : 6
TSPX_Ma29	MAC	PIXIT	mobile allocation for TC_26_6_13_9, its value shall indicate number of frequencies between 1 and 17. default : '00F070'O
TSPX_Chtp14	CH_TDMA	PIXIT	channel type for TC_26_6_13_10
TSPX_Maio30	INTEGER	PIXIT	hopping parameter for TC_26_6_13_10, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma30, default : 11
TSPX_Ma30	MAC	PIXIT	mobile allocation for TC_26_6_13_10, its value shall indicate number of frequencies between 1 and 17. default : '017F69'O
TSPX_Hsn30	INTEGER	PIXIT	hopping parameter for TC_26_6_13_10, default : 62
TSPX_Maio31	INTEGER	PIXIT	hopping parameter for TC_26_6_13_10, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma31, default : 1
TSPX_Ma31	MAC	PIXIT	mobile allocation for TC_26_6_13_10, its value shall indicate number of frequencies between 1 and 17. default : '000101'O
TSPX_MOBscSvcA	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcB	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcC	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcD	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcE	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcF	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcG	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcH	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcI	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcJ	SERVICES	PIXIT	any supported MO basic service
TSPX_MT_DualModSvc	SERVICES	PIXIT	any supported MT dual mode service
TSPX_MT_NonSptSvc	SERVICES	PIXIT	any not supported MT basic service
TSPX_MTBscSvcA	SERVICES	PIXIT	any supported MT basic service

TSPX_MTBscSvcB	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcC	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcD	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcE	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcF	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcG	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcH	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcI	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcJ	SERVICES	PIXIT	any supported MT basic service
TSPX_MTNIC_BscSvcA	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcB	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcC	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcD	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcE	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcF	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcG	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcH	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcI	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcJ	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_Telephony_Rate	RATE	PIXIT	channel rate for TS11, default value : C_Full
TSPX_MOChRateA	RATE	PIXIT	channel rate for TSPX_MOBscSvcA, default value : C_Full
TSPX_MOChRateB	RATE	PIXIT	channel rate for TSPX_MOBscSvcB, default value : C_Full
TSPX_MOChRateC	RATE	PIXIT	channel rate for TSPX_MOBscSvcC, default value : C_Full
TSPX_MOChRateD	RATE	PIXIT	channel rate for TSPX_MOBscSvcD, default value : C_Full
TSPX_MOChRateE	RATE	PIXIT	channel rate for TSPX_MOBscSvcE, default value : C_Full
TSPX_MOChRateF	RATE	PIXIT	channel rate for TSPX_MOBscSvcF, default value : C_Full
TSPX_MOChRateG	RATE	PIXIT	channel rate for TSPX_MOBscSvcG, default value : C_Full
TSPX_MOChRateH	RATE	PIXIT	channel rate for TSPX_MOBscSvcH, default value : C_Full
TSPX_MOChRateI	RATE	PIXIT	channel rate for TSPX_MOBscSvcI, default value : C_Full
TSPX_MOChRateJ	RATE	PIXIT	channel rate for TSPX_MOBscSvcJ, default value : C_Full
TSPX_MTChRateA	RATE	PIXIT	channel rate for TSPX_MTBscSvcA, default value : C_Full
TSPX_MTChRateB	RATE	PIXIT	channel rate for TSPX_MTBscSvcB, default value : C_Full
TSPX_MTChRateC	RATE	PIXIT	channel rate for TSPX_MTBscSvcC, default value : C_Full
TSPX_MTChRateD	RATE	PIXIT	channel rate for

			TSPX_MTBscSvcD, default value : C_Full
TSPX_MTChRateE	RATE	PIXIT	channel rate for TSPX_MTBscSvcE, default value : C_Full
TSPX_MTChRateF	RATE	PIXIT	channel rate for TSPX_MTBscSvcF, default value : C_Full
TSPX_MTChRateG	RATE	PIXIT	channel rate for TSPX_MTBscSvcG, default value : C_Full
TSPX_MTChRateH	RATE	PIXIT	channel rate for TSPX_MTBscSvcH, default value : C_Full
TSPX_MTChRateI	RATE	PIXIT	channel rate for TSPX_MTBscSvcI, default value : C_Full
TSPX_MTChRateJ	RATE	PIXIT	channel rate for TSPX_MTBscSvcJ, default value : C_Full
TSPX_MTNIC_ChRateA	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcA, default value : C_Full
TSPX_MTNIC_ChRateB	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcB, default value : C_Full
TSPX_MTNIC_ChRateC	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcC, default value : C_Full
TSPX_MTNIC_ChRateD	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcD, default value : C_Full
TSPX_MTNIC_ChRateE	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcE, default value : C_Full
TSPX_MTNIC_ChRateF	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcF, default value : C_Full
TSPX_MTNIC_ChRateG	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcG, default value : C_Full
TSPX_MTNIC_ChRateH	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcH, default value : C_Full
TSPX_MTNIC_ChRateI	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcI, default value : C_Full
TSPX_MTNIC_ChRateJ	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcJ, default value : C_Full
TSPX_EmgCallRate	RATE	PIXIT	Rate for the basic service supported for MO emergency calls
TSPX_MO_NonCallSS	SERVICES	PIXIT	any supported MO non-call releated supplementary service
TSPX_MO_BscSvc_SMS	SERVICES	PIXIT	any supported MO SMS calls
TSPX_MO_DualModSvc	SERVICES	PIXIT	any supported MO dual mode call
TSPX_MO_DualModRate	RATE	PIXIT	Rate for the supported MO dual mode call in TSPX_MO_DualModSvc
TSPX_EFR_SpchPrefsNo	INTEGER	PIXIT	Total no. of Speech Versions supported
TSPX_EFR_SpchPref1	B_4	PIXIT	Speech Version supported, No.1
TSPX_EFR_SpchPref2	B_4	PIXIT	Speech Version supported, No.2
TSPX_EFR_SpchPref3	B_4	PIXIT	Speech Version supported, No.3
Detailed Comments:			

Test case selection expression definitions

Test Case Selection Expression Definitions		
Expression Name	**Selection Expression**	**Comments**
SelExpr_0000	(TSPC_FullRateOnly OR TSPC_DualRate) AND TSPC_EFR	
SelExpr_0001	(TSPC_CC AND TSPC_FullRateOnly) AND TSPC_EFR	
SelExpr_0002	TSPC_TeleSvc AND TSPC_EFR	
SelExpr_0003	(TSPC_FullRateSpeech OR TSPC_HalfRateSpeech) AND TSPC_EFR	
SelExpr_0005	TRUE	
Detailed Comments:		

Test suite constant declarations

Test Suite Constant Declarations			
Constant Name	**Type**	**Value**	**Comments**
C_arfcnA	INTEGER	20	ARFCN 20
C_arfcn_1	INTEGER	1	ARFCN 1
C_arfcn_2	INTEGER	2	ARFCN 2
C_arfcn_1015	INTEGER	1015	ARFCN 1015
C_arfcn_10	INTEGER	10	ARFCN 10
C_arfcn_34	INTEGER	34	ARFCN 34
C_arfcnB	INTEGER	10	ARFCN of cell B
C_arfcnC	INTEGER	30	ARFCN 30
C_arfcnH	INTEGER	44	ARFCN 44
C_arfcnAd	INTEGER	590	ARFCN 590
C_arfcnAd_1	INTEGER	520	ARFCN 520
C_arfcnAd_2	INTEGER	514	ARFCN 514
C_arfcnAd_3	INTEGER	715	ARFCN 715
C_arfcnAd_4	INTEGER	747	ARFCN 747
C_arfcnAd_5	INTEGER	734	ARFCN 734
C_arfcnAd_6	INTEGER	759	ARFCN 759
C_arfcnBd	INTEGER	520	ARFCN 520
C_arfcnCd	INTEGER	700	ARFCN 700
C_arfcnHd	INTEGER	810	ARFCN 810
C_arfcn_tchA	INTEGER	30	ARFCN f30
C_arfcn_tchAd	INTEGER	650	ARFCN 650
C_arfcn_tchB	INTEGER	114	ARFCN 114
C_arfcn_tchBd	INTEGER	844	ARFCN 844
C_arfcnEgsm	INTEGER	990	ARFCN 990
C_arfcnEgsm_iacmd	INTEGER	20	ARFCN 20
C_arfcnEgsm_asscmd	INTEGER	20	ARFCN 20
C_BCCHcarrierA_ho	INTEGER	20	BCCH frequency number of cell A for Ho cases (GSM: 20)
C_BCCHcarrierA_hod	INTEGER	747	BCCH frequency number of cell A for Ho cases (DCS:747)
C_BCCHcarrierB_ho	INTEGER	40	BCCH frequency number of cell B for Ho cases (GSM: 40)
C_BCCHcarrierB_hod	INTEGER	764	BCCH frequency number of cell B for Ho cases (DCS:764)
C_Noarfcn	INTEGER	65534	No arfcn defined (hopping channel)
C_ATT_0	INTEGER	0	Attach value 0
C_ATT_1	INTEGER	1	Attach value 1
C_NCC	NCC	'001'B	Network color code
C_NCC_0	NCC	'000'B	Network color code value '000'B
C_NCC_1	NCC	'001'B	Network color code value '001'B
C_NCC_2	NCC	'010'B	Network color code value '010'B
C_NCC_3	NCC	'011'B	Network color code value '011'B
C_NCC_4	NCC	'100'B	Network color code value '100'B
C_NCC_5	NCC	'101'B	Network color code value '101'B
C_NCCP_2	NCCP	'02'O	Network color code permitted value '02'O
C_NCCP_10	NCCP	'0A'O	Network color code permitted value '0A'O
C_BCC	BCC	'101'B	Base Station color code
C_BCC_0	BCC	'000'B	Base Station color code value 0
C_BCC_1	BCC	'001'B	Base Station color code value 1
C_BCC_5	BCC	'101'B	Base Station color code value 5
C_cch_1nonComb	B_3	'000'B	cch_con = 1 basic ch non combined
C_cch_1Comb	B_3	'001'B	cch_con = 1 basic ch combined

C_cch_2Comb	B_3	'011'B	cch_con = 2 basic ch combined
C_cch_3Comb	B_3	'101'B	cch_con = 3 basic ch combined
C_cch_4Comb	B_3	'111'B	cch_con = 4 basic ch combined
C_cch_2nonComb	B_3	'010'B	cch_con = 2 basic chs non combined
C_cch_3nonComb	B_3	'100'B	cch_con = 3 basic chs non combined
C_cch_4nonComb	B_3	'110'B	cch_con = 4 basic chs non combined
C_cch_XnonComb	B_3	'??0'B	cch_con = non combined all number value permitted
C_cch_NotAllowed	B_3	'??1'B	cch_con = value reserved after testing '001'B
C_cchd_e_407	OCTETSTRING	'8DEA0DF4CC6C4AFAFB10000000000000'O	f-list for cellchdescr using of 128 format. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}. Length = 16 TC_26_10_4, k=2, c=1 TC_26.10.6, c=1
C_cchd_e_412	OCTETSTRING	'00020800000007F020000008000080000'O	f-list for cellchdescr using of bit map 0 format. The coded set is {20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114,115} Length = 16 TC_26_10_4, k=2, c=6
C_cchd_e_414	OCTETSTRING	'00000000000000002000002000020000000'O	f-list for cellchdescr using of bit map 0 format. Length = 16 f_list: 30, 50 ,70 TC: 26.10.6 c=6
C_cchd_e_415	OCTETSTRING	'841EEA893EF981438000000000000000'O	f-list for cellchdescr using range 1024 format. The coded set is {0, 30, 40, 66, 80, 1005, 1010, 1015}. Length = 16 TC_26_10_4, k=2, c=4 TC: 26.10.6 c=4
C_ChMod_s	CHMOD_VAL	'00000000'B	Mode : signalling only.
C_ChMod_r	CHMOD_VAL	'00000001'B	Mode : speech full or half rate.
C_ChMod_r2	CHMOD_VAL	'00100001'B	Mode : enhanced speech full or half rate - version 2.
C_ChMod_r3	CHMOD_VAL	'01000001'B	Mode : enhanced speech full or half rate - version 3.
C_ChMod_12k	CHMOD_VAL	'00000011'B	Mode : 12k radio rate.
C_ChMod_6k	CHMOD_VAL	'00001011'B	Mode : 6k radio rate.
C_ChMod_3k	CHMOD_VAL	'00010011'B	Mode : 3.6k radio rate.
C_Sap0	SAPID	'00'O	service access point 0
C_Sap3	SAPID	'03'O	service access point 3
C_CellA	CellID	"C_CellA"	cell A -- cell 1
C_CellB	CellID	"C_CellB"	cell B -- cell 2
C_CellC	CellID	"C_CellC"	cell C -- cell 3
C_CellD	CellID	"C_CellD"	cell D -- cell 4
C_CellE	CellID	"C_CellE"	cell E -- cell 5
C_CellF	CellID	"C_CellF"	cell F -- cell 6
C_CellG	CellID	"C_CellG"	cell G -- cell 7
C_CellH	CellID	"C_CellH"	cell H -- cell 8
C_ci_cellA	CI	'0001'O	Cell Id for cell A
C_ci_cellB	CI	'0002'O	Cell Id for cell B
C_ci_cellC	CI	'0003'O	Cell Id for cell C
C_ci_cellD	CI	'0004'O	Cell Id for cell D
C_ci_cellE	CI	'0005'O	Cell Id for cell E
C_ci_cellF	CI	'0006'O	Cell Id for cell F
C_ci_cellG	CI	'0007'O	Cell Id for cell G
C_ci_cellH	CI	'0008'O	Cell Id for cell H
C_cksn_nokey	BITSTRING	'111'B	No key available
C_E_default	INTEGER	63	Default field strength
C_E_neighbourdefault	INTEGER	53	Default field strength of

			neighbouring cells.
C_E_suitable	INTEGER	45	Field strength of a suitable cell
C_E_notsuitable	INTEGER	-10	Field strength of a not suitable cell for reselection
C_E_notsuitable_fSel	INTEGER	24	Field strength of a not suitable cell for selection
C_flist_e_401	OCTETSTRING	'8DF68AEC00'O	It includes the list of f's in EGSM test cases. The coded set is {1005, 1010, 1015}. range 128 format Length = 5 TC_26_10_4, k=1, c=1 TC: 26.10.5.1 k=1/2,c=1
C_flist_e_402	OCTETSTRING	'8A2481FF03F8'O	It includes the list of f's in EGSM test cases. The coded set is {73, 74, 75, 76, 77}, range 256 format. Length = 6 TC_26_10_4, k=1, c=2 TC: 26.10.5.1 k=1/2,c=2
C_flist_e_403	OCTETSTRING	'89EA00BFC040'O	It includes the list of f's in EGSM test cases. The coded set is {980, 981, 982, 983}, range 512 format. Length = 6 TC_26_10_4, k=1, c=3 TC: 26.10.5.1 k=1/2,c=3
C_flist_e_404	OCTETSTRING	'801EED02BEC0'O	It includes the list of f's in EGSM test cases. The coded set is {30, 40, 1010, 1015}, range 1024 format. Length = 6 TC_26_10_4, k=1, c=4 TC: 26.10.5.1 k=1/2,c=4
C_flist_e_405	OCTETSTRING	'8FEA001E000010'O	It includes the list of f's in EGSM test cases. The coded set is {980, 991, 992, 993, 994, 1015}. Length = 7 TC_26_10_4, k=1, c=5 TC: 26.10.5.1 k=1/2,c=5
C_flist_e_406	OCTETSTRING	'00000000000000020000008000080000'O	It includes the list of f's in EGSM test cases. The coded set is {20, 40, 66}, bitmap 0 format. Length = 16 TC_26_10_4, k=1, c=6 TC: 26.10.5.1 k=2,c=6
C_flist_e_407	OCTETSTRING	'8DEA0DF4CC6C4AFAFB10000000000000'O	It includes the list of f's in EGSM test cases. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}, range 128 format. Length = 16 TC_26_10_4, k=2, c=1 TC: 26.10.5.1 k=3,c=1 TC: 26.10.6 c=1
C_flist_e_408	OCTETSTRING	'8A0A1CFD3EF4610E2FFFFA0000'O	It includes the list of f's in EGSM test cases. The coded set is {20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114, 115}, range 256 format. Length = 13 TC_26_10_4, k=2, c=2 TC: 26.10.5.1 k=3,c=2 TC: 26.10.6 c=2
C_flist_e_409	OCTETSTRING	'89EA037F433C7B042BFEBFEC10000000'O	It includes the list of f's in EGSM test cases. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}, range 512 format. Length = 16 TC_26_10_4, k=2, c=3 TC: 26.10.5.2 k=3,c=3 TC: 26.10.6 c=3

C_flist_e_410	OCTETSTRING	'841EEA893EF98143B161'O	Length = 10 TC: 26.10.5.2 k=3,c=4
C_flist_e_411	OCTETSTRING	'8FEA703E084210'O	It includes the list of f's in EGSM test cases. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}, variable bitmap format. Length = 7 TC_26_10_4, k=2, c=5 TC: 26.10.5.1 k=3,c=5 TC: 26.10.6 c=5
C_flist_e_412	OCTETSTRING	'0002080000007F020000008000080000'O	It includes the list of f's in EGSM test cases. The coded set is {20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114, 115}, bitmap 0 format. Length = 16 TC_26_10_4, k=2, c=6 TC: 26.10.5.1 k=3,c=6
C_flist_e_415	OCTETSTRING	'841EEA893EF9814380'O	It includes the list of f's in EGSM test cases. The coded set is {0, 30, 40, 66, 80, 1005, 1010, 1015}, range 1024 format. Length = 9 Format: 1024 TC_26_10_4, k=2, c=4 TC: 26.10.6 c=4
C_flist_e_416	OCTETSTRING	'0002080000007F020000008000080000'O	bit map 0 format. The coded set is {20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114} Length = 16
C_BABR_0	B_3	'000'B	BABR value 0
C_BABR_1	B_3	'001'B	BABR value 1
C_BABR_2	B_3	'010'B	BABR value 2
C_BPM_0	B_3	'000'B	BPM value 0, 2 multiframes
C_BPM_2	B_3	'010'B	BPM value 2, 4 multiframes
C_BPM_3	B_3	'011'B	BPM value 3, 5 multiframes
C_BPM_7	B_3	'111'B	BPM value 7, 9 multiframes
C_T3212_0	OCTETSTRING	'00'O	T3212 infinite
C_T3212_1	OCTETSTRING	'01'O	T3212 6 mn / 1 deci hours
C_T3212_2	OCTETSTRING	'02'O	T3212 12 mn / 2 deci hours
C_T3212_5	OCTETSTRING	'05'O	T3212 30 mn / 5 deci hours
C_IMSI	INTEGER	0	MI type: IMSI
C_TMSI	INTEGER	1	MI type: TMSI
C_IMEI	INTEGER	2	MI type: IMEI
C_IMEISV	INTEGER	3	MI type: IMEISV
C_shortIMSI	HEXSTRING	'001011234'H	A short IMSI
C_LAC_1	OCTETSTRING	'0001'O	lac 1.
C_LAC_2	OCTETSTRING	'0002'O	lac 2.
C_LAC_3	OCTETSTRING	'0003'O	lac 3.
C_LAC_4	OCTETSTRING	'0004'O	lac 4.
C_LAC_5	OCTETSTRING	'0005'O	lac 5.
C_LAC_6	OCTETSTRING	'0006'O	lac 6.
C_LAC_7	OCTETSTRING	'0007'O	lac 7.
C_LAC_8	OCTETSTRING	'0008'O	lac 8.
C_LAC_spe	OCTETSTRING	'5344'O	lac '5344'H.
C_lacdeleted	OCTETSTRING	'FFFE'O	lac value deleted.
C_MCC_1	OCTETSTRING	'00F1'O	MCC 1
C_MCC_2	OCTETSTRING	'00F2'O	MCC 2
C_MCC_3	OCTETSTRING	'00F3'O	MCC 3
C_MCC_4	OCTETSTRING	'00F4'O	MCC 4
C_MCC_5	OCTETSTRING	'00F5'O	MCC 5
C_MCC_6	OCTETSTRING	'00F6'O	MCC 6
C_MCC_7	OCTETSTRING	'00F7'O	MCC 7
C_MCC_8	OCTETSTRING	'00F8'O	MCC 8
C_MCC_9	OCTETSTRING	'013B'O	MCC 9
C_Max_1	INTEGER	1	Max Retrans value 1
C_Max_2	INTEGER	2	Max Retrans value 2

C_Max_7	INTEGER	7	Max Retrans value 7
C_NECI_0	INTEGER	0	NECI value 0, not supported
C_NECI_1	INTEGER	1	NECI value 1, supported
C_PLMN_1	OCTETSTRING	'10'O	PLMN 1 (mnc=01)
C_PLMN_2	OCTETSTRING	'20'O	PLMN 2 (mnc=02)
C_PLMN_3	OCTETSTRING	'30'O	PLMN 3 (mnc=03)
C_PLMN_4	OCTETSTRING	'F0'O	PLMN 4
C_PLMN_5	OCTETSTRING	'F2'O	PLMN 5
C_PLMN_6	OCTETSTRING	'F3'O	PLMN 6
C_PLMN_7	OCTETSTRING	'F4'O	PLMN 7
C_PLMN_8	OCTETSTRING	'F5'O	PLMN 8
C_PLMN_9	OCTETSTRING	'F6'O	PLMN 9
C_PLMN_10	OCTETSTRING	'F7'O	PLMN 10
C_PLMN_Home	OCTETSTRING	'10'O	PLMN 1 (mnc=01)
C_TxInt_5	INTEGER	5	Tx Integer value 5
C_TxInt_7	INTEGER	7	Tx Integer value 7
C_TxInt_10	INTEGER	10	Tx Integer value 10
C_NotCombined	BOOLEAN	FALSE	CCCH not combined with SDCCH
C_Combined	BOOLEAN	TRUE	CCCH combined with SDCCH
C_FCCH_A	LOGICCH	"C_FCCH_A"	frequency correction channel of cell A(1)
C_FCCH_B	LOGICCH	"C_FCCH_B"	frequency correction channel of cell B(2)
C_FCCH_C	LOGICCH	"C_FCCH_C"	frequency correction channel of cell C(3)
C_FCCH_D	LOGICCH	"C_FCCH_D"	frequency correction channel of cell D(4)
C_FCCH_E	LOGICCH	"C_FCCH_E"	frequency correction channel of cell E(5)
C_FCCH_F	LOGICCH	"C_FCCH_F"	frequency correction channel of cell F(6)
C_FCCH_G	LOGICCH	"C_FCCH_G"	frequency correction channel of cell G(7)
C_FCCH_H	LOGICCH	"C_FCCH_H"	frequency correction channel of cell H(8)
C_SCH_A	LOGICCH	"C_SCH_A"	sync channel of cell A(1)
C_SCH_B	LOGICCH	"C_SCH_B"	sync channel of cell B(2)
C_SCH_C	LOGICCH	"C_SCH_C"	sync channel of cell C(3)
C_SCH_D	LOGICCH	"C_SCH_D"	sync channel of cell D(4)
C_SCH_E	LOGICCH	"C_SCH_E"	sync channel of cell E(5)
C_SCH_F	LOGICCH	"C_SCH_F"	sync channel of cell F(6)
C_SCH_G	LOGICCH	"C_SCH_G"	sync channel of cell G(7)
C_SCH_H	LOGICCH	"C_SCH_H"	sync channel of cell H(8)
C_BCCH_A_1	LOGICCH	"C_BCCH_A_1"	1st broadcast channel of cell A
C_BCCH_A_2	LOGICCH	"C_BCCH_A_2"	2nd broadcast channel of cell A
C_BCCH_A_3	LOGICCH	"C_BCCH_A_3"	3rd broadcast channel of cell A
C_BCCH_A_4	LOGICCH	"C_BCCH_A_4"	4th broadcast channel of cell A
C_BCCH_B_1	LOGICCH	"C_BCCH_B_1"	1st broadcast channel of cell B
C_BCCH_C_1	LOGICCH	"C_BCCH_C_1"	1st broadcast channel of cell C
C_BCCH_D_1	LOGICCH	"C_BCCH_D_1"	1st broadcast channel of cell D
C_BCCH_E_1	LOGICCH	"C_BCCH_E_1"	1st broadcast channel of cell E
C_BCCH_F_1	LOGICCH	"C_BCCH_F_1"	1st broadcast channel of cell F
C_BCCH_G_1	LOGICCH	"C_BCCH_G_1"	1st broadcast channel of cell G
C_BCCH_H_1	LOGICCH	"C_BCCH_H_1"	1st broadcast channel of cell H
C_CBCH_A	LOGICCH	"C_CBCH_A"	Cell broadcast channel in cell A
C_AGCH_A_1	LOGICCH	"C_AGCH_A_1"	1st down link CCCH (AGCH) of cell A
C_AGCH_A_2	LOGICCH	"C_AGCH_A_2"	2nd down link CCCH (AGCH) of cell A
C_AGCH_A_3	LOGICCH	"C_AGCH_A_3"	3rd down link CCCH (AGCH) of cell A
C_AGCH_A_4	LOGICCH	"C_AGCH_A_4"	4th down link CCCH (AGCH) of cell A
C_AGCH_B_1	LOGICCH	"C_AGCH_B_1"	1st down link CCCH (AGCH) of cell B
C_AGCH_B_2	LOGICCH	"C_AGCH_B_2"	2nd down link CCCH (AGCH) of cell B

C_AGCH_B_3	LOGICCH	"C_AGCH_B_3"	3rd down link CCCH (AGCH) of cell B
C_AGCH_B_4	LOGICCH	"C_AGCH_B_4"	4th down link CCCH (AGCH) of cell B
C_AGCH_C_1	LOGICCH	"C_AGCH_C_1"	1st down link CCCH (AGCH) of cell C
C_AGCH_C_2	LOGICCH	"C_AGCH_C_2"	2nd down link CCCH (AGCH) of cell C
C_AGCH_C_3	LOGICCH	"C_AGCH_C_3"	3rd down link CCCH (AGCH) of cell C
C_AGCH_C_4	LOGICCH	"C_AGCH_C_4"	4th down link CCCH (AGCH) of cell C
C_AGCH_D_1	LOGICCH	"C_AGCH_D_1"	1st down link CCCH (AGCH) of cell D
C_AGCH_E_1	LOGICCH	"C_AGCH_E_1"	1st down link CCCH (AGCH) of cell E
C_AGCH_F_1	LOGICCH	"C_AGCH_F_1"	1st down link CCCH (AGCH) of cell F
C_AGCH_G_1	LOGICCH	"C_AGCH_G_1"	1st down link CCCH (AGCH) of cell G
C_AGCH_H_1	LOGICCH	"C_AGCH_H_1"	1st down link CCCH (AGCH) of cell H
C_PCH_A_1	LOGICCH	"C_PCH_A_1"	1st down link CCCH (PCH) of cell A
C_PCH_A_2	LOGICCH	"C_PCH_A_2"	2nd down link CCCH (PCH) of cell A
C_PCH_A_3	LOGICCH	"C_PCH_A_3"	3rd down link CCCH (PCH) of cell A
C_PCH_A_4	LOGICCH	"C_PCH_A_4"	4th down link CCCH (PCH) of cell A
C_PCH_B_1	LOGICCH	"C_PCH_B_1"	1st down link CCCH (PCH) of cell B
C_PCH_B_2	LOGICCH	"C_PCH_B_2"	2nd down link CCCH (PCH) of cell B
C_PCH_B_3	LOGICCH	"C_PCH_B_3"	3rd down link CCCH (PCH) of cell B
C_PCH_B_4	LOGICCH	"C_PCH_B_4"	4th down link CCCH (PCH) of cell B
C_PCH_C_1	LOGICCH	"C_PCH_C_1"	1st down link CCCH (PCH) of cell C
C_PCH_C_2	LOGICCH	"C_PCH_C_2"	2nd down link CCCH (PCH) of cell C
C_PCH_C_3	LOGICCH	"C_PCH_C_3"	3rd down link CCCH (PCH) of cell C
C_PCH_C_4	LOGICCH	"C_PCH_C_4"	4th down link CCCH (PCH) of cell C
C_PCH_D_1	LOGICCH	"C_PCH_D_1"	1st down link CCCH (PCH and AGCH) of cell D
C_PCH_E_1	LOGICCH	"C_PCH_E_1"	1st down link CCCH (PCH) of cell E
C_PCH_F_1	LOGICCH	"C_PCH_F_1"	1st down link CCCH (PCH) of cell F
C_PCH_G_1	LOGICCH	"C_PCH_G_1"	1st down link CCCH (PCH) of cell G
C_PCH_H_1	LOGICCH	"C_PCH_H_1"	1st down link CCCH (PCH) of cell H
C_RACH_A_1	LOGICCH	"C_RACH_A_1"	1st uplink CCCH (RACH) of cell A
C_RACH_A_2	LOGICCH	"C_RACH_A_2"	2nd uplink CCCH (RACH) of cell A
C_RACH_A_3	LOGICCH	"C_RACH_A_3"	3rd uplink CCCH (RACH) of cell A
C_RACH_A_4	LOGICCH	"C_RACH_A_4"	4th uplink CCCH (RACH) of cell A
C_RACH_B_1	LOGICCH	"C_RACH_B_1"	1st uplink CCCH (RACH) of cell B
C_RACH_B_2	LOGICCH	"C_RACH_B_1"	2nd uplink CCCH (RACH) of cell B
C_RACH_B_3	LOGICCH	"C_RACH_B_3"	3rd uplink CCCH (RACH) of cell B
C_RACH_B_4	LOGICCH	"C_RACH_B_4"	4th uplink CCCH (RACH) of cell B
C_RACH_C_1	LOGICCH	"C_RACH_C_1"	1st uplink CCCH (RACH) of cell C
C_RACH_C_2	LOGICCH	"C_RACH_C_2"	2nd uplink CCCH (RACH) of cell C
C_RACH_C_3	LOGICCH	"C_RACH_C_3"	3rd uplink CCCH (RACH) of cell C

C_RACH_C_4	LOGICCH	"C_RACH_C_4"	4th uplink CCCH (RACH) of cell C
C_RACH_D_1	LOGICCH	"C_RACH_D_1"	1st uplink CCCH (RACH) of cell D
C_RACH_E_1	LOGICCH	"C_RACH_E_1"	1st uplink CCCH (RACH) of cell E
C_RACH_F_1	LOGICCH	"C_RACH_F_1"	1st uplink CCCH (RACH) of cell F
C_RACH_G_1	LOGICCH	"C_RACH_G_1"	1st uplink CCCH (RACH) of cell G
C_RACH_H_1	LOGICCH	"C_RACH_H_1"	1st uplink CCCH (RACH) of cell H
C_FACCHF_A_1	LOGICCH	"C_FACCHF_A_1"	FACCH associated with 1st TCH/F of cell A
C_FACCHF_A_2	LOGICCH	"C_FACCHF_A_2"	FACCH associated with 2nd TCH/F of cell A
C_FACCHF_A_3	LOGICCH	"C_FACCHF_A_3"	FACCH associated with 3rd TCH/F of cell A
C_FACCHF_B_1	LOGICCH	"C_FACCHF_B_1"	FACCH associated with 1st TCH/F of cell B
C_FACCHF_B_2	LOGICCH	"C_FACCHF_B_2"	FACCH associated with 2nd TCH/F of cell B
C_FACCHF_B_3	LOGICCH	"C_FACCHF_B_3"	FACCH associated with 3rd TCH/F of cell B
C_FACCHF_C_1	LOGICCH	"C_FACCHF_C_1"	FACCH associated with 1sd TCH/F of cell C
C_FACCHF_C_2	LOGICCH	"C_FACCHF_C_2"	FACCH associated with 2nd TCH/F of cell C
C_FACCHF_C_3	LOGICCH	"C_FACCHF_C_3"	FACCH associated with 3rd TCH/F of cell C
C_FACCHF_H_1	LOGICCH	"C_FACCHF_H_1"	FACCH associated with 1st TCH/F of cell H
C_FACCHH_A_1	LOGICCH	"C_FACCHH_A_1"	FACCH associated with 1st TCH/H of cell A
C_FACCHH_A_2	LOGICCH	"C_FACCHH_A_2"	FACCH associated with 2nd TCH/H of cell A
C_FACCHH_A_3	LOGICCH	"C_FACCHH_A_3"	FACCH associated with 3rd TCH/H of cell A
C_FACCHH_B_1	LOGICCH	"C_FACCHH_B_1"	FACCH associated with 1st TCH/H of cell B
C_FACCHH_B_2	LOGICCH	"C_FACCHH_B_2"	FACCH associated with 2nd TCH/H of cell B
C_FACCHH_B_3	LOGICCH	"C_FACCHH_B_3"	FACCH associated with 3rd TCH/H of cell B
C_FACCHH_C_1	LOGICCH	"C_FACCHH_C_1"	FACCH associated with 1st TCH/H of cell C
C_FACCHH_C_2	LOGICCH	"C_FACCHH_C_2"	FACCH associated with 2nd TCH/H of cell C
C_FACCHH_C_3	LOGICCH	"C_FACCHH_C_3"	FACCH associated with 3rd TCH/H of cell C
C_FACCHH0_A_1	LOGICCH	"C_FACCHH0_A_1"	FACCH associated with 1st TCH/H0 of cell A
C_FACCHH1_A_1	LOGICCH	"C_FACCHH1_A_1"	FACCH associated with 1st TCH/H1 of cell A
C_FACCHH0_A_2	LOGICCH	"C_FACCHH0_A_2"	FACCH associated with 2nd TCH/H0 of cell A
C_FACCHH1_A_2	LOGICCH	"C_FACCHH1_A_2"	FACCH associated with 2nd TCH/H1 of cell A
C_FACCHH0_A_3	LOGICCH	"C_FACCHH0_A_3"	FACCH associated with 3rd TCH/H0 of cell A
C_FACCHH1_A_3	LOGICCH	"C_FACCHH1_A_3"	FACCH associated with 3rd TCH/H1 of cell A
C_FACCHH0_B_1	LOGICCH	"C_FACCHH0_B_1"	FACCH associated with 1st TCH/H0 of cell B
C_FACCHH1_B_1	LOGICCH	"C_FACCHH1_B_1"	FACCH associated with 1st TCH/H1 of cell B
C_FACCHH0_B_2	LOGICCH	"C_FACCHH0_B_2"	FACCH associated with 2nd TCH/H0 of cell B
C_FACCHH1_B_2	LOGICCH	"C_FACCHH1_B_2"	FACCH associated with 2nd TCH/H1 of cell B
C_FACCHH0_B_3	LOGICCH	"C_FACCHH0_B_3"	FACCH associated with 3rd TCH/H0 of cell B
C_FACCHH1_B_3	LOGICCH	"C_FACCHH1_B_3"	FACCH associated with 3rd TCH/H1 of cell B
C_FACCHH0_C_1	LOGICCH	"C_FACCHH0_C_1"	FACCH associated with 1st TCH/H0 of cell C

C_FACCHH1_C_1	LOGICCH	"C_FACCHH1_C_1"	FACCH associated with 1st TCH/H1 of cell C
C_FACCHH0_C_2	LOGICCH	"C_FACCHH0_C_2"	FACCH associated with 2nd TCH/H0 of cell C
C_FACCHH1_C_2	LOGICCH	"C_FACCHH1_C_2"	FACCH associated with 2nd TCH/H1 of cell C
C_FACCHH0_C_3	LOGICCH	"C_FACCHH0_C_3"	FACCH associated with 3rd TCH/H0 of cell C
C_FACCHH1_C_3	LOGICCH	"C_FACCHH1_C_3"	FACCH associated with 3rd TCH/H1 of cell C
C_SACCH_A	LOGICCH	"C_SACCH_A"	all SACCHs of cell A
C_SACCH_B	LOGICCH	"C_SACCH_B"	all SACCHs of cell B
C_SACCH_C	LOGICCH	"C_SACCH_C"	all SACCHs of cell C
C_SACCH_D	LOGICCH	"C_SACCH_D"	all SACCHs of cell D
C_SACCH_E	LOGICCH	"C_SACCH_E"	all SACCHs of cell E
C_SACCH_F	LOGICCH	"C_SACCH_F"	all SACCHs of cell F
C_SACCH_G	LOGICCH	"C_SACCH_G"	all SACCHs of cell G
C_SACCH_H	LOGICCH	"C_SACCH_H"	all SACCHs of cell H
C_SACCHF_A_1	LOGICCH	"C_SACCHF_A_1"	SACCH associated with 1st TCH/F of cell A
C_SACCHF_A_2	LOGICCH	"C_SACCHF_A_2"	SACCH associated with 2nd TCH/F of cell A
C_SACCHF_B_1	LOGICCH	"C_SACCHF_B_1"	SACCH associated with 1st TCH/F of cell B
C_SACCHF_B_2	LOGICCH	"C_SACCHF_B_2"	SACCH associated with 2nd TCH/F of cell B
C_SACCHF_H_1	LOGICCH	"C_SACCHF_H_1"	SACCH associated with 1st TCH/F of cell H
C_SACCHH_A_1	LOGICCH	"C_SACCHH_A_1"	SACCH associated with 1st TCH/H of cell A
C_SACCHH_A_2	LOGICCH	"C_SACCHH_A_2"	SACCH associated with 2nd TCH/H of cell A
C_SACCHH_A_3	LOGICCH	"C_SACCHH_A_3"	SACCH associated with 3rd TCH/H of cell A
C_SACCHH_B_1	LOGICCH	"C_SACCHH_B_1"	SACCH associated with 1st TCH/H of cell B
C_SACCHH_B_2	LOGICCH	"C_SACCHH_B_2"	SACCH associated with 2nd TCH/H of cell B
C_SACCHH_B_3	LOGICCH	"C_SACCHH_B_3"	SACCH associated with 3rd TCH/H of cell B
C_SACCHH_C_1	LOGICCH	"C_SACCHH_C_1"	SACCH associated with 1st TCH/H of cell C
C_SACCHH_C_2	LOGICCH	"C_SACCHH_C_2"	SACCH associated with 2nd TCH/H of cell C
C_SACCHH_C_3	LOGICCH	"C_SACCHH_C_3"	SACCH associated with 3rd TCH/H of cell C
C_SACCHH0_A_1	LOGICCH	"C_SACCHH0_A_1"	SACCH associated with 1st TCH/H_0 of cell A
C_SACCHH1_A_1	LOGICCH	"C_SACCHH1_A_1"	SACCH associated with 1st TCH/H_1 of cell A
C_SACCHH0_A_2	LOGICCH	"C_SACCHH0_A_2"	SACCH associated with 2nd TCH/H_0 of cell A
C_SACCHH1_A_2	LOGICCH	"C_SACCHH1_A_2"	SACCH associated with 2nd TCH/H_1 of cell A
C_SACCHH0_A_3	LOGICCH	"C_SACCHH0_A_3"	SACCH associated with 3rd TCH/H_0 of cell A
C_SACCHH1_A_3	LOGICCH	"C_SACCHH1_A_3"	SACCH associated with 3rd TCH/H_1 of cell A
C_SACCHH0_B_1	LOGICCH	"C_SACCHH0_B_1"	SACCH associated with 1st TCH/H_0 of cell B
C_SACCHH1_B_1	LOGICCH	"C_SACCHH1_B_1"	SACCH associated with 1st TCH/H_1 of cell B
C_SACCHH0_B_2	LOGICCH	"C_SACCHH0_B_2"	SACCH associated with 2nd TCH/H_0 of cell B
C_SACCHH1_B_2	LOGICCH	"C_SACCHH1_B_2"	SACCH associated with 2nd TCH/H_1 of cell B
C_SACCHH0_B_3	LOGICCH	"C_SACCHH0_B_3"	SACCH associated with 3rd TCH/H_0 of cell B
C_SACCHH1_B_3	LOGICCH	"C_SACCHH1_B_3"	SACCH associated with 3rd TCH/H_1 of cell B

C_SACCHH0_C_1	LOGICCH	"C_SACCHH0_C_1"	SACCH associated with 1st TCH/H_0 of cell C
C_SACCHH1_C_1	LOGICCH	"C_SACCHH1_C_1"	SACCH associated with 1st TCH/H_1 of cell C
C_SACCHH0_C_2	LOGICCH	"C_SACCHH0_C_2"	SACCH associated with 2nd TCH/H_0 of cell C
C_SACCHH1_C_2	LOGICCH	"C_SACCHH1_C_2"	SACCH associated with 2nd TCH/H_1 of cell C
C_SACCHH0_C_3	LOGICCH	"C_SACCHH0_C_3"	SACCH associated with 3rd TCH/H_0 of cell C
C_SACCHH1_C_3	LOGICCH	"C_SACCHH1_C_3"	SACCH associated with 3rd TCH/H_1 of cell C
C_SACCHC4_A	LOGICCH	"C_SACCHC4_A"	SACCHC4 channel of cell A
C_SACCHC4_B	LOGICCH	"C_SACCHC4_B"	SACCHC4 channel of cell B
C_SACCHC4_C	LOGICCH	"C_SACCHC4_C"	SACCHC4 channel of cell C
C_SACCHC4_D	LOGICCH	"C_SACCHC4_D"	SACCHC4 channel of cell D
C_SACCHC4_E	LOGICCH	"C_SACCHC4_E"	SACCHC4 channel of cell E
C_SACCHC4_F	LOGICCH	"C_SACCHC4_F"	SACCHC4 channel of cell F
C_SACCHC4_G	LOGICCH	"C_SACCHC4_G"	SACCHC4 channel of cell G
C_SACCHC4_H	LOGICCH	"C_SACCHC4_H"	SACCHC4 channel of cell H
C_SACCHC40_A	LOGICCH	"C_SACCHC40_A"	SACCH/C4(0) associated with SDCCH/4(0) of cell A
C_SACCHC41_A	LOGICCH	"C_SACCHC41_A"	SACCH/C4(1) associated with SDCCH/4(1) of cell A
C_SACCHC42_A	LOGICCH	"C_SACCHC42_A"	SACCH/C4(2) associated with SDCCH/4(2) of cell A
C_SACCHC43_A	LOGICCH	"C_SACCCH43_A"	SACCH/C4(3) associated with SDCCH/4(3) of cell A
C_SACCHC40_B	LOGICCH	"C_SACCHC40_B"	SACCH/C4(0) associated with SDCCH/4(0) of cell B
C_SACCHC41_B	LOGICCH	"C_SACCHC41_B"	SACCH/C4(1) associated with SDCCH/4(1) of cell B
C_SACCHC42_B	LOGICCH	"C_SACCHC42_B"	SACCH/C4(2) associated with SDCCH/4(2) of cell B
C_SACCHC43_B	LOGICCH	"C_SACCHC43_B"	SACCH/C4(3) associated with SDCCH/4(3) of cell B
C_SACCHC40_C	LOGICCH	"C_SACCHC40_C"	SACCH/C4(0) associated with SDCCH/4(0) of cell C
C_SACCHC41_C	LOGICCH	"C_SACCHC41_C"	SACCH/C4(1) associated with SDCCH/4(1) of cell C
C_SACCHC42_C	LOGICCH	"C_SACCHC42_C"	SACCH/C4(2) associated with SDCCH/4(2) of cell C
C_SACCHC43_C	LOGICCH	"C_SACCHC43_C"	SACCH/C4(3) associated with SDCCH/4(3) of cell C
C_SACCHC40_D	LOGICCH	"C_SACCHC40_D"	SACCH/C4(0) associated with SDCCH/4(0) of cell D
C_SACCHC41_D	LOGICCH	"C_SACCHC41_D"	SACCH/C4(1) associated with SDCCH/4(1) of cell D
C_SACCHC42_D	LOGICCH	"C_SACCHC42_D"	SACCH/C4(2) associated with SDCCH/4(2) of cell D
C_SACCHC43_D	LOGICCH	"C_SACCCH43_D"	SACCH/C4(3) associated with SDCCH/4(3) of cell D
C_SACCHC40_E	LOGICCH	"C_SACCHC40_E"	SACCH/C4(0) associated with SDCCH/4(0) of cell E
C_SACCHC41_E	LOGICCH	"C_SACCHC41_E"	SACCH/C4(1) associated with SDCCH/4(1) of cell E
C_SACCHC42_E	LOGICCH	"C_SACCHC42_E"	SACCH/C4(2) associated with SDCCH/4(2) of cell E
C_SACCHC43_E	LOGICCH	"C_SACCHC43_E"	SACCH/C4(3) associated with SDCCH/4(3) of cell E
C_SACCHC40_F	LOGICCH	"C_SACCHC40_F"	SACCH/C4(0) associated with SDCCH/4(0) of cell F
C_SACCHC41_F	LOGICCH	"C_SACCHC41_F"	SACCH/C4(1) associated with SDCCH/4(1) of cell F
C_SACCHC42_F	LOGICCH	"C_SACCHC42_F"	SACCH/C4(2) associated with SDCCH/4(2) of cell F
C_SACCHC43_F	LOGICCH	"C_SACCHC43_F"	SACCH/C4(3) associated with SDCCH/4(3) of cell F
C_SACCHC40_G	LOGICCH	"C_SACCHC40_G"	SACCH/C4(0) associated with SDCCH/4(0) of cell G

C_SACCHC41_G	LOGICCH	"C_SACCHC41_G"	SACCH/C4(1) associated with SDCCH/4(1) of cell G
C_SACCHC42_G	LOGICCH	"C_SACCHC42_G"	SACCH/C4(2) associated with SDCCH/4(2) of cell G
C_SACCHC43_G	LOGICCH	"C_SACCHC43_G"	SACCH/C4(3) associated with SDCCH/4(3) of cell G
C_SACCHC40_H	LOGICCH	"C_SACCHC40_C"	SACCH/C4(0) associated with SDCCH/4(0) of cell H
C_SACCHC41_H	LOGICCH	"C_SACCHC41_C"	SACCH/C4(1) associated with SDCCH/4(1) of cell H
C_SACCHC42_H	LOGICCH	"C_SACCHC42_C"	SACCH/C4(2) associated with SDCCH/4(2) of cell H
C_SACCHC43_H	LOGICCH	"C_SACCHC43_C"	SACCH/C4(3) associated with SDCCH/4(3) of cell H
C_SACCHC8_A_1	LOGICCH	"C_SACCHC8_A_1"	1st SACCH/8 channel of cell A
C_SACCHC8_A_2	LOGICCH	"C_SACCHC8_A_2"	2nd SACCH/8 channel of cell A
C_SACCHC8_A_3	LOGICCH	"C_SACCHC8_A_3"	3rd SACCH/8 channel of cell A
C_SACCHC8_B_1	LOGICCH	"C_SACCHC8_B_1"	1st SACCH/8 channel of cell B
C_SACCHC8_B_2	LOGICCH	"C_SACCHC8_B_2"	2nd SACCH/8 channel of cell B
C_SACCHC8_B_3	LOGICCH	"C_SACCHC8_B_3"	3rd SACCH/8 channel of cell B
C_SACCHC8_C_1	LOGICCH	"C_SACCHC8_C_1"	1st SACCH/8 channel of cell C
C_SACCHC8_C_2	LOGICCH	"C_SACCHC8_C_2"	2nd SACCH/8 channel of cell C
C_SACCHC8_C_3	LOGICCH	"C_SACCHC8_C_3"	3rd SACCH/8 channel of cell C
C_SACCHC80_A_1	LOGICCH	"C_SACCHC80_A_1"	SACCH/C8(0) associated with 1st SDCCH/8(0) of cell A
C_SACCHC81_A_1	LOGICCH	"C_SACCHC81_A_1"	SACCH/C8(1) associated with 1st SDCCH/8(1) of cell A
C_SACCHC82_A_1	LOGICCH	"C_SACCHC82_A_1"	SACCH/C8(2) associated with 1st SDCCH/8(2) of cell A
C_SACCHC83_A_1	LOGICCH	"C_SACCHC83_A_1"	SACCH/C8(3) associated with 1st SDCCH/8(3) of cell A
C_SACCHC84_A_1	LOGICCH	"C_SACCHC84_A_1"	SACCH/C8(4) associated with 1st SDCCH/8(4) of cell A
C_SACCHC85_A_1	LOGICCH	"C_SACCHC85_A_1"	SACCH/C8(5) associated with 1st SDCCH/8(5) of cell A
C_SACCHC86_A_1	LOGICCH	"C_SACCHC86_A_1"	SACCH/C8(6) associated with 1st SDCCH/8(6) of cell A
C_SACCHC87_A_1	LOGICCH	"C_SACCHC87_A_1"	SACCH/C8(7) associated with 1st SDCCH/8(7) of cell A
C_SACCHC80_A_2	LOGICCH	"C_SACCHC80_A_2"	SACCH/C8(0) associated with 2nd SDCCH/8(0) of cell A
C_SACCHC81_A_2	LOGICCH	"C_SACCHC81_A_2"	SACCH/C8(1) associated with 2nd SDCCH/8(1) of cell A
C_SACCHC82_A_2	LOGICCH	"C_SACCHC82_A_2"	SACCH/C8(2) associated with 2nd SDCCH/8(2) of cell A
C_SACCHC83_A_2	LOGICCH	"C_SACCHC83_A_2"	SACCH/C8(3) associated with 2nd SDCCH/8(3) of cell A
C_SACCHC84_A_2	LOGICCH	"C_SACCHC84_A_2"	SACCH/C8(4) associated with 2nd SDCCH/8(4) of cell A
C_SACCHC85_A_2	LOGICCH	"C_SACCHC85_A_2"	SACCH/C8(5) associated with 2nd SDCCH/8(5) of cell A
C_SACCHC86_A_2	LOGICCH	"C_SACCHC86_A_2"	SACCH/C8(6) associated with 2nd SDCCH/8(6) of cell A
C_SACCHC87_A_2	LOGICCH	"C_SACCHC87_A_2"	SACCH/C8(7) associated with 2nd SDCCH/8(7) of cell A
C_SACCHC80_A_3	LOGICCH	"C_SACCHC80_A_3"	SACCH/C8(0) associated with 3rd SDCCH/8(0) of cell A
C_SACCHC81_A_3	LOGICCH	"C_SACCHC81_A_3"	SACCH/C8(1) associated with 3rd SDCCH/8(1) of cell A
C_SACCHC82_A_3	LOGICCH	"C_SACCHC82_A_3"	SACCH/C8(2) associated with 3rd SDCCH/8(2) of cell A
C_SACCHC83_A_3	LOGICCH	"C_SACCHC83_A_3"	SACCH/C8(3) associated with 3rd SDCCH/8(3) of cell A
C_SACCHC84_A_3	LOGICCH	"C_SACCHC84_A_3"	SACCH/C8(4) associated with 3rd SDCCH/8(4) of cell A
C_SACCHC85_A_3	LOGICCH	"C_SACCHC85_A_3"	SACCH/C8(5) associated with 3rd SDCCH/8(5) of cell A
C_SACCHC86_A_3	LOGICCH	"C_SACCHC86_A_3"	SACCH/C8(6) associated with 3rd SDCCH/8(6) of cell A
C_SACCHC87_A_3	LOGICCH	"C_SACCHC87_A_3"	SACCH/C8(7) associated with 3rd

			SDCCH/8(7) of cell A
C_SACCHC80_B_1	LOGICCH	"C_SACCHC80_B_1"	SACCH/C8(0) associated with 1st SDCCH/8(0) of cell B
C_SACCHC81_B_1	LOGICCH	"C_SACCHC81_B_1"	SACCH/C8(1) associated with 1st SDCCH/8(1) of cell B
C_SACCHC82_B_1	LOGICCH	"C_SACCHC82_B_1"	SACCH/C8(2) associated with 1st SDCCH/8(2) of cell B
C_SACCHC83_B_1	LOGICCH	"C_SACCHC83_B_1"	SACCH/C8(3) associated with 1st SDCCH/8(3) of cell B
C_SACCHC84_B_1	LOGICCH	"C_SACCHC84_B_1"	SACCH/C8(4) associated with 1st SDCCH/8(4) of cell B
C_SACCHC85_B_1	LOGICCH	"C_SACCHC85_B_1"	SACCH/C8(5) associated with 1st SDCCH/8(5) of cell B
C_SACCHC86_B_1	LOGICCH	"C_SACCHC86_B_1"	SACCH/C8(6) associated with 1st SDCCH/8(6) of cell B
C_SACCHC87_B_1	LOGICCH	"C_SACCHC87_B_1"	SACCH/C8(7) associated with 1st SDCCH/8(7) of cell B
C_SACCHC80_B_2	LOGICCH	"C_SACCHC80_B_2"	SACCH/C8(0) associated with 2nd SDCCH/8(0) of cell B
C_SACCHC81_B_2	LOGICCH	"C_SACCHC81_B_2"	SACCH/C8(1) associated with 2nd SDCCH/8(1) of cell B
C_SACCHC82_B_2	LOGICCH	"C_SACCHC82_B_2"	SACCH/C8(2) associated with 2nd SDCCH/8(2) of cell B
C_SACCHC83_B_2	LOGICCH	"C_SACCHC83_B_2"	SACCH/C8(3) associated with 2nd SDCCH/8(3) of cell B
C_SACCHC84_B_2	LOGICCH	"C_SACCHC84_B_2"	SACCH/C8(4) associated with 2nd SDCCH/8(4) of cell B
C_SACCHC85_B_2	LOGICCH	"C_SACCHC85_B_2"	SACCH/C8(5) associated with 2nd SDCCH/8(5) of cell B
C_SACCHC86_B_2	LOGICCH	"C_SACCHC86_B_2"	SACCH/C8(6) associated with 2nd SDCCH/8(6) of cell B
C_SACCHC87_B_2	LOGICCH	"C_SACCHC87_B_2"	SACCH/C8(7) associated with 2nd SDCCH/8(7) of cell B
C_SACCHC80_B_3	LOGICCH	"C_SACCHC80_B_3"	SACCH/C8(0) associated with 3rd SDCCH/8(0) of cell B
C_SACCHC81_B_3	LOGICCH	"C_SACCHC81_B_3"	SACCH/C8(1) associated with 3rd SDCCH/8(1) of cell B
C_SACCHC82_B_3	LOGICCH	"C_SACCHC82_B_3"	SACCH/C8(2) associated with 3rd SDCCH/8(2) of cell B
C_SACCHC83_B_3	LOGICCH	"C_SACCHC83_B_3"	SACCH/C8(3) associated with 3rd SDCCH/8(3) of cell B
C_SACCHC84_B_3	LOGICCH	"C_SACCHC84_B_3"	SACCH/C8(4) associated with 3rd SDCCH/8(4) of cell B
C_SACCHC85_B_3	LOGICCH	"C_SACCHC85_B_3"	SACCH/C8(5) associated with 3rd SDCCH/8(5) of cell B
C_SACCHC86_B_3	LOGICCH	"C_SACCHC86_B_3"	SACCH/C8(6) associated with 3rd SDCCH/8(6) of cell B
C_SACCHC87_B_3	LOGICCH	"C_SACCHC87_B_3"	SACCH/C8(7) associated with 3rd SDCCH/8(7) of cell B
C_SACCHC80_C_1	LOGICCH	"C_SACCHC80_C_1"	SACCH/C8(0) associated with 1st SDCCH/8(0) of cell C
C_SACCHC81_C_1	LOGICCH	"C_SACCHC81_C_1"	SACCH/C8(1) associated with 1st SDCCH/8(1) of cell C
C_SACCHC82_C_1	LOGICCH	"C_SACCHC82_C_1"	SACCH/C8(2) associated with 1st SDCCH/8(2) of cell C
C_SACCHC83_C_1	LOGICCH	"C_SACCHC83_C_1"	SACCH/C8(3) associated with 1st SDCCH/8(3) of cell C
C_SACCHC84_C_1	LOGICCH	"C_SACCHC84_C_1"	SACCH/C8(4) associated with 1st SDCCH/8(4) of cell C
C_SACCHC85_C_1	LOGICCH	"C_SACCHC85_C_1"	SACCH/C8(5) associated with 1st SDCCH/8(5) of cell C
C_SACCHC86_C_1	LOGICCH	"C_SACCHC86_C_1"	SACCH/C8(6) associated with 1st SDCCH/8(6) of cell C
C_SACCHC87_C_1	LOGICCH	"C_SACCHC87_C_1"	SACCH/C8(7) associated with 1st SDCCH/8(7) of cell C
C_SACCHC80_C_2	LOGICCH	"C_SACCHC80_C_2"	SACCH/C8(0) associated with 2nd SDCCH/8(0) of cell C
C_SACCHC81_C_2	LOGICCH	"C_SACCHC81_C_2"	SACCH/C8(1) associated with 2nd SDCCH/8(1) of cell C
C_SACCHC82_C_2	LOGICCH	"C_SACCHC82_C_2"	SACCH/C8(2) associated with 2nd SDCCH/8(2) of cell C

C_SACCHC83_C_2	LOGICCH	"C_SACCHC83_C_2"	SACCH/C8(3) associated with 2nd SDCCH/8(3) of cell C
C_SACCHC84_C_2	LOGICCH	"C_SACCHC84_C_2"	SACCH/C8(4) associated with 2nd SDCCH/8(4) of cell C
C_SACCHC85_C_2	LOGICCH	"C_SACCHC85_C_2"	SACCH/C8(5) associated with 2nd SDCCH/8(5) of cell C
C_SACCHC86_C_2	LOGICCH	"C_SACCHC86_C_2"	SACCH/C8(6) associated with 2nd SDCCH/8(6) of cell C
C_SACCHC87_C_2	LOGICCH	"C_SACCHC87_C_2"	SACCH/C8(7) associated with 2nd SDCCH/8(7) of cell C
C_SACCHC80_C_3	LOGICCH	"C_SACCHC80_C_3"	SACCH/C8(0) associated with 3rd SDCCH/8(0) of cell C
C_SACCHC81_C_3	LOGICCH	"C_SACCHC81_C_3"	SACCH/C8(1) associated with 3rd SDCCH/8(1) of cell C
C_SACCHC82_C_3	LOGICCH	"C_SACCHC82_C_3"	SACCH/C8(2) associated with 3rd SDCCH/8(2) of cell C
C_SACCHC83_C_3	LOGICCH	"C_SACCHC83_C_3"	SACCH/C8(3) associated with 3rd SDCCH/8(3) of cell C
C_SACCHC84_C_3	LOGICCH	"C_SACCHC84_C_3"	SACCH/C8(4) associated with 3rd SDCCH/8(4) of cell C
C_SACCHC85_C_3	LOGICCH	"C_SACCHC85_C_3"	SACCH/C8(5) associated with 3rd SDCCH/8(5) of cell C
C_SACCHC86_C_3	LOGICCH	"C_SACCHC86_C_3"	SACCH/C8(6) associated with 3rd SDCCH/8(6) of cell C
C_SACCHC87_C_3	LOGICCH	"C_SACCHC87_C_3"	SACCH/C8(7) associated with 3rd SDCCH/8(7) of cell C
C_SDCCH4_A	LOGICCH	"C_SDCCH4_A"	SDCCH/4 channel of cell A
C_SDCCH4_B	LOGICCH	"C_SDCCH4_B"	SDCCH/4 channel of cell B
C_SDCCH4_C	LOGICCH	"C_SDCCH4_C"	SDCCH/4 channel of cell C
C_SDCCH4_D	LOGICCH	"C_SDCCH4_D"	SDCCH/4 channel of cell D
C_SDCCH4_E	LOGICCH	"C_SDCCH4_E"	SDCCH/4 channel of cell E
C_SDCCH4_F	LOGICCH	"C_SDCCH4_F"	SDCCH/4 channel of cell F
C_SDCCH4_G	LOGICCH	"C_SDCCH4_G"	SDCCH/4 channel of cell G
C_SDCCH4_H	LOGICCH	"C_SDCCH4_H"	SDCCH/4 channel of cell H
C_SDCCH40_A	LOGICCH	"C_SDCCH40_A"	SDCCH/4(0) of cell A
C_SDCCH41_A	LOGICCH	"C_SDCCH41_A"	SDCCH/4(1) of cell A
C_SDCCH42_A	LOGICCH	"C_SDCCH42_A"	SDCCH/4(2) of cell A
C_SDCCH43_A	LOGICCH	"C_SDCCH43_A"	SDCCH/4(3) of cell A
C_SDCCH40_B	LOGICCH	"C_SDCCH40_B"	SDCCH/4(0) of cell B
C_SDCCH41_B	LOGICCH	"C_SDCCH41_B"	SDCCH/4(1) of cell B
C_SDCCH42_B	LOGICCH	"C_SDCCH42_B"	SDCCH/4(2) of cell B
C_SDCCH43_B	LOGICCH	"C_SDCCH43_B"	SDCCH/4(3) of cell B
C_SDCCH40_C	LOGICCH	"C_SDCCH40_C"	SDCCH/4(0) of cell C
C_SDCCH41_C	LOGICCH	"C_SDCCH41_C"	SDCCH/4(1) of cell C
C_SDCCH42_C	LOGICCH	"C_SDCCH42_C"	SDCCH/4(2) of cell C
C_SDCCH43_C	LOGICCH	"C_SDCCH43_C"	SDCCH/4(3) of cell C
C_SDCCH40_D	LOGICCH	"C_SDCCH40_D"	SDCCH/4(0) of cell D
C_SDCCH41_D	LOGICCH	"C_SDCCH41_D"	SDCCH/4(1) of cell D
C_SDCCH42_D	LOGICCH	"C_SDCCH42_D"	SDCCH/4(2) of cell D
C_SDCCH43_D	LOGICCH	"C_SDCCH43_D"	SDCCH/4(3) of cell D
C_SDCCH40_E	LOGICCH	"C_SDCCH40_E"	SDCCH/4(0) of cell E
C_SDCCH41_E	LOGICCH	"C_SDCCH41_E"	SDCCH/4(1) of cell E
C_SDCCH42_E	LOGICCH	"C_SDCCH42_E"	SDCCH/4(2) of cell E
C_SDCCH43_E	LOGICCH	"C_SDCCH43_E"	SDCCH/4(3) of cell E
C_SDCCH40_F	LOGICCH	"C_SDCCH40_F"	SDCCH/4(0) of cell F
C_SDCCH41_F	LOGICCH	"C_SDCCH41_F"	SDCCH/4(1) of cell F
C_SDCCH42_F	LOGICCH	"C_SDCCH42_F"	SDCCH/4(2) of cell F
C_SDCCH43_F	LOGICCH	"C_SDCCH43_F"	SDCCH/4(3) of cell F
C_SDCCH40_G	LOGICCH	"C_SDCCH40_G"	SDCCH/4(0) of cell G
C_SDCCH41_G	LOGICCH	"C_SDCCH41_G"	SDCCH/4(1) of cell G
C_SDCCH42_G	LOGICCH	"C_SDCCH42_G"	SDCCH/4(2) of cell G
C_SDCCH43_G	LOGICCH	"C_SDCCH43_G"	SDCCH/4(3) of cell G
C_SDCCH40_H	LOGICCH	"C_SDCCH40_H"	SDCCH/4(0) of cell H
C_SDCCH41_H	LOGICCH	"C_SDCCH41_H"	SDCCH/4(1) of cell H
C_SDCCH42_H	LOGICCH	"C_SDCCH42_H"	SDCCH/4(2) of cell H
C_SDCCH43_H	LOGICCH	"C_SDCCH43_H"	SDCCH/4(3) of cell H
C_SDCCH8_A_1	LOGICCH	"C_SDCCH8_A_1"	1st SDCCH/8 channel of cell A

C_SDCCH8_A_2	LOGICCH	"C_SDCCH8_A_2"	2nd SDCCH/8 channel of cell A
C_SDCCH8_A_3	LOGICCH	"C_SDCCH8_A_3"	3rd SDCCH/8 channel of cell A
C_SDCCH8_B_1	LOGICCH	"C_SDCCH8_B_1"	1st SDCCH/8 channel of cell B
C_SDCCH8_B_2	LOGICCH	"C_SDCCH8_B_2"	2nd SDCCH/8 channel of cell B
C_SDCCH80_A_1	LOGICCH	"C_SDCCH80_A_1"	1st SDCCH/8(0) of cell A
C_SDCCH81_A_1	LOGICCH	"C_SDCCH81_A_1"	1st SDCCH/8(1) of cell A
C_SDCCH82_A_1	LOGICCH	"C_SDCCH82_A_1"	1st SDCCH/8(2) of cell A
C_SDCCH83_A_1	LOGICCH	"C_SDCCH83_A_1"	1st SDCCH/8(3) of cell A
C_SDCCH84_A_1	LOGICCH	"C_SDCCH84_A_1"	1st SDCCH/8(4) of cell A
C_SDCCH85_A_1	LOGICCH	"C_SDCCH85_A_1"	1st SDCCH/8(5) of cell A
C_SDCCH86_A_1	LOGICCH	"C_SDCCH86_A_1"	1st SDCCH/8(6) of cell A
C_SDCCH87_A_1	LOGICCH	"C_SDCCH87_A_1"	1st SDCCH/8(7) of cell A
C_SDCCH80_A_2	LOGICCH	"C_SDCCH80_A_2"	2nd SDCCH/8(0) of cell A
C_SDCCH81_A_2	LOGICCH	"C_SDCCH81_A_2"	2nd SDCCH/8(1) of cell A
C_SDCCH82_A_2	LOGICCH	"C_SDCCH82_A_2"	2nd SDCCH/8(2) of cell A
C_SDCCH83_A_2	LOGICCH	"C_SDCCH83_A_2"	2nd SDCCH/8(3) of cell A
C_SDCCH84_A_2	LOGICCH	"C_SDCCH84_A_2"	2nd SDCCH/8(4) of cell A
C_SDCCH85_A_2	LOGICCH	"C_SDCCH85_A_2"	2nd SDCCH/8(5) of cell A
C_SDCCH86_A_2	LOGICCH	"C_SDCCH86_A_2"	2nd SDCCH/8(6) of cell A
C_SDCCH87_A_2	LOGICCH	"C_SDCCH87_A_2"	2nd SDCCH/8(7) of cell A
C_SDCCH80_A_3	LOGICCH	"C_SDCCH80_A_3"	3rd SDCCH/8(0) of cell A
C_SDCCH81_A_3	LOGICCH	"C_SDCCH81_A_3"	3rd SDCCH/8(1) of cell A
C_SDCCH82_A_3	LOGICCH	"C_SDCCH82_A_3"	3rd SDCCH/8(2) of cell A
C_SDCCH83_A_3	LOGICCH	"C_SDCCH83_A_3"	3rd SDCCH/8(3) of cell A
C_SDCCH84_A_3	LOGICCH	"C_SDCCH84_A_3"	3rd SDCCH/8(4) of cell A
C_SDCCH85_A_3	LOGICCH	"C_SDCCH85_A_3"	3rd SDCCH/8(5) of cell A
C_SDCCH86_A_3	LOGICCH	"C_SDCCH86_A_3"	3rd SDCCH/8(6) of cell A
C_SDCCH87_A_3	LOGICCH	"C_SDCCH87_A_3"	3rd SDCCH/8(7) of cell A
C_SDCCH80_B_1	LOGICCH	"C_SDCCH80_B_1"	1st SDCCH/8(0) of cell B
C_SDCCH81_B_1	LOGICCH	"C_SDCCH81_B_1"	1st SDCCH/8(1) of cell B
C_SDCCH82_B_1	LOGICCH	"C_SDCCH82_B_1"	1st SDCCH/8(2) of cell B
C_SDCCH83_B_1	LOGICCH	"C_SDCCH83_B_1"	1st SDCCH/8(3) of cell B
C_SDCCH84_B_1	LOGICCH	"C_SDCCH84_B_1"	1st SDCCH/8(4) of cell B
C_SDCCH85_B_1	LOGICCH	"C_SDCCH85_B_1"	1st SDCCH/8(5) of cell B
C_SDCCH86_B_1	LOGICCH	"C_SDCCH86_B_1"	1st SDCCH/8(6) of cell B
C_SDCCH87_B_1	LOGICCH	"C_SDCCH87_B_1"	1st SDCCH/8(7) of cell B
C_SDCCH80_B_2	LOGICCH	"C_SDCCH80_B_2"	2nd SDCCH/8(0) of cell B
C_SDCCH81_B_2	LOGICCH	"C_SDCCH81_B_2"	2nd SDCCH/8(1) of cell B
C_SDCCH82_B_2	LOGICCH	"C_SDCCH82_B_2"	2nd SDCCH/8(2) of cell B
C_SDCCH83_B_2	LOGICCH	"C_SDCCH83_B_2"	2nd SDCCH/8(3) of cell B
C_SDCCH84_B_2	LOGICCH	"C_SDCCH84_B_2"	2nd SDCCH/8(4) of cell B
C_SDCCH85_B_2	LOGICCH	"C_SDCCH85_B_2"	2nd SDCCH/8(5) of cell B
C_SDCCH86_B_2	LOGICCH	"C_SDCCH86_B_2"	2nd SDCCH/8(6) of cell B
C_SDCCH87_B_2	LOGICCH	"C_SDCCH87_B_2"	2nd SDCCH/8(7) of cell B
C_SDCCH80_B_3	LOGICCH	"C_SDCCH80_B_3"	3rd SDCCH/8(0) of cell B
C_SDCCH81_B_3	LOGICCH	"C_SDCCH81_B_3"	3rd SDCCH/8(1) of cell B
C_SDCCH82_B_3	LOGICCH	"C_SDCCH82_B_3"	3rd SDCCH/8(2) of cell B
C_SDCCH83_B_3	LOGICCH	"C_SDCCH83_B_3"	3rd SDCCH/8(3) of cell B
C_SDCCH84_B_3	LOGICCH	"C_SDCCH84_B_3"	3rd SDCCH/8(4) of cell B
C_SDCCH85_B_3	LOGICCH	"C_SDCCH85_B_3"	3rd SDCCH/8(5) of cell B
C_SDCCH86_B_3	LOGICCH	"C_SDCCH86_B_3"	3rd SDCCH/8(6) of cell B
C_SDCCH87_B_3	LOGICCH	"C_SDCCH87_B_3"	3rd SDCCH/8(7) of cell B
C_SDCCH80_C_1	LOGICCH	"C_SDCCH80_C_1"	1st SDCCH/8(0) of cell C
C_SDCCH81_C_1	LOGICCH	"C_SDCCH81_C_1"	1st SDCCH/8(1) of cell C
C_SDCCH82_C_1	LOGICCH	"C_SDCCH82_C_1"	1st SDCCH/8(2) of cell C
C_SDCCH83_C_1	LOGICCH	"C_SDCCH83_C_1"	1st SDCCH/8(3) of cell C
C_SDCCH84_C_1	LOGICCH	"C_SDCCH84_C_1"	1st SDCCH/8(4) of cell C
C_SDCCH85_C_1	LOGICCH	"C_SDCCH85_C_1"	1st SDCCH/8(5) of cell C
C_SDCCH86_C_1	LOGICCH	"C_SDCCH86_C_1"	1st SDCCH/8(6) of cell C
C_SDCCH87_C_1	LOGICCH	"C_SDCCH87_C_1"	1st SDCCH/8(7) of cell C
C_SDCCH80_C_2	LOGICCH	"C_SDCCH80_C_2"	2nd SDCCH/8(0) of cell C
C_SDCCH81_C_2	LOGICCH	"C_SDCCH81_C_2"	2nd SDCCH/8(1) of cell C
C_SDCCH82_C_2	LOGICCH	"C_SDCCH82_C_2"	2nd SDCCH/8(2) of cell C
C_SDCCH83_C_2	LOGICCH	"C_SDCCH83_C_2"	2nd SDCCH/8(3) of cell C
C_SDCCH84_C_2	LOGICCH	"C_SDCCH84_C_2"	2nd SDCCH/8(4) of cell C

C_SDCCH85_C_2	LOGICCH	"C_SDCCH85_C_2"	2nd SDCCH/8(5) of cell C
C_SDCCH86_C_2	LOGICCH	"C_SDCCH86_C_2"	2nd SDCCH/8(6) of cell C
C_SDCCH87_C_2	LOGICCH	"C_SDCCH87_C_2"	2nd SDCCH/8(7) of cell C
C_SDCCH80_C_3	LOGICCH	"C_SDCCH80_C_3"	3rd SDCCH/8(0) of cell C
C_SDCCH81_C_3	LOGICCH	"C_SDCCH81_C_3"	3rd SDCCH/8(1) of cell C
C_SDCCH82_C_3	LOGICCH	"C_SDCCH82_C_3"	3rd SDCCH/8(2) of cell C
C_SDCCH83_C_3	LOGICCH	"C_SDCCH83_C_3"	3rd SDCCH/8(3) of cell C
C_SDCCH84_C_3	LOGICCH	"C_SDCCH84_C_3"	3rd SDCCH/8(4) of cell C
C_SDCCH85_C_3	LOGICCH	"C_SDCCH85_C_3"	3rd SDCCH/8(5) of cell C
C_SDCCH86_C_3	LOGICCH	"C_SDCCH86_C_3"	3rd SDCCH/8(6) of cell C
C_SDCCH87_C_3	LOGICCH	"C_SDCCH87_C_3"	3rd SDCCH/8(7) of cell C
C_CHSDCCH8_FH	OCTETSTRING	'00'O	ch.type SDCCH8 with FH
C_CHSDCCH8_NonFH	OCTETSTRING	'01'O	ch.type SDCCH8 No FH
C_CHSDCCH4_NonFH	OCTETSTRING	'03'O	ch.type SDCCH4 NoFH
C_CHTCHF_FH	OCTETSTRING	'04'O	ch.type TCHF FH
C_CHTCHF_NonFH	OCTETSTRING	'05'O	ch.type TCHF NonFH
C_CHTCHH_FH	OCTETSTRING	'06'O	ch.type TCHH FH
C_CHTCHH_NonFH	OCTETSTRING	'07'O	ch.type TCHH NonFH
C_TCHF_ACCHF_1	LOGCH	"C_TCHF_ACCHF_1"	first TCH/F + ACCHs type channel
C_TCHF_ACCHF_2	LOGCH	"C_TCHF_ACCHF_2"	second TCH/F + ACCHs type channel
C_TCHH_ACCHH_1	LOGCH	"C_TCHH_ACCHH_1"	first TCH/H + ACCHs type channel
C_TCHH_ACCHH_2	LOGCH	"C_TCHH_ACCHH_2"	second TCH/H + ACCHs type channel
C_FCCH_SCH_BCCH_CCCH	LOGCH	"C_FCCH_SCH_BCCH_CCCH"	FCCH_SCH_BCCH_CCCH type channel
C_BCCH_CCCH_2	LOGCH	"C_BCCH_CCCH_2"	second BCCH_CCCH type channel
C_BCCH_CCCH_3	LOGCH	"C_BCCH_CCCH_3"	third BCCH_CCCH type channel
C_BCCH_CCCH_4	LOGCH	"C_BCCH_CCCH_4"	fourth BCCH_CCCH type channel
C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4	LOGCH	"C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4"	combined CCCH type channel
C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4	LOGCH	"C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4"	combined CCCH type channel for cell broadcast SM
C_SDCCH8_SACCHC8_1	LOGCH	"C_SDCCH8_SACCHC8_1"	first SDCCH/8 type channel
C_SDCCH8_SACCHC8_2	LOGCH	"C_SDCCH8_SACCHC8_2"	second SDCCH/8 type channel
C_SDCCH8_SACCHC8_3	LOGCH	"C_SDCCH8_SACCHC8_3"	3rd SDCCH/8 type channel
C_S0	SN	'000'B	time slot 0
C_S2	SN	'010'B	time slot 2
C_S3	SN	'011'B	time slot 3
C_S4	SN	'100'B	time slot 4
C_S6	SN	'110'B	time slot 6
C_SAVE	INTEGER	0	the OC_SaveAndProc saves input values
C_PROC	INTEGER	1	the OC_SvaAndProc processes saved values
C_RETRV	INTEGER	1	
C_U1	CCSTATE	1	CC state U1
C_U3	CCSTATE	3	CC state U3
C_U4	CCSTATE	4	CC state U4
C_U6	CCSTATE	6	CC state U6
C_U7	CCSTATE	7	CC state U7
C_U8	CCSTATE	8	CC state U8
C_U9	CCSTATE	9	CC state U9
C_U10	CCSTATE	10	CC state U10
C_U11	CCSTATE	11	CC state U11
C_U12	CCSTATE	12	CC state U12
C_U19	CCSTATE	19	CC state U19
C_U26	CCSTATE	26	CC state U26
C_NxtButOne	SENDINGMODE	1	send the second message on the next but one paging subblock
C_FmrAGB	SENDINGMODE	2	send the second message on the former access grant block
C_BfReOcc	SENDINGMODE	3	send the second message before

			the MS's original paging subchannel re-occurs, but later than the next paging block of that CCCH
C_NxtButOneNxt	SENDINGMODE	4	nothing is sent in the next but one paging sub block, then send the second message in the next paging subblock of the MS's paging subchannel
C_CMServiceTypeE	CMSVTYPE	'0010'B	CM Service Type for emergency call.
C_Telephony	IA5String	"C_Telephony"	telephony service (TS11)
C_EmgCall	IA5String	"C_EmgCall"	emergency call service (TS12)
C_AltSpchFax	IA5String	"C_AltSpchFax"	alternate speech and G3 fax service (TS61)
C_AutoFax	IA5String	"C_AutoFax"	automatic G3 fax service (TS62)
C_Async300	IA5String	"C_Async300"	data circuit duplex async. 300 bit/s service (BS21)
C_Async1200	IA5String	"C_Async1200"	data circuit duplex async. 1200 bit/s service (BS22)
C_Async120075	IA5String	"C_Async120075"	data circuit duplex async. 1200/75 bit/s service (BS23)
C_Async2400	IA5String	"C_Async2400"	data circuit duplex async. 2400 bit/s service (BS24)
C_Async4800	IA5String	"C_Async4800"	data circuit duplex async. 4800 bit/s service (BS25)
C_Async9600	IA5String	"C_Async9600"	data circuit duplex async 9600 bit/s service (BS26)
C_Sync1200	IA5String	"C_Sync1200"	data circuit duplex sync. 1200 bit/s service (BS31)
C_Sync2400	IA5String	"C_Sync2400"	data circuit duplex sync. 2400 bit/s service (BS32)
C_Sync4800	IA5String	"C_Sync4800"	data circuit duplex sync. 4800 bit/s service (BS33)
C_Sync9600	IA5String	"C_Sync9600"	data circuit duplex sync 9600 bit/s service (BS34)
C_PAD300	IA5String	"C_PAD300"	PAD access 300 bit/s service (BS41)
C_PAD1200	IA5String	"C_PAD1200"	PAD access 1200 bit/s service (BS42)
C_PAD120075	IA5String	"C_PAD120075"	PAD access 1200/75 bit/s service (BS43)
C_PAD2400	IA5String	"C_PAD2400"	PAD access 2400 bit/s service (BS44)
C_PAD4800	IA5String	"C_PAD4800"	PAD access 4800 bit/s service (BS45)
C_PAD9600	IA5String	"C_PAD9600"	PAD access 9600 bit/s service (BS46)
C_Packet2400	IA5String	"C_Packet2400"	packt access 2400 bit/s service (BS51)
C_Packet4800	IA5String	"C_Packet4800"	packt access 4800 bit/s service (BS52)
C_Packet9600	IA5String	"C_Packet9600"	packt access 9600 bit/s service (BS53)
C_AltSpchData	IA5String	"C_AltSpchData"	alternate speech/data service (BS61)
C_SpchData	IA5String	"C_SpchData"	speech followed data service (BS81)
C_I	INTEGER	1	I command of L 2
C_one	INTEGER	1	one frequency, no hopping
C_Immass	BITSTRING	'000'B	activation for immediate assignment
C_Ass	BITSTRING	'001'B	activation for assignment
C_Asynho	BITSTRING	'010'B	activation for non synchronous handover
C_Synho	BITSTRING	'011'B	activation for synchronous handover
C_Rcv	BITSTRING	'101'B	activation for receiving only
C_LocEndRel	RelMode	'01'O	local end release
C_MaxPwrLvlD	INTEGER	15	Max power level for MS DCS

C_MaxPwrLvlG	INTEGER	19	Max power level for MS GSM
C_rc_conditIEerror	REJCAU	'64'O	reject cause: Conditional IE error
C_rc_congestion	REJCAU	'16'O	reject cause: congestion
C_rc_illegal_ms	REJCAU	'03'O	reject cause: illegal MS
C_rc_illegal_me	REJCAU	'06'O	reject cause: illegal ME
C_rc_imsiunknownhlr	REJCAU	'02'O	reject cause: IMSI unknown in HLR
C_rc_imsiunknownvlr	REJCAU	'04'O	reject cause: IMSI unknown in VLR
C_rc_invalidmaninfo	REJCAU	'60'O	reject cause: invalid mandatory information
C_rc_LAnotallowed	REJCAU	'0C'O	reject cause: LocationArea is not allowed
C_rc_networkfailure	REJCAU	'11'O	reject cause: network failure
C_rc_notidentified	REJCAU	'26'O	reject cause: can not be identified
C_rc_plmn_not	REJCAU	'0B'O	reject cause: PLMN not allowed
C_rc_protocolerror	REJCAU	'6F'O	reject cause: Protocol error unspecified
C_rc_reqservoptnotsub	REJCAU	'21'O	reject cause: requested service option not subscribed
C_rc_roamingnot	REJCAU	'0D'O	reject cause: Roaming not allowed
C_Restablishment	B_1	'0'B	Restablishment allowed
C_noRestablishment	B_1	'1'B	No restablishment allowed
C_SlotNU	SN	'111'B	Slot Number not used
C_CellReselectHys12	INTEGER	6	Cell Reselection Hysteresis value 12db
C_CellReselectHys4	INTEGER	2	Cell Reselection Hysteresis value 4db
C_RI_alternate	B_8	'11010001'B	Bcap repeat indicator for alternate (circular selection)
C_RI_follow	B_8	'11010011'B	Bcap repeat indicator for successive (sequential selection)
C_SIMIn	BOOLEAN	FALSE	SIM is always inserted in a test case
C_SIMneedRmv	BOOLEAN	TRUE	Removal of SIM is needed in a test case (SIM is not always inserted.)
C_Yes	BOOLEAN	TRUE	
C_No	BOOLEAN	FALSE	
C_Start_Imsi	INTEGER	1	Test case body starts without TMSI
C_Start_Tmsi	INTEGER	2	Test case body starts with Tmsi_01
C_Start_TmsiOff	INTEGER	5	Test case body starts with Tmsi_01. MS has been deactivated.
C_Start_Rej	INTEGER	3	Test case body starts with LAI deleted. MS has been deactivated.
C_Start_PLMNnot	INTEGER	4	Test case body starts with PLMN not allowed
C_Test_fh	INTEGER	1	Test case with possibly frequency hopping
C_Test_nfh	INTEGER	2	Test case with no frequency hopping
C_Test_egsm	INTEGER	3	Test case for egsm
C_Test_cbms	INTEGER	4	Test case for CBMS
C_normal_updating	B_2	'00'B	normal location updating
C_periodic_updating	B_2	'01'B	periodic location updating

C_normalOrperiodic	B_2	'0?'B	periodic or normal location updating
C_imsi_attach	B_2	'10'B	location updating IMSI attach
C_norm_period_attach	B_2	'??'B	imsi attach or periodic location updating or normal location updating
C_normal_paging	B_2	'00'B	normal page mode
C_extended_paging	B_2	'01'B	extended paging
C_reorg_paging	B_2	'10'B	paging reorganisation
C_sab_paging	B_2	'11'B	same as before paging
C_non_synchronised	B_2	'00'B	non-synchronised
C_synchronised	B_2	'01'B	synchronised
C_pre_synchronised	B_2	'10'B	pre-synchronised
C_pseudo_synchronised	B_2	'11'B	pseudo-synchronised
C_not_report_otd	B_1	'0'B	not report observed time difference
C_report_otd	B_1	'1'B	report observed time difference
C_Tzone0	TZONES	0	Time zone 0 (used in TC_34_2_1, TC_34_2_5_3)
C_Tzone1	TZONES	4	Time zone 4 (used in TC_34_2_2, TC_34_2_7)
C_Tzone2	TZONES	10	Time zone 10 (used in TC_34_2_3)
C_Tzone3	TZONES	15	Time zone 15 (used in TC_34_2_4)
C_Tzone4	TZONES	25	Time zone 25 (used in TC_34_2_5_1)
C_Tzone5	TZONES	35	Time zone 35 (used in TC_34_2_5_2)
C_Tzone6	TZONES	45	Time zone 45 (used in TC_34_2_8)
C_T_AssCmd	INTEGER	900	600ms + 300ms(tolerance)
C_T_Wait1stChReq	INTEGER	35000	max time for full BCCH data decoding
C_T_mrsrp	INTEGER	1500	3 times time btw 2 msr rpt.
C_T_T3124	INTEGER	675	T3124(675ms)
C_T_T3211plus	INTEGER	20000	20s
C_T_T3212	INTEGER	360000	T3212(6min)
C_T_T3212min	INTEGER	345000	T3212(6min) - 15sec
C_T_T3212dif	INTEGER	15000	Difference between T3212 and T3212min
C_T_T3210	INTEGER	20000	timer T3210 in ms
C_T_T3211min	INTEGER	13500	90% T3211 in ms
C_T_T3230min	INTEGER	13500	90% T3230 in ms
C_T_T3240min	INTEGER	9000	90% T3240 in ms
C_T_T3240tol	INTEGER	2000	20% T3240 in ms
C_T_Wait	INTEGER	5000	5s. T3126max
C_T_Wait1	INTEGER	1000	1s
C_T_Wait_FAC	INTEGER	1500	1.5s Facility response time (1s) + tolerance (0.5s)
C_release_time	INTEGER	11	release timer value
C_RegCFNRy	INTEGER	1	registration of call forwarding service for CFNRy (speech)
C_RegCFU	INTEGER	2	registration of call forwarding service for CFU (all facsimile)
C_RegCF	INTEGER	3	registration of call forwarding service for CF (all facsimile)
C_ErsCFC	INTEGER	4	erasure of call forwarding service for CFC (all facsimile)
C_ErsCFNRc	INTEGER	5	erasure of call forwarding service for CFNRc (all basic services)
C_ErsCFNRy	INTEGER	6	erasure of call forwarding service for CFNRy (all facsimile)
C_ActCF	INTEGER	7	activation of call forwarding service for CF (all synchronous services)
C_ActCFU	INTEGER	8	activation of call forwarding service for CFU (all basic services)

C_DeactCFC	INTEGER	9	deactivation of call forwarding service for CFC (speech)
C_DeactCFNRc	INTEGER	10	deactivation of call forwarding service for CFNRc (all facsimile)
C_InterrogCFC	INTEGER	11	interrogation of call forwarding service for CFC (Speech)
C_InterrogCFB	INTEGER	12	interrogation of call forwarding service for CFB (all basic services)
C_NotifyCFB	INTEGER	13	notification of call forwarding service for CFB (incoming call is forwarded)
C_NotifyCFU	INTEGER	14	notification of call forwarding service for CFU (provisioned, registered, active)
C_NotifyCFC	INTEGER	15	notification of call forwarding service for CFC (provisioned, registered, active)
C_NotifyCFNRc	INTEGER	16	notification of call forwarding service for CFNRc (Ms not reachable)
C_RegPswd	INTEGER	17	Registration of password for all call barring services
C_ActBOAC	INTEGER	18	Activation of BOAC
C_ActBICRoam	INTEGER	19	Activation of BICRoam
C_ActBOIC	INTEGER	20	Activation of BOIC
C_ActBAIC	INTEGER	21	Activation of BAIC
C_DeactBO	INTEGER	22	Deactivation of BO
C_DeactBI	INTEGER	23	Deactivation of BI
C_DeactBOICExHC	INTEGER	24	Deactivation of BOICExHC
C_NotifyBI	INTEGER	25	Notify of BI
C_InterrogBOIC	INTEGER	26	Interrogation of BOIC
C_InterrogBOICExHC	INTEGER	27	Interrogation of BOICExHC
C_Full	IA5String	"F"	full rate channel (any speech version or data)
C_Full_v2	IA5String	"Fv2"	full rate speech channel, version 2
C_Half	IA5String	"H"	half rate channel
C_nirr_dontcare	B_1	'?'B	NIRR no meaning or requested
C_Rchr_dontcare	B_2	'??'B	radio channel requirement not checked
C_Rchr_Full	B_2	'01'B	radio channel requirement : full rate
C_Rchr_DualForDualH	B_2	'1?'B	radio channel requirement : dual rate support/half rate preferred or dual rate support/full rate preferred
C_Rchr_DualForDualH0	B_2	'10'B	radio channel requirement : dual rate support/half rate preferred or dual rate support/full rate preferred
C_StartingTimeHO	INTEGER	238	Starting time 238 (1.1s), in TC_26_6_5_1, TC_26_6_5_4_2
C_StrT_1000	INTEGER	1000	Starting time 1000, in TC_26_6_13_1
C_Norm	OCTETSTRING	'00'O	Establish mode: normal
C_Speech	B_3	'000'B	Info. Transfer Cap. speech
C_3100Hz	B_3	'010'B	Info. Transfer Cap. 3.1 kHz
C_UDI	B_3	'001'B	Info. Transfer Cap. UDI
C_FAX3	B_3	'011'B	Info. Transfer Cap. FAX3
C_SDUintegrity	B_2	'00'B	Structure SDU integrity
C_Unstructured	B_2	'11'B	Structure Unstructured
C_struc_dontchk	B_2	'??'B	Structure not checked
C_nirr_nomeaning	B_1	'0'B	NIRR no meaning
C_no_rate_adaption	B_2	'00'B	Rate adaption no
C_rate_adaption_V110	B_2	'01'B	Rate adaption V110
C_rate_adaption_X31	B_2	'10'B	Rate adaption X31 flag
C_sacp_NA	B_3	'001'B	Signaling access NA (sec.11.8.1.5.1)
C_I440_450	B_3	'001'B	Signaling access I440/450

C_X21	B_3	'010'B	Signaling access X21
C_X28_unui	B_3	'100'B	Signaling access X28 U NUI
C_X28_nond	B_3	'101'B	Signaling access X28 non dedicated
C_X32	B_3	'110'B	Signaling access X32
C_Synchronous	B_1	'0'B	Synchronous
C_Asynchronous	B_1	'1'B	Asynchronous
C_stopbit_NA	B_1	'0'B	Number of Stop bits NA
C_stopbit_dontcare	B_1	'?'B	Number of Stop bits not care
C_databit_NA	B_1	'1'B	Number of Data bits not care
C_databit_dontcare	B_1	'?'B	Number of Data bits NA
C_300bs	B_4	'0001'B	User rate 300 bs
C_1200bs	B_4	'0010'B	User rate 1200 bs
C_120075bs	B_4	'0111'B	User rate 1200/75 bs
C_2400bs	B_4	'0011'B	User rate 2400 bs
C_4800bs	B_4	'0100'B	User rate 4800 bs
C_9600bs	B_4	'0101'B	User rate 9600 bs
C_parity_NA	B_3	'011'B	Parity NA
C_parity_dontcare	B_3	'???'B	Parity not care
C_ir_8kbs	B_2	'10'B	Intermediate rate 8 Kbs
C_ir_16kbs	B_2	'11'B	Intermediate rate 16 Kbs
C_ir_8or16kbs	B_2	'1?'B	Intermediate rate 8 or 16 Kbs
C_transparent	B_2	'00'B	Connection element T
C_notransparent	B_2	'01'B	Connection element NT
C_BothT	B_2	'10'B	Connection element both T
C_BothNT	B_2	'11'B	Connection element both NT
C_modemt_none	B_5	'00000'B	no modem
C_modemt_V21	B_5	'00001'B	modem V21
C_modemt_V22	B_5	'00010'B	modem V22
C_modemt_V22bis	B_5	'00011'B	modem V22bis
C_modemt_V23	B_5	'00100'B	modem V23
C_modemt_V32	B_5	'00110'B	modem V32
C_ISO6429	B_5	'01000'B	Inband flow control
C_COPnoFLCT	B_5	'01100'B	No flow control
C_X25_flct	B_5	'00110'B	X25 flow control
C_Outband	FLWCNTL	0	outband flow control
C_Inband	FLWCNTL	1	outband flow control
C_Nocontrol	FLWCNTL	2	outband flow control
C_uil2p_dontcare	B_5	'?????'B	not checked
C_HSN_0	INTEGER	0	cyclic hopping
maxAddressLength	Asn1Integer	20	
maxISDN_AddressLength	Asn1Integer	9	
maxISDN_SubaddressLength	Asn1Integer	21	
maxNumOfBasicServiceGroups	Asn1Integer	13	
maxNumOfCUG	Asn1Integer	10	
maxNumberOfSegmentsPerDataInterval	Asn1Integer	8191	
maxSignalInfoLength	Asn1Integer	200	
maxUSSD_StringLength	Asn1Integer	160	
max10TimesIncrement	Asn1Integer	8191	
max10TimesIncrementPerDataInterval	Asn1Integer	8191	
max10TimesInitialTime	Asn1Integer	8191	
max100TimesScalingFactor	Asn1Integer	8191	
max10TimesTimeInterval	Asn1Integer	8191	
max10TimesUnitsPerTime	Asn1Integer	8191	
Detailed Comments:			

Test case variable declarations

Test Case Variable Declarations			
Variable Name	**Type**	**Value**	**Comments**
TCV_CC	BOOLEAN	TSPC_Serv_TS11 OR TSPC_Serv_TS12 OR TSPC_Serv_TS61 OR TSPC_Serv_TS62 OR TSPC_Serv_BS21 OR TSPC_Serv_BS22 OR TSPC_Serv_BS23 OR TSPC_Serv_BS24 OR TSPC_Serv_BS25 OR TSPC_Serv_BS26 OR TSPC_Serv_BS31 OR TSPC_Serv_BS32 OR TSPC_Serv_BS33 OR TSPC_Serv_BS34 OR TSPC_Serv_BS41 OR TSPC_Serv_BS42 OR TSPC_Serv_BS43 OR TSPC_Serv_BS44 OR TSPC_Serv_BS45 OR TSPC_Serv_BS46 OR TSPC_Serv_BS51 OR TSPC_Serv_BS52 OR TSPC_Serv_BS53 OR TSPC_Serv_BS61 OR TSPC_Serv_BS81	
TCV_AssCmd	ASS_CMD_PDU		to hold ASSIGNMENT COMMAND PDU
TCV_Bcap1	BCAP		to hold bearer capability
TCV_BcapMT1	BCAP		to hold BC for MT call
TCV_BcapMO1	BCAP		to hold BC for MO call
TCV_Bcap2	BCAP		to hold bearer capability
TCV_BcapMT2	BCAP		to hold BC for MT call
TCV_bs_ag_blks_res	INTEGER		
TCV_bs_cc_chans	INTEGER		
TCV_bs_pa_mfrms	INTEGER		
TCV_CalledNum	CDPN		to hold called party number
TCV_CallProc	CALL_PROC_PDU		to hold CALL PROCEEDING PDU
TCV_CallCfm	CALL_CO_PDU		to hold CALL COMFIRM PDU
TCV_Cau	CAU		to hold cause
TCV_Cau0	CAU		to hold cause
TCV_CAgsm	CCHD		To compute Cell Channel Descriptor for gsm
TCV_CAdcs	CCHD		To compute Cell Channel Descriptor for dcs
TCV_CB1	OCTETSTRING		store SMSCB block data
TCV_CB2	OCTETSTRING		store SMSCB block data
TCV_CB3	OCTETSTRING		store SMSCB block data
TCV_CB4	OCTETSTRING		store SMSCB block data
TCV_CE	CE		
TCV_Ccchg	INTEGER		CCCH_GROUP
TCV_Ccd0A	CCD		a copy of control channel description in use for cell A
TCV_Ccd0B	CCD		a copy of control channel description in use for cell B
TCV_Ccd0C	CCD		a copy of control channel description in use for cell C
TCV_Ccd0D	CCD		a copy of control channel description in use for cell D
TCV_Ccd0E	CCD		a copy of control channel description in use for cell E
TCV_Ccd0F	CCD		a copy of control channel description in use for cell F
TCV_Ccd0G	CCD		a copy of control channel description in use for cell G
TCV_Ccd0H	CCD		a copy of control channel description in use for cell H

TCV_cchdescr	OCTETSTRING		Frequency list of Cell Channel Descr, used in EGSM
TCV_CCst	CST		to hold call status
TCV_cchannel	B_3		to hold the type of the control channel in IdleUpdated test step
TCV_ch	LOGICCH	"dummy"	to hold logic channel
TCV_ch1	LOGICCH	"dummy"	to hold logic channel
TCV_sacch	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_B	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_C	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_D	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_E	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_F	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_G	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_H	LOGICCH	"dummy"	to hold logic channel
TCV_sacch8	LOGICCH	"dummy"	to hold logic channel
TCV_sacchTch	LOGICCH	"dummy"	to hold logic channel
TCV_sacchTch1	LOGICCH	"dummy"	to hold logic channel
TCV_sacchTch_H	LOGICCH	"dummy"	to hold logic channel
TCV_chTch	LOGICCH	"dummy"	to hold logic channel
TCV_chTch1	LOGICCH	"dummy"	to hold logic channel
TCV_chSms	LOGICCH	"dummy"	to hold logic channel
TCV_agch	LOGICCH	"dummy"	downlink access grant channel
TCV_WorkingCh	LOGICCH	"dummy"	to hold logic channel
TCV_cellid	CellID		to hold the cell identifier
TCV_Char	IA5String		to hold the IA5 char. corresponding to a DTMF tone
TCV_chd1	CHD		to hold channel description IE
TCV_cchd1	CCHD		to hold cell channel description IE
TCV_ChMod	CHMOD		to hold channel mode IE
TCV_ChModb	CHMOD		to hold channel mode IE for second mode in dual mode services
TCV_Tchtype	CH_TDMA	'00001'B	Traffic channel type and TDMA offset in channel description IE
TCV_chtype	CH_TDMA		to channel type and TDMA offset in channel description IE
TCV_cksn	BITSTRING		to hold the CKSN.
TCV_Cnt	INTEGER		general loop counter
TCV_Cnt1	INTEGER		general loop counter
TCV_Cnt2	INTEGER		general loop counter
TCV_Cntstart	BOOLEAN	FALSE	general loop counter start decision
TCV_Cntref	INTEGER		Reference for general loop counter
TCV_CphAlg	BITSTRING		ciphering algorithm
TCV_CphKey	BITSTRING		ciphering key
TCV_CphMd	CPHMS		ciphering mode setting
TCV_CPDataRetx	INTEGER		Number of CP Data retransmissions for SMS
TCV_ChRate	IA5String	C_Full	to hold channel rate
TCV_ChRate1	IA5String	C_Full	to hold channel rate
TCV_counter_c	INTEGER		loop counter c
TCV_counter_k	INTEGER		loop counter k
TCV_Fk	INTEGER		to hold an interval between events
TCV_flist	OCTETSTRING		Frequency List, used in EGSM for IE "frequency list" or "frequency short list"
TCV_flistl	LENGTH		Length of Frequency List, used in EGSM for IE "frequency list" or "frequency short list"
TCV_Fn	FN		to hold the frame number (T1' , T2, T3)
TCV_Fn1	FN		to hold the frame number (T1' , T2, T3)
TCV_FnAss	FN		to hold the frame number in AssCmd statement
TCV_FnBurst	FN		to hold the frame number of first

			burst of handover access
TCV_FollowingOctets	OCTETSTRING		Contains the octets following the invoke id
TCV_HoaccPara	HOACC_PARA		Handover access timing and power
TCV_Horf	HORF		Variable for HO reference in HO-Messages of HO cases.
TCV_HoCMD	HO_CMD_PDU		to hold handover PDU
TCV_Hrf	HORF		to hold handover reference
TCV_iel	LENGTH		Length of IE
TCV_kcnt	INTEGER		loop counter
TCV_K	INTEGER		
TCV_L1Head0	L1HD		to hold L 1 header
TCV_L1Head	L1HD		to hold L 1 header
TCV_lac	OCTETSTRING		to hold the lac. 2 octets
TCV_lacunknown	OCTETSTRING	'??'O	to hold a wild card lac when the lac is unknown
TCV_n	INTEGER		
TCV_Neci	BITSTRING		
TCV_M	INTEGER		counter
TCV_maio	MAIO		to hold MAIO
TCV_mae1	BITSTRING		Mobile Allocation in EGSM. It includes the 1.octet of MA
TCV_mae2	BITSTRING		Mobile Allocation in EGSM. It includes the 2.octet of MA
TCV_Max	BITSTRING		to hold Max-retrans bitstring
TCV_MemCapExcd	BOOLEAN		RP error Memory Capacity Exceeded was sent
TCV_MsrRes	MSRR		to hold measurement results
TCV_Mt	BITSTRING		to hold message type
TCV_Mt1	BITSTRING		to hold message type
TCV_Modify	MODIFY_PDU		To hold Modify message
TCV_Null	BOOLEAN		to collect useless result from some test suite operations
TCV_PgCh	LOGICCH	"dummy"	logical channel for paging
TCV_Pgg	PGG		paging group
TCV_PreviousOctets	OCTETSTRING		Contains the octets previous to the invoke id
TCV_ProtErrorUnspec	BOOLEAN		RP error Protocol Error Unspecified was sent
TCV_Pwrlvl	BITSTRING		to hold the power level
TCV_Pwrlvl_ho	BITSTRING		power level for HO_CMD, used in HO cases.
TCV_RaCh	LOGICCH		random access channel parameter.
TCV_Rchr	B_2		to hold Radio channel requirement
TCV_Res	BOOLEAN		to hold the result of a test suite operation
TCV_Rpmr	MR		RP message reference for SMS
TCV_Rqr	RQR		to hold the request reference
TCV_Rqr9	RQR		to hold the request reference differing from TCV_Rqr
TCV_Rqr10	RQR		to hold the request reference differing from TCV_Rqr, TCV_Rqr9
TCV_Rqr11	RQR		to hold the request reference differing from TCV_Rqr, TCV_Rqr9, TCV_Rqr10
TCV_Rr	BITSTRING		to hold the random request reference
TCV_Rr1	BITSTRING		to hold the random reference
TCV_Service	SERVICES		to hold basicservice selected
TCV_Service1	SERVICES		to hold basicservice selected
TCV_Setup_mo	SETUP_MO_PDU		to hold the SETUP PDU (MO)
TCV_Setup_mo1	SETUP_MO_PDU		to hold the SETUP PDU (MO)
TCV_Setup_mt	SETUP_MT_PDU		to hold the SETUP PDU (MT)
TCV_Setup_mt1	SETUP_MT_PDU		to hold the 2nd SETUP PDU (MT)

TCV_2ndtest	BOOLEAN		2nd test required in TC_11_1_1
TCV_EFR_svi1	B_4		Holds 1st Speech Verson Indication from received BCAP
TCV_EFR_svi2	B_4		Holds 2nd Speech Verson Indication from received BCAP
TCV_EFR_svi3	B_4		Holds 3rd Speech Verson Indication from received BCAP
TCV_Esetup	ESETUP_PDU		to hold the emergency SETUP PDU
TCV_Esetup1	ESETUP_PDU		to hold the emergency SETUP PDU
TCV_Ecall	BOOLEAN	FALSE	it is set to TRUE, if the service under test is emergency call
TCV_slot	SN		To hold the default slot used during the entire TC.
TCV_slt2	SN		To hold a second working slot number.
TCV_Sres	OCTETSTRING		to hold the SRES returned
TCV_sysinf5	SYSINFO5_PDU		to hold the SysInfo5 PDU
TCV_sysinfo5	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell A
TCV_sysinfo5bis	SYSINFO5bis_PDU		to hold the SysInfo5bis PDU for cell A
TCV_sysinfo5_B	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell B
TCV_sysinfo5_C	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell c
TCV_sysinfo5_D	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell D
TCV_sysinfo5_E	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell E
TCV_sysinfo5_F	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell F
TCV_sysinfo5_G	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell G
TCV_sysinfo5_H	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell H
TCV_sysinfo5bis_A	SYSINFO5bis_PDU		to hold the SysInfo5bis PDU for cell A(TC_26_10_2_4_1)
TCV_sysinfo5bis_B	SYSINFO5bis_PDU		to hold the SysInfo5bis PDU for cell B
TCV_sysinf6	SYSINFO6_PDU		to hold the SysInfo6 PDU
TCV_sysinfo6	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell A
TCV_sysinfo6_B	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell B
TCV_sysinfo6_C	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell C
TCV_sysinfo6_D	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell D
TCV_sysinfo6_E	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell E
TCV_sysinfo6_F	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell F
TCV_sysinfo6_G	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell G
TCV_sysinfo6_H	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell H
TCV_temp	INTEGER		to hold results of expressions that are to be passed as parameters
TCV_ti_v	TI_V		to hold the transaction id. value
TCV_ti_v_2	TI_V		to hold the transaction id. value for another transaction
TCV_T	INTEGER		to hold Tx-integer chosen randomly
TCV_T_TwiceTC1M	INTEGER	TSPX_TC1M * 2000	Twice TC1M in ms
TCV_chdescr_arfcn	INTEGER		ARFCN carrier of actual cell
TCV_tch_arfcn	INTEGER		ARFCN carrier for traffic channel
TCV_asscmd_ts	BITSTRING		Time slot for channel description in assignment cmd.

TCV_ts	BITSTRING		Time slot for channel description in handover cmd.
TCV_ia_ts	BITSTRING		Time slot for channel description in immediate assignment
TCV_tsc	TSC		To hold the default training sequence used during the entire TC.
TCV_Td	OCTETSTRING		to hold timing difference
TCV_Time	INTEGER		to hold the measured value of T310 or T305 or T308
TCV_TI	TI		to hold the transaction ID for MO calls in sending evets
TCV_TI0	TI		to hold the transaction ID for MO calls in receiving evets
TCV_TI1	TI		to hold the transaction ID
TCV_TI2	TI		to hold the transaction ID
TCV_TI3	TI		to hold the transaction ID
TCV_tmp	INTEGER		temporary integer variable
TCV_Tx	BITSTRING		to hold Tx bitstring
TCV_S	INTEGER		to hold the S parameter
TCV_Upd	BOOLEAN		to hold the information whether the MS is updated or not,.
TCV_UssdString	IA5String		String for USS Data
TCV_UssdString1	IA5String		String for USS Data
TCV_UssdString2	IA5String		String for USS Data
TCV_InvkId	OCTETSTRING		to hold SS transaction ID
TCV_InvkId0	OCTETSTRING		to hold SS transaction ID
TCV_InvkId1	OCTETSTRING		to hold SS transaction ID
TCV_Comp	Components		to hold SS Components
TCV_Strt	STRT		to hold starting time
TCV_CBch	LOGICCH	"dummy"	Cell Broadcast channel
TCV_Tpmr	MR		TP message reference
TCV_TPOA1	BCDN		TP originating address digits
TCV_TPOA2	BCDN		TP originating address digits
TCV_TPDA	BCDN		TP destination address digits
TCV_RPOA_MT	BCDN		RP originating address digits for MT short messages
TCV_RPOA_MO	BCDN		RP originating address digits for MO short messages
TCV_RPOA1	BCDN		RP originating address digits for MT short messages
TCV_RPOA2	BCDN		RP originating address digits for MT short messages
TCV_RPDA_MT	BCDN		RP destination address digits for MT short messages
TCV_RPDA_MO	BCDN		RP destination address digits for MO short messages
TCV_SMTypeM	INTEGER		Replace short message type
TCV_SMTypeN	INTEGER		Replace short message type
TCV_SMcntns	OCTETSTRING		short message contents
TCV_SMSCBpack	SMSCBpack		CB message contents
TCV_freq	FRQPARA		to hold Freq type constraints
TCV_srv	SERVICES		to hold MT service
TCV_ur	B_4		to hold user rate
TCV_itc	B_3		to hold information transfer capability
TCV_sacp	B_3		to hold service access protocol
TCV_ce	B_2		to hold connection element
TCV_ce1	B_2		to hold connection element
TCV_ce3	B_2		to hold connection element
TCV_sa	B_1		to hold synchronous/asynchronous
TCV_ir	B_2		to hold intermediate rate
TCV_uil1p	B_2		to hold user information layer 1 protocol
TCV_modemt	B_5		to hold modem type
TCV_MODEM	MODEMTYPE		
TCV_ur2	B_4		to hold user rate for second

			configuration
TCV_itc2	B_3		to hold information transfer capability for second configuration
TCV_sacp2	B_3		to hold service access protocol for second configuration
TCV_strc	B_2		to hold structure
TCV_sa2	B_1		to hold synchronous/asynchronous for second configuration
TCV_supported	BOOLEAN	FALSE	used for indicating a selected service is declared supported or not
TCV_SpecialCase	BOOLEAN	FALSE	used for indicating whether a test case tests non-supported service
Detailed Comments:			

PCO declarations

PCO Declarations			
PCO Name	**Type**	**Role**	**Comments**
L	SAP0_3	LT	
Detailed Comments:	1. The PCO consists of multiple SAPs: the SAP0 and SAP3. 2. The lower tester (LT) is the user of the data link layer service. 3. The SAP0 at the lower tester controlling and observing the exchange of CC, MM, RR and SS PDUs (messages) on the DCCH, SACCH and/or RACH, BCCH, CCCH channels. 4. The SAP3 at the lower tester controlling and observing the exchange of SMS PDUs (messages) on the DCCH or SACCH channels.		

Timer declarations

Timer Declarations			
Timer Name	**Duration**	**Unit**	**Comments**
T_dly		ms	general purpose delay timer
T_dly1		ms	general purpose delay timer
T_dly2		ms	general purpose delay timer
T_dlyAss	5000	ms	AssCmd timer, 1)
T_guard	300	s	guard timer
T_release	C_release_time	s	release timer
Detailed Comments:	1) In order to give to the lower tester enough time to dequeue and to send out the call SETUP and the other call establishment related messages in the sending buffer before starting sending of ASSINGMENT COMMAND message, the default value 5s is used in the test step AssCh_complete. In the test step AssCh_failure the timer value is set to 4s. For the TC_26_6_4_1, 600ms is used.		

ASP type definitions

TTCN ASP Type definitions

ASP Type Definition		
ASP Name:	DL_EstRq (DL_ESTABLISH_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the establishment of multiple frame operation (L3 -> L2). The normal establishment procedure is initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstIn (DL_ESTABLISH_INDICATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation (L2 -> L3). The normal establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInCmsRq (DL_ESTABLISH_INDICATION_CM_SERVICE_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the MM CM SERVICE REQUEST message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	CMS_RQ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInLupRq (DL_ESTABLISH_INDICATION_LOCATION_UPDATING_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the MM LOCATION UPDATING REQUEST message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	LUP_RQ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInImsidIn (DL_ESTABLISH_INDICATION_IMSI_DETACH_INDICATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the MM IMSI DETACH INDICATION message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	IMSID_IN_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInPgRes (DL_ESTABLISH_INDICATION_PAGING_RESPONSE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the RR PAGING RESPONSE message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	PG_RES_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInCmreRq (DL_ESTABLISH_INDICATION_CM_REESTABLISHMENT_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the MM CM REESTABLISHMENT REQUEST message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	CMRE_RQ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstCo (DL_ESTABLISH_CONFIRM)	
PCO Type:	SAP0_3	
Comments:	The ASP is used by the L2 to inform the L3 about the establishment of multiple frame link (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_RelIn (DL_RELEASE_INDICATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the termination of an established multiple frame operation or to report an unsuccessful establishment attempt (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
release_mode	RelMode	
outstanding_indicator	BOOLEAN	
fn	FN	
Detailed Comments:	The outstanding indicator indicates whether or not there are outstanding acknowledgements or unsolved DL_DATA_REQUEST primitives.	

ASP Type Definition		
ASP Name:	DL_RacInChRq (DL_RANDOM_ACCESS_INDICATION_CHANNEL_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the arrival of an RR CHANNEL REQUEST message (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
fn	FN	
msg	CH_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_RacInHoacc (DL_RADOM_ACCESS_INDICATION_HANDOVER_ACCESS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the arrival of an RR HANDOVER ACCESS message (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HOACC_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmass (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	IMMASS_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmass_sp (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT message using unacknowledged operation (L3 -> L2) on the MS paging channel.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	IMMASS_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmassx (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT_EXTENDED)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT EXTENDED message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	IMMASSX_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmassx_sp (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT_EXTENDED)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT EXTENDED message using unacknowledged operation (L3 -> L2) on the MS paging channel.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	IMMASSX_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmassRej (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT REJECT message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	IMMASS_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmassRej_sp (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT REJECT message using unacknowledged operation (L3 -> L2) on the MS paging channel.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	IMMASS_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatInMsrRpt (DL_DATA_INDICATION_MEASUREMENT_REPORT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR MEASUREMENT REPORT message using unacknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MSR_RPT_PDU	
fn	FN	
Detailed Comments:	The ASPs are continuously received during the testing.	

ASP Type Definition		
ASP Name:	DL_UdatRqSMSCBData (DL_UNIT_DATA_REQUEST_SMSCB_DATA)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the SMSCB data using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SMSCB_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqPg1Rq (DL_UNIT_DATA_REQUEST_PAGING_REQUEST_TYPE1)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR PAGING REQUEST TYPE 1 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	PG1_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqPg2Rq (DL_UNIT_DATA_REQUEST_PAGING_REQUEST_TYPE2)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR PAGING REQUEST TYPE 2 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	PG2_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqPg3Rq (DL_UNIT_DATA_REQUEST_PAGING_REQUEST_TYPE3)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR PAGING REQUEST TYPE 3 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	PG3_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSchinfo (DL_UNIT_DATA_REQUEST_SYNCHRONIZATION_CHANNEL_INFORMATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYNCHRONIZATION CHANNEL INFORMATION message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SCHINFO_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo1 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE1)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 1 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO1_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo1_nh (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE1)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 3 message in case of non hopping	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO3_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo2 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE2)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 2 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO2_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo2bis (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE2bis)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 2bis message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO2bis_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo2ter (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE2ter)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 2ter message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO2ter_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo3 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE3)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 3 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO3_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo4 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE4)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 4 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO4_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo5 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE5)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 5 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO5_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo5bis (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE5bis)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 5bis message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO5bis_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo5ter (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE5ter)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 5ter message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO5ter_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo6 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE6)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 6 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO6_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo7 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE7)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 7 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO7_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo8 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE8)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 8 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO8_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatInClmChn (DL_UdatINDICATION_CLASSMARK_CHANGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR CLASSMARK CHANGE message using unacknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CLM_CHN_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAssCmd (DL_DATA_REQUEST_ASSIGNMENT_COMMAND)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR ASSIGNMENT COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ASS_CMD_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInAssCom (DL_DATA_INDICATION_ASSIGNMENT_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR ASSIGNMENT COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ASS_COM_PDU	
fn	FN	
Detailed Comments:	The ASP is a result of the ASP DL_RESUME_REQUEST of the SUT sending the ASSIGNMENT COMPLETE message.	

ASP Type Definition		
ASP Name:	DL_DatInAssfl (DL_DATA_INDICATION_ASSIGNMENT_FAILURE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR ASSIGNMENT FAILURE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ASSFL_PDU	
fn	FN	
Detailed Comments:	The ASP is a result of the ASP DL_RECONNECT_REQUEST of the SUT sending the ASSIGNMEN FAILURE message.	

ASP Type Definition		
ASP Name:	DL_DatInCmsRq (DL_DATA_INDICATION_CM_SERVICE_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM CM SERVICE REQUEST message (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CMS_RQ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqChmmo (DL_DATA_REQUEST_CHANNEL_MODE_MODIFY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CHANNEL MODE MODIFY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CHMMO_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInChmmoAck (DL_DATA_INDICATION_CHANNEL_MODE_MODIFY_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR CHANNEL MODE MODIFY ACKNOWLEDGE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CHMMO_ACK_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqChRel (DL_DATA_REQUEST_CHANNEL_RELEASE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CHANNEL_RELEASE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CH_REL_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqChRelErr (DL_DATA_REQUEST_CHANNEL_RELEASE_ERROR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CHANNEL_RELEASE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CH_REL_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCphmCmd (DL_DATA_REQUEST_CIPHERING_MODE_COMMAND)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CIPHERING_MODE_COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPHM_CMD_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCphmCmdErr (DL_DATA_REQUEST_CIPHERING_MODE_COMMAND_ERROR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the invalid RR CIPHERING_MODE_COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPHM_CMD_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCphmCom (DL_DATA_INDICATION_CIPHERING_MODE_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR CIPHERING MODE COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPHM_COM_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInClmChn (DL_DAT_INDICATION_CLASSMARK_CHANGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR CLASSMARK CHANGE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CLM_CHN_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqClmEnq (DL_DATA_REQUEST_CLASSMARK_ENQUIRY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CLASSMARK ENQUIRY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CLM_ENQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqFrqre (DL_DATA_REQUEST_FREQUENCY_REDEFINITION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR FREQUENCY REDEFINITION message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	FRQRE_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqHoCmd (DL_DATA_REQUEST_HANDOVER_COMMAND)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR HANDOVER COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HO_CMD_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInHoCom (DL_DATA_INDICATION_HANDOVER_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR HANDOVER COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HO_COM_PDU	
fn	FN	
Detailed Comments:	The ASP is a result of the ASP DL_RESUME_REQUEST of the SUT sending the HANDOVER COMPLETE message.	

ASP Type Definition		
ASP Name:	DL_DatInHofl (DL_DATA_INDICATION_HANDOVER_FAILURE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR HANDOVER FAILURE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HOFL_PDU	
fn	FN	
Detailed Comments:	The ASP is a result of the ASP DL_RECONNECT_REQUEST of the SUT sending the HANDOVER FAILURE message.	

ASP Type Definition		
ASP Name:	DL_DatRqPhyinfo (DL_DATA_REQUEST_PHYSICAL_INFORMATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR PHYSICAL INFORMATION message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	PHYINFO_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInRrst (DL_DATA_INDICATION_RR_STATUS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR STATUS message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	RRST_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAuthRej (DL_DATA_REQUEST_AUTHENTICATION_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM AUTHENTICATION REJECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	AUTH_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAuthRq (DL_DATA_REQUEST_AUTHENTICATION_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM AUTHENTICATION REQUEST message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	AUTH_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInAuthRes (DL_DATA_INDICATION_AUTHENTICATION_RESPONSE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM AUTHENTICATION RESPONSE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	AUTH_RES_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCmsAcp (DL_DATA_REQUEST_CM_SERVICE_ACCEPT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM CM SERVICE ACCEPT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CMS_ACP_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCmsRej (DL_DATA_REQUEST_CM_SERVICE_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM CM SERVICE REJECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CMS_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAbrt (DL_DATA_REQUEST_ABORT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM ABORT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ABRT_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqIdRq (DL_DATA_REQUEST_IDENTITY_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM IDENTITY REQUEST message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ID_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInIdRes (DL_DATA_INDICATION_IDENTIFY_RESPONSE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM IDENTITY RESPONSE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ID_RES_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqLupAcp (DL_DATA_REQUEST_LOCATION_UPDATING_ACCEPT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM LOCATION UPDATING ACCEPT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	LUP_ACP_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqLupAcpErr (DL_DATA_REQUEST_LOCATION_UPDATING_ACCEPT_ERR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM LOCATION UPDATING ACCEPT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	LUP_ACP_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqLupRej (DL_DATA_REQUEST_LOCATION_UPDATING_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM LOCATION UPDATING REJECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	LUP_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInMmst (DL_DATA_INDICATION_MM_STATUS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM STATUS message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MMST_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqTmsireCmd (DL_DATA_REQUEST_TMSI_REALLOCATION_COMMAND)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM TMSI REALLOCATION COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	TMSIRE_CMD_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInTmsireCom (DL_DATA_INDICATION_TMSI_REALLOCATION_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM TMSI REALLOCATION COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	TMSIRE_COM_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAlert (DL_DATA_REQUEST_ALERTING)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC ALERTING message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ALERT_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInAlert (DL_DATA_INDICATION_ALERTING)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC ALERTING message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ALERT_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCallCo (DL_DATA_INDICATION_CALL_CONFIRMED)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC CALL CONFIRMED message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CALL_CO_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCallProc (DL_DATA_REQUEST_CALL_PROCEEDING)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC CALL PROCEEDING message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CALL_PROC_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqConn (DL_DATA_REQUEST_CONNECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC CONNECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqConnErr (DL_DATA_REQUEST_CONNECT_ERR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC CONNECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInConn (DL_DATA_INDICATION_CONNECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC CONNECT message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqConnAck (DL_DATA_REQUEST_CONNECT_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC CONNECT ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInConnAck (DL_DATA_INDICATION_CONNECT_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the reception of the CC CONNECT ACKNOWLEDGE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_ACK_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqDisc (DL_DATA_REQUEST_DISCONNECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC DISCONNECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	DISC_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqDiscErr (DL_DATA_REQUEST_DISCONNECT_ERR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC DISCONNECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	DISC_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInDisc (DL_DATA_INDICATION_DISCONNECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC DISCONNECT message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	DISC_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInESetup (DL_DATA_INDICATION_EMERGENCY_SETUP)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC EMERGENCY_SETUP message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ESETUP_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqFac (DL_DATA_REQUEST_FACILITY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC FACILITY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	FAC_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInFac (DL_DATA_INDICATION_FACILITY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC FACILITY message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	FAC_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInHold (DL_DATA_INDICATION_HOLD)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC HOLD message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HOLD_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqHoldAck (DL_DATA_REQUEST_HOLD_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC HOLD ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HOLD_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqModify (DL_DATA_REQUEST_MODIFY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC MODIFY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInModify (DL_DATA_INDICATION_MODIFY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC MODIFY message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqModifyCom (DL_DATA_REQUEST_MODIFY_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC MODIFY COMPLETE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_COM_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqModifyRej (DL_DATA_REQUEST_MODIFY_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC MODIFY REJECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInModifyRej (DL_DATA_INDICATION_MODIFY_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC MODIFY REJECT message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_REJ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqNotify (DL_DATA_REQUEST_NOTIFY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC NOTIFY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	NOTIFY_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqProg (DL_DATA_REQUEST_PROGRESS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC PROGRESS message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	PROG_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqRegister (DL_DATA_REQUEST_REGISTER)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the SS REGISTER message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REGISTER_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInRegister (DL_DATA_INDICATION_REGISTER)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to receive the transmission of the SS REGISTER message using acknowledged operation.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REGISTER_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqRel (DL_DATA_REQUEST_RELEASE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC RELEASE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REL_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInRel (DL_DATA_INDICATION_RELEASE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC RELEASE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REL_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqRelCmp (DL_DATA_REQUEST_RELEASE_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC RELEASE COMPLETE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REL_COM_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInRelCmp (DL_DATA_INDICATION_RELEASE_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC RELEASE COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REL_COM_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqSetup (DL_DATA_REQUEST_SETUP)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC SETUP message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SETUP_MT_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInSetup (DL_DATA_INDICATION_SETUP)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC SETUP message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SETUP_MO_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInStartDtmf (DL_DATA_INDICATION_START_DTMF)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC START_DTMF message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	START_DTMF_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqStartDtmfAck (DL_DATA_REQUEST_START_DTMF_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC START DTMF ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	START_DTMF_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqStartDtmfRej (DL_DATA_REQUEST_START_DTMF_REJCT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC START DTMF REJCT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	START_DTMF_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInStopDtmf (DL_DATA_INDICATION_STOP_DTMF)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC STOP_DTMF message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	STOP_DTMF_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqStopDtmfAck (DL_DATA_REQUEST_STOP_DTMF_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC STOP DTMF ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	STOP_DTMF_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCcst (DL_DATA_REQUEST_CC_STATUS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC STATUS message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CCST_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCcst (DL_DATA_INDICATION_CC_STATUS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC STATUS message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CCST_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCcstEnq (DL_DATA_REQUEST_CC_STATUS_ENQ)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC STATUS_ENQ message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CCST_ENQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqUndefCC (DL_DATA_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to transmit any message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqUndefMM (DL_DATA_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to transmit any message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ID_RES_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqUndefRR (DL_DATA_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to transmit any message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	PART_REL_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqUnknown (DL_DATA_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to transmit any message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CCST_ENQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCpData (DL_DATA_INDICATION_CP_DATA)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the SMS CP DATA message using acknowledged operation (L2 -> L3) for MT.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CP_DATA_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCpData (DL_DATA_REQUEST_CP_DATA)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the SMS CP DATA message using acknowledged operation (L3 -> L2) for MO.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CP_DATA_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCpDataAck (DL_DATA_INDICATION_CPDATA_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the SMS CP DATA message using acknowledged operation (L2 -> L3) .	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPDATA_ACK_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCpDataAck (DL_DATA_REQUEST_CPDATA_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CP DATA ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPDATA_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCpError (DL_DATA_REQUEST_CP_ERROR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the SMS CP ERROR message using acknowledged operation (L3 -> L2) for MO.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPERR_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	MDL_RelRq (MDL_RELEASE_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the local end termiination of a previous established acknowledged mode operation (L3 -> L2). The local end termination procedure is initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
release_mode	RelMode	
Detailed Comments:		

PDU type definitions

TTCN PDU Type definitions

PDU Type Definition		
PDU Name:	ASS_CMD_PDU	
PCO Type:	SAP0_3	
Comments:	RR ASSSIGNMENT COMMAND n -> ms GSM 04.08, 9.1.2	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
ch1d_at	CHD	
pcmd	PCMD	
frql_at	FRQL	
cchd	CCHD	
ch1mod	CHMOD	
ch2d_at	CHD	
ch2mod	CHMOD	
ma_at	MA	
strt	STRT	
frql_bt	FRQL	
ch1d_bt	CHD	
ch2d_bt	CHD	
frqchs_bt	FRQCHS	
ma_bt	MA	
cphms	CPHMS	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ASS_COM_PDU	
PCO Type:	SAP0_3	
Comments:	RR ASSSIGNMENT COMPLETE ms -> n GSM 04.08, 9.1.3	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ASSFL_PDU	
PCO Type:	SAP0_3	
Comments:	RR ASSSIGNMENT FAILURE ms -> n GSM 04.08, 9.1.4	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CHMMO_PDU	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL MODE MODIFY n -> ms GSM 04.08, 9.1.5	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
chd	CHD	
chmod	CHMOD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CHMMO_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL MODE MODIFY ACKNOWLEDGE ms -> n GSM 04.08, 9.1.6	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
chd	CHD	
chmod	CHMOD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CH_REL_PDU	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL RELEASE n -> ms GSM 04.08, 9.1.7	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CH_REL_PDU_ERR	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL RELEASE containing additional unknown IE n -> ms GSM 04.08, 9.1.7	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
add	OCTETSTRING	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CH_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL REQUEST ms -> n GSM 04.08, 9.1.8	
Field Name	**Field Type**	**Comments**
ecau_rrf	BITSTRING [8]	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CPHM_CMD_PDU	
PCO Type:	SAP0_3	
Comments:	RR CIPHERING MODE COMMAND n -> ms GSM 04.08, 9.1.9	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
cph_res	CPH_RES	
cphms	CPHMS	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CPHM_CMD_PDU_ERR	
PCO Type:	SAP0_3	
Comments:	RR CIPHERING MODE COMMAND with additional unknown IE GSM 04.08, 9.1.9	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
cph_res	CPH_RES	
cphms	CPHMS	
add	OCTETSTRING	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CPHM_COM_PDU	
PCO Type:	SAP0_3	
Comments:	RR CIPHERING MODE COMPLETE ms -> n GSM 04.08, 9.1.10	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
mei	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CLM_CHN_PDU	
PCO Type:	SAP0_3	
Comments:	RR CLASSMARK CHANGE ms -> n GSM 04.08, 9.1.11	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
msclm	MSCLM2	
msclm_adi	MSCLM3	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CLM_ENQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR CLASSMARK ENQUIRY n -> ms GSM 04.08, 9.1.12	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	FRQRE_PDU	
PCO Type:	SAP0_3	
Comments:	RR FREQUENCY REDEFINITIONY n -> ms GSM 04.08, 9.1.13	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
chd	CHD	
ma	MA	
strt	STRT	
cchd	CCHD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HOACC_PDU	
PCO Type:	SAP0_3	
Comments:	RR HANDOVER ACCESS ms -> n GSM 04.08, 9.1.14	
Field Name	**Field Type**	**Comments**
horf	HORF	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HO_CMD_PDU	
PCO Type:	SAP0_3	
Comments:	RR HANDOVER COMMAND n -> ms GSM 04.08, 9.1.15	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
cd	CD	
ch1d_at	CHD	
horf	HORF	
pcmd	PCMD	
synchi	SYNCHI	
frqsl_at	FRQL	
frql_at	FRQL	
cchd	CCHD	
ch1mod	CHMOD	
ch2d_at	CHD	
ch2mod	CHMOD	
frqchs_at	FRQCHS	
ma_at	MA	
strt	STRT	
rtdif	TDIF	
ta	TA	
frqsl_bt	FRQL	
frql_bt	FRQL	
ch1d_bt	CHD	
ch2d_bt	CHD	
frqchs_bt	FRQCHS	
ma_bt	MA	
cphms	CPHMS	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HO_COM_PDU	
PCO Type:	SAP0_3	
Comments:	RR HANDOVER COMPLETE ms -> n GSM 04.08, 9.1.16	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
motdif	MTDIF	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HOFL_PDU	
PCO Type:	SAP0_3	
Comments:	RR HANDOVER FAILURE ms -> n GSM 04.08, 9.1.17	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	IMMASS_PDU	
PCO Type:	SAP0_3	
Comments:	RR IMMEDIATE ASSIGNMENT n -> ms GSM 04.08, 9.1.18	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
shoct	SHOCT	
pm	PM	
chd	CHD	
rqr	RQR	
ta	TA	
ma	MA	
strt	STRT	
iaroct	IARESTOCT	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	IMMASSX_PDU	
PCO Type:	SAP0_3	
Comments:	RR IMMEDIATE ASSIGNMENT n -> ms GSM 04.08, 9.1.19	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
shoct	SHOCT	
pm	PM	
chd1	CHD	
rqr1	RQR	
ta1	TA	
chd2	CHD	
rqr2	RQR	
ta2	TA	
ma	MA	
strt	STRT	
iaxroct	OCTETSTRING [1..4]	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	IMMASS_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	RR IMMEDIATE REJECT n -> ms GSM 04.08, 9.1.20	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
shoct	SHOCT	
pm	PM	
rqr1	RQR	
wi1	WI	
rqr2	RQR	
wi2	WI	
rqr3	RQR	
wi3	WI	
rqr4	RQR	
wi4	WI	
iarroct	OCTETSTRING [3]	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	MSR_RPT_PDU	
PCO Type:	SAP0_3	
Comments:	RR MEASUREMENT REPORT ms -> n GSM 04.08, 9.1.21	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
msrr	MSRR	
Detailed Comments:		

PDU Type Definition		
PDU Name:	PG1_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR PAGING REQUEST_TYPE1 n -> ms GSM 04.08, 9.1.22	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
chn_m1_2	CHNEED	
pm	PM	
mi1	MI	
mi2	MI	
p1roct	OCTETSTRING [0..17]	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	PG2_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR PAGING REQUEST TYPE2 n -> ms GSM 04.08, 9.1.23	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
chn_m1_2	CHNEED	
pm	PM	
mi1	TMSI	
mi2	TMSI	
mi3	MI	
p2roct	OCTETSTRING [1..11]	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	PG3_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR PAGING REQUEST TYPE3 n -> ms GSM 04.08, 9.1.24	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
chn_m1_2	CHNEED	
pm	PM	
mi1	TMSI	
mi2	TMSI	
mi3	TMSI	
mi4	TMSI	
p3roct	OCTETSTRING [3]	
Detailed Comments:	The message has an L2 pseudo length of 19 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	PG_RES_PDU	
PCO Type:	SAP0_3	
Comments:	RR PAGING RESPONSE ms -> n GSM 04.08, 9.1.25	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
shoct	SHOCT	
cphksn	CPHKSN	
msclm	MSCLM2	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	PART_REL_PDU	
PCO Type:	SAP0_3	
Comments:	RR PARTIAL RELEASE n -> ms GSM 04.08, 9.1.26	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
chd	MAC	
Detailed Comments:		

PDU Type Definition		
PDU Name:	PHYINFO_PDU	
PCO Type:	SAP0_3	
Comments:	RR PHYSICAL INFORMATION n -> ms GSM 04.08 clause 9.1.28	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
ta	TA	
Detailed Comments:		

PDU Type Definition		
PDU Name:	RRST_PDU	
PCO Type:	SAP0_3	
Comments:	RR STATUS n <-> ms GSM 04.08, 9.1.29	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SCHINFO_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYNCHRONIZATION CHANNEL INFORMATION n -> ms GSM 04.08, 9.1.30	
Field Name	**Field Type**	**Comments**
ncc	NCC	
bcc	BCC	
t1	T1	
t2	T2	
t3_	T3_	
Detailed Comments:	SCHINFO_PDU has a total lengh of 25 bits (GSM 04.04).	

PDU Type Definition		
PDU Name:	SYSINFO1_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE1 n -> ms GSM 04.08, 9.1.31	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
cchd	CCHD	
rachcp	RACHCP	
si1roct	OCTETSTRING [1]	
Detailed Comments:	The message has an L2 pseudo length of 21 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO2_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE2 n -> ms GSM 04.08, 9.1.32	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
bcchfl	NCD	
nccp	NCCP	
rachcp	RACHCP	
Detailed Comments:	The message has an L2 pseudo length of 22 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO2bis_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE2bis or 2ter n -> ms GSM 04.08, 9.1.33, 9.1.33a	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
xbcchfl	NCD	
rachcp	RACHCP	
si2bisroct	OCTETSTRING [1..4]	
Detailed Comments:	This PDU type is used both for SYSINFO2bis, as well as for 2ter. The 2bis message has an L2 pseudo length of 21 octets and a total length of 23 octets. The 2ter message has an L2 pseudo length of 18 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO2ter_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION 2ter n -> ms GSM 04.08, 9.1.34	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
xbcchfl	NCD2	
si2terroct	OCTETSTRING[4]	
Detailed Comments:	The 2ter message has an L2 pseudo length of 18 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO3_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE3 n -> ms GSM 04.08, 9.1.34	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
ci	CI	
lai	LAI	
ccd	CCD	
co	CO	
csp	CSP	
rachcp	RACHCP	
si3roct	SI3RO	
Detailed Comments:	The message has an L2 pseudo length of 18 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO4_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE4 n -> ms GSM 04.08, 9.1.35	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
lai	LAI	
csp	CSP	
rachcp	RACHCP	
cbchd	CHD	
cbchma	MA	
si4roct	SI4RO	
Detailed Comments:	The message has a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO5_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE5 n -> ms GSM 04.08, 9.1.36	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
bcchfl	NCD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SYSINFO5bis_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE5bis n -> ms GSM 04.08, 9.1.38	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
xbcchfl	NCD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SYSINFO5ter_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE5ter n -> ms GSM 04.08, 9.1.39	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
xbcchfl	NCD2	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SYSINFO6_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE6 n -> ms GSM 04.08, 9.1.38	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
ci	CI	
lai	LAI	
co	CO	
nccp	NCCP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SYSINFO7_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE7 n -> ms GSM 04.08, 9.1.41	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
si7roct	SI7RO	
Detailed Comments:	The message has a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO8_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE8 n -> ms GSM 04.08, 9.1.42	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
si8roct	SI7RO	
Detailed Comments:	The message has a total length of 23 octets.	

PDU Type Definition		
PDU Name:	ABRT_PDU	
PCO Type:	SAP0_3	
Comments:	MM ABORT n -> ms GSM 04.08, 9.2.8	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
rejcau	REJCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	AUTH_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	MM AUTHENTICATION REJECT n -> ms GSM 04.08, 9.2.1	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	AUTH_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM AUTHENTICATION REQUEST n -> ms GSM 04.08, 9.2.2	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
shoct	SHOCT	
cphksn	CPHKSN	
rand	RAND	
Detailed Comments:		

PDU Type Definition		
PDU Name:	AUTH_RES_PDU	
PCO Type:	SAP0_3	
Comments:	MM AUTHENTICATION RESPONSE ms -> n GSM 04.08, 9.2.3	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
sres	SRES	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CMRE_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM CM REESTABLISHMENT REQUEST ms -> n GSM 04.08, 9.2.4	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
shoct	SHOCT	
cphksn	CPHKSN	
msclm	MSCLM2	
mi	MI	
lai	LAI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CMS_ACP_PDU	
PCO Type:	SAP0_3	
Comments:	MM CM SERVICE ACCEPT n -> ms GSM 04.08, 9.2.5	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CMS_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	MM CM SERVICE REJECT n -> ms GSM 04.08, 9.2.6	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
mmcau	REJCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CMS_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM CM SERVICE REQUEST ms -> n GSM 04.08, 9.2.9	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
cphksn	CPHKSN	
svtype	CMSVTYPE	
msclm	MSCLM2	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ID_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM IDENTITY REQUEST n -> ms GSM 04.08, 9.2.10	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
shoct	SHOCT	
idtype	IDTYPE	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ID_RES_PDU	
PCO Type:	SAP0_3	
Comments:	MM IDENTITY RESPONSE ms -> n GSM 04.08, 9.2.11	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	IMSID_IN_PDU	
PCO Type:	SAP0_3	
Comments:	MM IMSI DETACH INDICATION ms -> n GSM 04.08, 9.2.12	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
msclm	MSCLM1	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	LUP_ACP_PDU	
PCO Type:	SAP0_3	
Comments:	MM LOCATION UPDATING ACCEPT n -> ms GSM 04.08, 9.2.13	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
lai	LAI	
mi	MI	
fop	IEI_8	
Detailed Comments:		

PDU Type Definition		
PDU Name:	LUP_ACP_PDU_ERR	
PCO Type:	SAP0_3	
Comments:	MM LOCATION UPDATING ACCEPT n -> ms GSM 04.08, 9.2.13	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
lai	LAI	
mi	MI	
dupmi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	LUP_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	MM LOCATION UPDATING REJECT n -> ms GSM 04.08, 9.2.14	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
rejcau	REJCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	LUP_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM LOCATION UPDATING REQUEST ms -> n GSM 04.08, 9.2.15	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
cphksn	CPHKSN	
lutype	LUT	
lai	LAI	
msclm	MSCLM1	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	MMST_PDU	
PCO Type:	SAP0_3	
Comments:	MM STATUS n <-> ms GSM 04.08, 9.2.16	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
rejcau	REJCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	TMSIRE_CMD_PDU	
PCO Type:	SAP0_3	
Comments:	MM TMSI REALLOCATION COMMAND n -> ms GSM 04.08, 9.2.17	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
lai	LAI	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	TMSIRE_COM_PDU	
PCO Type:	SAP0_3	
Comments:	MM TMSI REALLOCATION COMPLETE ms -> n GSM 04.08, 9.2.18	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ALERT_PDU	
PCO Type:	SAP0_3	
Comments:	CC ALERTING ms <-> n GSM 04.08, 9.3.1	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
fie	FIE	
pi	PI	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CALL_CO_PDU	
PCO Type:	SAP0_3	
Comments:	CC CALL CONFIRMED ms -> n GSM 04.08, 9.3.2	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcri	RPI	
bcap1	BCAP	
bcap2	BCAP	
cau	CAU	
cccap	CCCAP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CALL_PROC_PDU	
PCO Type:	SAP0_3	
Comments:	CC CALL PROCEEDING n -> ms GSM 04.08, 9.3.3	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcri	RPI	
bcap1	BCAP	
bcap2	BCAP	
fie	FIE	
pi	PI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CONN_PDU	
PCO Type:	SAP0_3	
Comments:	CC CONNECT n <-> ms GSM 04.08, 9.3.5	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
fie	FIE	
pi	PI	
cnn	CNN	
cns	CNS	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CONN_PDU_ERR	
PCO Type:	SAP0_3	
Comments:	CC CONNECT n <-> ms Used as invalid message	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
unknown	UNKWN	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CONN_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CC CONNECT ACKNOWLEDGE ms <-> n GSM 04.08, 9.3.6	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	DISC_PDU	
PCO Type:	SAP0_3	
Comments:	CC DISCONNECT ms <-> n (both directions) GSM 04.08, 9.3.7	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
fie	FIE	
pi	PI	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	DISC_PDU_ERR	
PCO Type:	SAP0_3	
Comments:		
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
unknown	UNKWN	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ESETUP_PDU	
PCO Type:	SAP0_3	
Comments:	CC EMERGENCY SETUP ms -> n GSM 04.08, 9.3.8	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcap	BCAP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	FAC_PDU	
PCO Type:	SAP0_3	
Comments:	CC FACILITY n <-> ms GSM 04.08, 9.3.9	
Field Name	**Field Type**	**Comments**
ti	TI	
cc_sspd	PD	
mt	MT	
fie	FIE	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HOLD_PDU	
PCO Type:	SAP0_3	
Comments:	CC HOLD ms -> n GSM 04.08, 9.3.10	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HOLD_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CC HOLD ACKNOWLEDGE n -> ms GSM 04.08, 9.3.11	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	MODIFY_PDU	
PCO Type:	SAP0_3	
Comments:	CC MODIFY ms <-> n GSM 04.08, 9.3.13	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcap	BCAP	
llcmp	LLCMP	
hlcmp	HLCMP	
rcsd	RCSD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	MODIFY_COM_PDU	
PCO Type:	SAP0_3	
Comments:	CC MODIFY COMPLETE ms <-> n GSM 04.08, 9.3.14	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcap	BCAP	
llcmp	LLCMP	
hlcmp	HLCMP	
rcsd	RCSD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	MODIFY_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	CC MODIFY REJECT ms <-> n GSM 04.08, 9.3.15	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcap	BCAP	
cau	CAU	
llcmp	LLCMP	
hlcmp	HLCMP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	NOTIFY_PDU	
PCO Type:	SAP0_3	
Comments:	CC NOTIFY ms <-> n GSM 04.08, 9.3.16	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
nti	NTI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	REGISTER_PDU	
PCO Type:	SAP0_3	
Comments:	SS REGISTER ms <-> n GSM 04.80, 2.4	
Field Name	**Field Type**	**Comments**
ti	TI	
sspd	PD	
mt	MT	
fie	FIE	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	PROG_PDU	
PCO Type:	SAP0_3	
Comments:	CC PROGRESS n -> ms GSM 04.08, 9.3.17	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
pi	PI	
uu	UU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	REL_PDU	
PCO Type:	SAP0_3	
Comments:	CC RELEASE n <-> ms GSM 04.08, 9.3.18	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
cau2	CAU	
fie	FIE	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	REL_COM_PDU	
PCO Type:	SAP0_3	
Comments:	CC or SS RELEASE COMPLETE n <-> ms GSM 04.08, 9.3.19; GSM 04.80, 2.5	
Field Name	**Field Type**	**Comments**
ti	TI	
cc_sspd	PD	
mt	MT	
cau	CAU	
fie	FIE	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SETUP_MO_PDU	
PCO Type:	SAP0_3	
Comments:	CC SETUP n <- ms GSM 04.08, 9.3.23.2	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcri	RPI	
bcap1	BCAP	
bcap2	BCAP	
fie	FIE	
cgps	CGPS	
cdpn	CDPN	
cdps	CDPS	
llcri	RPI	
llcmp1	LLCMP	
llcmp2	LLCMP	
hlcri	RPI	
hlcmp1	HLCMP	
hlcmp2	HLCMP	
uu	UU	
ssvi	SSVI	
clirsup	CLRSUP	
clirinv	CLRINV	
cccap	CCCAP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SETUP_MT_PDU	
PCO Type:	SAP0_3	
Comments:	CC SETUP n -> ms GSM 04.08, 9.3.23.1	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcri	RPI	
bcap1	BCAP	
bcap2	BCAP	
fie	FIE	
pi	PI	
sig	SIGNAL	
cgpn	CGPN	
cgps	CGPS	
cdpn	CDPN	
cdps	CDPS	
llcri	RPI	
llcmp1	LLCMP	
llcmp2	LLCMP	
hlcri	RPI	
hlcmp1	HLCMP	
hlcmp2	HLCMP	
uu	UU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SMSCB_PDU	
PCO Type:	SAP0_3	
Comments:	SMSCB message GSM 03.41, 9.3	
Field Name	**Field Type**	**Comments**
blocktype	BLOCKTYPE	
serial_number	SERIAL_NUMBER	
message_id	OCTETSTRING[2]	
dcs	TPDCS	
page_param	BITSTRING[8]	
message_contents	OCTETSTRING	
Detailed Comments:		

PDU Type Definition		
PDU Name:	START_DTMF_PDU	
PCO Type:	SAP0_3	
Comments:	CC START DTMF ms -> n GSM 04.08, 9.3.24	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
kpf	KPF	
Detailed Comments:		

PDU Type Definition		
PDU Name:	START_DTMF_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CC START DTMF ACKNOWLEDGE n -> ms GSM 04.08, 9.3.25	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
kpf	KPF	
Detailed Comments:		

PDU Type Definition		
PDU Name:	START_DTMF_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	CC START DTMF REJECT n -> ms GSM 04.08, 9.3.26	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CCST_PDU	
PCO Type:	SAP0_3	
Comments:	CC STATUS n <-> ms GSM 04.08, 9.3.27	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
cst	CST	
acst	ACST	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CCST_ENQ_PDU	
PCO Type:	SAP0_3	
Comments:	CC STATUS ENQUIRY ms <-> n GSM 04.08, 9.3.28	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	STOP_DTMF_PDU	
PCO Type:	SAP0_3	
Comments:	CC STOP DTMF ms -> n GSM 04.08, 9.3.29	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	STOP_DTMF_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CC STOP DTMF acknowledge n -> ms GSM 04.08, 9.3.30	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CP_DATA_PDU	
PCO Type:	SAP0_3	
Comments:	SMS CP DATA ms <-> n GSM 04.11, 7.2.1	
Field Name	**Field Type**	**Comments**
ti	TI	
smspd	PD	
mt	MT	
CPdata	CPDATA	
Detailed Comments:	CPDATA contains RPDU - RP_ACK or RP_ERROR, either returning to the MS in case of MO, or reporting the outcome of a MT messaging attempt in case of MT .	

PDU Type Definition		
PDU Name:	CPDATA_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CP DATA ACKNOWLEDGE ms <-> n GSM 04.11, 7.2.2	
Field Name	**Field Type**	**Comments**
ti	TI	
smspd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CPERR_PDU	
PCO Type:	SAP0_3	
Comments:	CP_ERROR n <-> ms GSM 04.11, 7.2.3	
Field Name	**Field Type**	**Comments**
ti	TI	
smspd	PD	
mt	MT	
cp_cause	CP_CAU	
Detailed Comments:	CP_UDAT contains RPDU, RP_ACK or RP_ERROR, either returning to the MS in case of MO, or reporting the outcome of a MT messaging attempt in case of MT .	

Constraints Part

Test suite type constraint declarations

Structured type constraint declarations

Structured Type Constraint		
Constraint Name:	EFR_Bcap_Speech_MO2(rcr:B_2)	
Structured Type:	BCAP	
Derivation Path:		
Comments:	speech bearer capability for direction ms->n. An EFR ms MUST send two 3a_etc (speech preference) octets.	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	'03'O	
extb3	'0'B	
rchr	rcr	
cs	'0'B	
tm	'0'B	
itc	'000'B	
oct_3a_etc_1	EFR_Bcap_3a_etc	
oct_3a_etc_2	EFR_Bcap_3a_etc	
oct_3a_etc_3	-	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	-	
spb	-	
strc	-	
dplxm	-	
config	-	
nirr	-	
est	-	
extb5	-	
accid	-	
ra	-	
sacp	-	
extb6	-	
l1id	-	
uil1	-	
sb	-	
extb6a	-	
nsb	-	
nb	-	
ndb	-	
ur	-	
extb6b	-	
ir	-	
nictx	-	
nicrx	-	
pi	-	
extb6c	-	
ce	-	
modemt	-	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	EFR_Bcap_Speech_MO3(rcr:B_2)	
Structured Type:	BCAP	
Derivation Path:		
Comments:	speech bearer capability for direction ms->n. An EFR ms must send THREE 3a_etc (speech preference) octets.	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	'04'O	
extb3	'0'B	
rchr	rcr	
cs	'0'B	
tm	'0'B	
itc	'000'B	
oct_3a_etc_1	EFR_Bcap_3a_etc	
oct_3a_etc_2	EFR_Bcap_3a_etc	
oct_3a_etc_3	EFR_Bcap_3a_etc	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	-	
spb	-	
strc	-	
dplxm	-	
config	-	
nirr	-	
est	-	
extb5	-	
accid	-	
ra	-	
sacp	-	
extb6	-	
l1id	-	
uil1	-	
sb	-	
extb6a	-	
nsb	-	
nb	-	
ndb	-	
ur	-	
extb6b	-	
ir	-	
nictx	-	
nicrx	-	
pi	-	
extb6c	-	
ce	-	
modemt	-	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	EFR_Bcap_Speech_MT	
Structured Type:	BCAP	
Derivation Path:		
Comments:	speech bearer capability for direction n->ms	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	'01'O	
extb3	'1'B	
rchr	'01'B	
cs	'0'B	
tm	'0'B	
itc	'000'B	
oct_3a_etc_1	-	
oct_3a_etc_2	-	
oct_3a_etc_3	-	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	-	
spb	-	
strc	-	
dplxm	-	
config	-	
nirr	-	
est	-	
extb5	-	
accid	-	
ra	-	
sacp	-	
extb6	-	
l1id	-	
uil1	-	
sb	-	
extb6a	-	
nsb	-	
nb	-	
ndb	-	
ur	-	
extb6b	-	
ir	-	
nictx	-	
nicrx	-	
pi	-	
extb6c	-	
ce	-	
modemt	-	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	EFR_Bcap_Speech_MT2	
Structured Type:	BCAP	
Derivation Path:		
Comments:	speech bearer capability for direction n->ms	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	'01'O	
extb3	'0'B	
rchr	'01'B	
cs	'0'B	
tm	'0'B	
itc	'000'B	
oct_3a_etc_1	EFR_Bcap_3a_etc	
oct_3a_etc_2	EFR_Bcap_3a_etc	
oct_3a_etc_3	-	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	-	
spb	-	
strc	-	
dplxm	-	
config	-	
nirr	-	
est	-	
extb5	-	
accid	-	
ra	-	
sacp	-	
extb6	-	
l1id	-	
uil1	-	
sb	-	
extb6a	-	
nsb	-	
nb	-	
ndb	-	
ur	-	
extb6b	-	
ir	-	
nictx	-	
nicrx	-	
pi	-	
extb6c	-	
ce	-	
modemt	-	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	EFR_Bcap_Speech_MT3	
Structured Type:	BCAP	
Derivation Path:		
Comments:	speech bearer capability for direction n->ms	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	'01'O	
extb3	'0'B	
rchr	'01'B	
cs	'0'B	
tm	'0'B	
itc	'000'B	
oct_3a_etc_1	EFR_Bcap_3a_etc	
oct_3a_etc_2	EFR_Bcap_3a_etc	
oct_3a_etc_3	EFR_Bcap_3a_etc	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	-	
spb	-	
strc	-	
dplxm	-	
config	-	
nirr	-	
est	-	
extb5	-	
accid	-	
ra	-	
sacp	-	
extb6	-	
l1id	-	
uil1	-	
sb	-	
extb6a	-	
nsb	-	
nb	-	
ndb	-	
ur	-	
extb6b	-	
ir	-	
nictx	-	
nicrx	-	
pi	-	
extb6c	-	
ce	-	
modemt	-	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	EFR_Bcap_3a_etc	
Structured Type:	EFR_BCAP_3A_ETC	
Derivation Path:		
Comments:	speech bearer capability for direction ms->n.	
Element Name	**Element Value**	**Comments**
extb	'?'B	
cod	'0'B	
spr2	'00'B	
svi	'0???'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_Omit	
Structured Type:	NCD	
Derivation Path:		
Comments:	Omit Bcch Frequency list (Neighbour cells description)	
Element Name	**Element Value**	**Comments**
rfl2	-	
extind	-	
baind	-	
rfl4	-	
rfl	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_01	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under GSM900 with the ARFCN list = {10, 20, 40, 80, 90, 100, 110, 120}.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'0'B	
baind	'0'B	
rfl4	'0000'B	
rfl	'80200802008000000008000080200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_02	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Alternative neighbour cells description for SYSTEM INFORMATION 2 and 5 under GSM900. These are ARFCNs 15, 85, 95, 105, 115, and 122	
Element Name	**Element Value**	**Comments**
rfl4	'0010'B	
rfl	'04010040100000000000000000004000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_03	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 for DCS1800 for cell B with the ARFCN list = {520, 600, 700, 780, 810, 870}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'04412C168E440000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_04	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Alternative neighbour cells description for SYSTEM INFORMATION 2 and 5 for DCS1800.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'09412C168E440000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_45	
Structured Type:	NCD	
Derivation Path:	BcchFreqLst_01.	
Comments:	Default neighbour cells description of cell B for SYSTEM INFORMATION 2 and 5 under GSM900.	
Element Name	**Element Value**	**Comments**
rfl	'80200802008000000000000000200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_47	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2bis and 5bis in cell B under EGSMwith the ARFCN list = {1005, 1010, 1015}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'0'B	
rfl4	'1101'B	
rfl	'F68AEC000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_48	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 for DCS1800 for cell B in RR, MM and SMS test cases with the ARFCN list = {520, 590, 600, 700, 780, 810, 870}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'04411307BB00C80000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CallCntrlCap	
Structured Type:	CCCAP	
Derivation Path:		
Comments:	CC state 'st'	
Element Name	**Element Value**	**Comments**
iei	'00010101'B	
iel	'01'O	
spr	'0000000'B	
dtmf	TSPX_DTMF	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_01	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with cause value = #16, location = user.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	'0000'B	
extb3a	-	
rec	-	
extb4	'1'B	
cau_class	'001'B	
cau_va	'0000'B	
cau_di	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Cause_Def	
Structured Type:	CAU	
Derivation Path:		
Comments:	cause information element with any valid cause value, any valid location.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	?	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
location	('0000'B, '0001'B, '0010'B, '0011'B, '0100'B, '0101'B, '0111'B, '1010'B)	
extb3a	-	
rec	-	
extb4	?	
cau_class	?	
cau_va	?	
cau_di	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_Omit	
Structured Type:	CCHD	
Derivation Path:		
Comments:	Omit descriptor	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_02	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in RR testing of GSM 900 with the ARFCN_list={20,30,50,70}.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'00000000000000200002000020080000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_03	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in RR testing of DCS1800 with the ARFCN list = {590, 650, 750, 850}..	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'89272827190000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_04	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell B in RR testing of GSM 900.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'00000000000000000000000000000200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_20_Aman	
Structured Type:	CCHD	
Derivation Path:		
Comments:	Cell Channel Description for any cell in RR testing of GSM 900 HO cases with the ARFCN_list={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'0002080000000F020408320202090200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_201_Ad	
Structured Type:	CCHD	
Derivation Path:		
Comments:	Cell Channel Description for cell A in RR testing of DCS1800 HO cases with the ARFCN_list={734,741,747,754,759,762,766,767,773,775,779,782,791,798,829,832,844} using 256 format.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'8B6F14F32FC602C59EFFA54999400000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_202_Adman	
Structured Type:	CCHD	
Derivation Path:		
Comments:	Cell Channel Description for cell A in RR testing of DCS1800 HO cases with the Complete Cell Allocation of Cell A in HO cases using 512 format. ARFCN_list={734,741,747,754,759,762,766,767,773,775,779,782,791,798,829,832,844}	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'896F0A7CC5FC700A8B9F7FF452463340'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_20_Bman	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell B in RR testing of GSM 900 HO cases with the complete Cell Allocation ARFCN_list={14,18,22,24,30,31,38,40,60,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'0002080000000F02080000A060A22000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_201_Bd	
Structured Type:	CCHD	
Derivation Path:		
Comments:	Cell Channel Description for cell B in RR testing of DCS1800 HO cases with the ARFCN_list={739,743,746,749,756,758,761,764,771,779,782,791,798,829,832,844} using 256 format.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'8A71907137B602C5DEF7A348D8000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellDescrp(ncc1, bcc1:B_3; par_arfcn: INTEGER)	
Structured Type:	CD	
Derivation Path:		
Comments:	BCCH channel for Cell A	
Element Name	**Element Value**	**Comments**
bcch_arfcn_h	OC_MostBits(INT_TO_BIT(par_arfcn, 10), 2)	
ncc	ncc1	
bcc	bcc1	
bcch_arfcn_l	OC_LeastBits(INT_TO_BIT(par_arfcn, 10), 8)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellOpt_01	
Structured Type:	CO	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
sprb	'0'B	
pwrc	'0'B	
dtx	'10'B	
rlt	'0001'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellSelPara(crh, mtmc:INTEGER; neci:B_1)	
Structured Type:	CSP	
Derivation Path:		
Comments:	Default value.	
Element Name	**Element Value**	**Comments**
crh	INT_TO_BIT(crh, 3)	
mtmc	INT_TO_BIT(mtmc, 5)	
acs	'0'B	
neci	neci	
ram	'000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChDescrp_fh(par_chtype: CH_TDMA; ts_ccch: SN; tsc:TSC; maio:MAIO; hsn:HSN)	
Structured Type:	CHD	
Derivation Path:		
Comments:	A derived constraint for <<Channel description >>ie with TCH/F freq. hopping in any CELL.	
Element Name	**Element Value**	**Comments**
iei	-	
cht_schn	par_chtype	
tn	ts_ccch	
tsc	tsc	
hch	'1'B	
maio	maio	
hsn	hsn	
spr	-	
arfcn	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChDescrp_nfh(type:CH_TDMA; slot:SN; tsc:TSC; par_arfcn: INTEGER)	
Structured Type:	CHD	
Derivation Path:		
Comments:	Channel description for CC testing, non hopping, no starting time.	
Element Name	**Element Value**	**Comments**
iei	-	
cht_schn	type	
tn	slot	
tsc	tsc	
hch	'0'B	
maio	-	
hsn	-	
spr	'00'B	
arfcn	INT_TO_BIT(par_arfcn, 10)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod(mode:B_8)	
Structured Type:	CHMOD	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
mode	mode	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_omit	
Structured Type:	CHMOD	
Derivation Path:		
Comments:	parametrised mode.	
Element Name	**Element Value**	**Comments**
iei	-	
mode	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_EFR2	
Structured Type:	CHMOD	
Derivation Path:	ChMod_sign.	
Comments:	enhanced full/half rate speech - version 2.	
Element Name	**Element Value**	**Comments**
mode	C_ChMod_r2	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_EFR2_iei	
Structured Type:	CHMOD	
Derivation Path:		
Comments:	Version 2 full rate speech.	
Element Name	**Element Value**	**Comments**
iei	'01100011'B	
mode	C_ChMod_r2	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_EFR3	
Structured Type:	CHMOD	
Derivation Path:	ChMod_sign.	
Comments:	enhanced full/half rate speech - version 3.	
Element Name	**Element Value**	**Comments**
mode	C_ChMod_r3	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_sign	
Structured Type:	CHMOD	
Derivation Path:		
Comments:	signalling only	
Element Name	**Element Value**	**Comments**
iei	-	
mode	C_ChMod_s	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_sign_iei	
Structured Type:	CHMOD	
Derivation Path:		
Comments:	signalling only	
Element Name	**Element Value**	**Comments**
iei	'01100011'B	
mode	C_ChMod_s	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_speech	
Structured Type:	CHMOD	
Derivation Path:	ChMod_sign.	
Comments:	speech full or half rate.	
Element Name	**Element Value**	**Comments**
mode	C_ChMod_r	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_speech_iei	
Structured Type:	CHMOD	
Derivation Path:		
Comments:	speech full or half rate.	
Element Name	**Element Value**	**Comments**
iei	'01100011'B	
mode	C_ChMod_r	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_12k	
Structured Type:	CHMOD	
Derivation Path:	ChMod_sign.	
Comments:	data 9.6 kb/s full rate, (12.0 kb/s air interface)	
Element Name	**Element Value**	**Comments**
mode	C_ChMod_12k	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_6k	
Structured Type:	CHMOD	
Derivation Path:	ChMod_sign.	
Comments:	data 4.8 kb/s full rate, (6.0 kb/s air interface)	
Element Name	**Element Value**	**Comments**
mode	C_ChMod_6k	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_3k	
Structured Type:	CHMOD	
Derivation Path:	ChMod_sign.	
Comments:	data 2.4 kb/s full rate, (3.6 kb/s air interface)	
Element Name	**Element Value**	**Comments**
mode	C_ChMod_3k	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Chneed_01	
Structured Type:	CHNEED	
Derivation Path:		
Comments:	any channel	
Element Name	**Element Value**	**Comments**
ch2	'00'B	
ch1	'00'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CiphRes_01	
Structured Type:	CPH_RES	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
sprb	'000'B	
cr	'0'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ClassMark2	
Structured Type:	MSCLM2	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iel	'03'O	
spr1	'0'B	
rl	TSPX_RevLevel	
spr2	'0'B	
a5_1	TSPX_CiphAlgA5_1	
rfpc	TSPX_RfPwrCap	
spr3	'0'B	
psc	TSPX_pSyncCap	
sssi	TSPX_SSscrnInd	
smc	TSPX_SMCap	
spr4	'00'B	
fc	TSPX_frqCap	
cm3	TSPX_ClsMk3	
spr5	'00000'B	
a5_3	TSPX_CiphAlgA5_3	
a5_2	TSPX_CiphAlgA5_2	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CntrlChDscrp(att: INTEGER; babr, cch_con, bpm:B_3; timer:OCTETSTRING)	
Structured Type:	CCD	
Derivation Path:		
Comments:	Generic Control channel descriptor.	
Element Name	**Element Value**	**Comments**
spr1	'0'B	
att	INT_TO_BIT(att,1)	
babr	babr	
ccch_con	cch_con	
spr2	'00000'B	
bpm	bpm	
t3212	timer	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphKeySN_01(cksn:B_3)	
Structured Type:	CPHKSN	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
sprb	'0'B	
ks	cksn	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphKeySN_07(par: BITSTRING)	
Structured Type:	CPHKSN	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
sprb	'0'B	
ks	par	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphMod_omit	
Structured Type:	CPHMS	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
algid	-	
sc	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphMod_01	
Structured Type:	CPHMS	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
algid	TSPX_CphAlgDef	
sc	'1'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphMod_02	
Structured Type:	CPHMS	
Derivation Path:	CphMod_01.	
Comments:	No ciphering	
Element Name	**Element Value**	**Comments**
sc	'0'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqBCCH(arfcn:INTEGER)	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Broadcast channel.	
Element Name	**Element Value**	**Comments**
hch	'0'B	
maio	-	
hsn	-	
spr	'00'B	
arfcn	INT_TO_BIT(arfcn, 10)	
maclength	'00'O	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freqchseq_omit	
Structured Type:	FRQCHS	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
larfcn	-	
incs	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freqchseq_03	
Structured Type:	FRQCHS	
Derivation Path:		
Comments:	Frequency channel sequence with the List_ARFCN={14,114}	
Element Name	**Element Value**	**Comments**
iei	'01101001'B	
larfcn	'0E'O	
incs	'000000A000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freqchseq_06	
Structured Type:	FRQCHS	
Derivation Path:		
Comments:	Frequency channel sequence with the List_ARFCN={14,18,22,24,30,31,38,53,66,73,74,75,76,108,114} see Frql_20_B5	
Element Name	**Element Value**	**Comments**
iei	'01101001'B	
larfcn	'0E'O	
incs	'442617FD71110026'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Freqchseq_22	
Structured Type:	FRQCHS	
Derivation Path:		
Comments:	Frequency channel sequence with the List_ARFCN={10,17,20,26,59,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	'01101001'B	
larfcn	'0A'O	
incs	'7360037711100260'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCH(arfcn:INTEGER)	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Default Traffic channel non frequency hopping	
Element Name	**Element Value**	**Comments**
hch	'0'B	
maio	-	
hsn	-	
spr	'00'B	
arfcn	INT_TO_BIT(arfcn, 10)	
maclength	'00'O	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCH_omit	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Omitted FRQPARA when not needed in parameters list.	
Element Name	**Element Value**	**Comments**
hch	-	
maio	-	
hsn	-	
spr	-	
arfcn	-	
maclength	-	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof1	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping traffic channel of cell A for GSM.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 12), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00001111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof1d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping channel of cell A for DCS1800	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 9), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_Ad	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof_ind3	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping traffic channel of cell A for GSM which indicates three frequences (73, 74, 75) used in TC_26_12_2_2, M=7.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 3), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000000'B	
mac_7n	'00111000'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof_ind3d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping channel of cell A for DCS1800 which indicates only 3 frequencies (734, 762, 791) used in TC_26_12_2_2, M=7	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 3), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000000'B	
mac_7n	'00010000'B	
mac_6n	'00100001'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof_ind5	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping traffic channel of cell A for GSM which indicates five frequences (17, 34, 46, 52, 73) used in TC_26_12_2_2, M=6.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 5), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000000'B	
mac_7n	'00001001'B	
mac_6n	'10010010'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa_hof_ind5d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping channel of cell A for DCS1800 which indicates only 5 frequencies (734, 741, 762, 791, 832) used in TC_26_12_2_2, M=6	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 5), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000000'B	
mac_7n	'10010000'B	
mac_6n	'00100011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof6	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 2), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'10000000'B	
mac_7n	'00000001'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof6d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 3), 6)	
hsn	'000000'B	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'00000111'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B10d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof1	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Broadcast channel of cell B.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 12), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00001111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B3	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof1d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Broadcast channel of cell B for hopping DCS1800	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 12), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00001111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B3d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof3	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 15), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'01111111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B5	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof3d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 11), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'02'O	
mac_8n	'00000111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_B5d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof_ind8	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping traffic channel (TC_26_12_2_2, M = 6 and 10)	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 8), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'11111111'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_ind8B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHb_hof_ind8d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Broadcast channel of cell B for hopping DCS1800 (TC_26_12_2_2, M=6 and 10)	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 8), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'11111111'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_ind8Bd	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCHa_hof1	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	hopping with {73,74,75} of cell allocation A.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 3), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000000'B	
mac_7n	'00111000'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqSDCCHa_hof1d	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	a hopping channel of cell A with {773,775,779} of cell allocation A.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	INT_TO_BIT((TSPX_MAIO MOD 3), 6)	
hsn	INT_TO_BIT(TSPX_HSN, 6)	
spr	-	
arfcn	-	
maclength	'03'O	
mac_8n	'00000000'B	
mac_7n	'00000111'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_20_A0d	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_omit	
Structured Type:	FRQL	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
fl	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_A	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time used in HO_case in cell A. List_ARFCN={10,17,20,26,59,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020400000002090200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_A0	
Structured Type:	FRQL	
Derivation Path:		
Comments:	Complete Cell Allocation of cell A in HO cases. ARFCN_list={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020408320202090200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_A0d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	Complete Cell Allocation of Cell A in HO cases using 512 format. frequency list, after time for a target cell used in HO_case in cell A for DCS1800. List_ARFCN={734,741,747,754,759,762,766,767,773,775,779,782,791,798,829,832,844}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'896F0A7CC5FC700A8B9F7FF452463340'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_Ad	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list used in HO_case in cell A for DCS1800 using 256 format for the List_ARFCN={747, 775, 779, 782, 791, 798, 829, 832, 844}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'09'O	
fl	'8B7599F045EFA499C0'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B0	
Structured Type:	FRQL	
Derivation Path:		
Comments:	Complete Cell Allocation of cell B in HO cases. all of the CA: ARFCN_list={14,18,22,24,30,31,38,40,60,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F02080000A060A22000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B1	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. List_ARFCN={14,18,22,24,30,31,38,53,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020010002060222000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B3	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. List_ARFCN={14,18,22,24,60,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020800000000A22000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B3d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B using 1024 format. List_ARFCN={749, 758, 761, 764, 771, 779, 782, 791, 798, 829, 832, 844}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'0D'O	
fl	'830EF70BFEB843C3FBF9F1D018'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B5	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. List_ARFCN={14,18,22,24,30,31,38,53,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'0002080000000F020010002060A22000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B5d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B. List_ARFCN={756, 758, 761, 771, 779, 782, 791, 798, 829, 832, 844}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'09'O	
fl	'8D7A23B07B3D7FD4D0'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B8d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B using 1024 format. List_ARFCN= complete cell allocation of cell B in HO cases. 739,743,746,749,756,758,761,764,771,779,782,791,798,829,832,844	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'8303F886FDBC0148BEFDFBF41890633C'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_20_B10d	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list, after time for cell B using range 256 format, ARFCN={764,779,782}	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'04'O	
fl	'8B7E097D'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_ind8B	
Structured Type:	FRQL	
Derivation Path:		
Comments:	Uses bit map 0 to allocate the following 8 frequencies:(14, 18, 24, 30, 38, 60, 73, 74)	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'00000000000003000800002020422000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_ind8Bd	
Structured Type:	FRQL	
Derivation Path:		
Comments:	Uses range 256 to allocate the following 8 frequencies:(739, 743, 746, 749, 761, 771, 782, 791)	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'08'O	
fl	'8B718B712BEEA192'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Hlcmp_AnyOrOmit	
Structured Type:	HLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE with parameter "any"' or "omit".	
Element Name	**Element Value**	**Comments**
iei	'01111101'B IF_PRESENT	
iel	*	
extb3	*	
cs	*	
in	*	
pmpp	*	
extb4	*	
hlci	*	
extb4a	*	
ehlci	*	
Detailed Comments:	used in test cases where the HLC is not checked	

Structured Type Constraint		
Constraint Name:	Hlcmp_NotApplicable	
Structured Type:	HLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE wich is not applicable for a specific service (empty IE).	
Element Name	**Element Value**	**Comments**
iei	'01111101'B	
iel	'00'O	
extb3	-	
cs	-	
in	-	
pmpp	-	
extb4	-	
hlci	-	
extb4a	-	
ehlci	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct(iel:LENGTH)	
Structured Type:	IARESTOCT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iaroct1	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O,(11 - OC_OctToInt(iel)))	
iaroct2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct_01	
Structured Type:	IARESTOCT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iaroct1	'2B2B2B2B2B2B2B2B2B2B2B'O	
iaroct2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Llcmp_AnyOrOmit	
Structured Type:	LLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE which has any or omit value in all fields.	
Element Name	**Element Value**	**Comments**
iei	'01111100'B IF_PRESENT	
iel	*	
extb3	*	
cs	*	
itc	*	
extb3a	*	
negind	*	
spb3a	*	
extb4	*	
tm	*	
itr	*	
extb4_1	*	
rateml	*	
extb5	*	
l1id	*	
uil1	*	
extb5a	*	
sb	*	
neg	*	
ur	*	
extb5b1	*	
ir	*	
nictx	*	
nicrx	*	
fctx	*	
fcrx	*	
spb5b1	*	
extb5b2	*	
hdrb	*	
mfs	*	
mode	*	
llineg	*	
ass	*	
ibob	*	
spb5b2	*	
extb5c	*	
nsb	*	
ndb	*	
pi	*	
extb5d	*	
dplxm	*	
modemt	*	
extb6	*	
l2id	*	
uil2	*	
extb6a1	*	
modl2	*	
spb6a1	*	
q933	*	
extb6a2	*	
usl2pi	*	
extb6b	*	
ws	*	
extb7	*	
l3id	*	
uil3	*	
extb7a1	*	
modl3	*	

spb7a1	*	
extb7a2	*	
usl3pi	*	
extb7b	*	
spb7b	*	
dps	*	
extb7c	*	
pws	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Llcmp_NotApplicable	
Structured Type:	LLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE which is not applicable for a specific service (empty IE).	
Element Name	**Element Value**	**Comments**
iei	'01111100'B	
iel	'00'O	
extb3	-	
cs	-	
itc	-	
extb3a	-	
negind	-	
spb3a	-	
extb4	-	
tm	-	
itr	-	
extb4_1	-	
rateml	-	
extb5	-	
l1id	-	
uil1	-	
extb5a	-	
sb	-	
neg	-	
ur	-	
extb5b1	-	
ir	-	
nictx	-	
nicrx	-	
fctx	-	
fcrx	-	
spb5b1	-	
extb5b2	-	
hdrb	-	
mfs	-	
mode	-	
llineg	-	
ass	-	
ibob	-	
spb5b2	-	
extb5c	-	
nsb	-	
ndb	-	
pi	-	
extb5d	-	
dplxm	-	
modemt	-	
extb6	-	
l2id	-	
uil2	-	
extb6a1	-	
modl2	-	
spb6a1	-	
q933	-	
extb6a2	-	
usl2pi	-	
extb6b	-	
ws	-	
extb7	-	
l3id	-	
uil3	-	
extb7a1	-	
modl3	-	

spb7a1	-	
extb7a2	-	
usl3pi	-	
extb7b	-	
spb7b	-	
dps	-	
extb7c	-	
pws	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Llcmp_BS2xOrBS4x_UDI	
Structured Type:	LLCMP	
Derivation Path:		
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.2.1 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iei	'01111100'B	
iel	('06'O, '07'O, '08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'01000'B	
extb3a	'1'B IF_PRESENT	
negind	*	
spb3a	'000000'B IF_PRESENT	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb4_1	-	
rateml	-	
extb5	'0'B	
l1id	'01'B	
uil1	'00001'B	
extb5a	'0'B	
sb	'1'B	
neg	'0'B	
ur	('00010'B, '00011'B, '00101'B, '01000'B, '10111'B, '11110'B)	
extb5b1	'0'B	
ir	('01'B, '10'B)	
nictx	'?'B	
nicrx	'?'B	
fctx	'0'B	
fcrx	'0'B	
spb5b1	'0'B	
extb5b2	-	
hdrb	-	
mfs	-	
mode	-	
llineg	-	
ass	-	
ibob	-	
spb5b2	-	
extb5c	'?'B	
nsb	('01'B, '11'B)	
ndb	('10'B, '11'B)	
pi	('000'B,'010'B, '011'B, '100'B, '101'B)	
extb5d	'1'B IF_PRESENT	
dplxm	*	
modemt	*	
extb6	*	
l2id	*	
uil2	*	
extb6a1	*	
modl2	*	
spb6a1	*	
q933	*	
extb6a2	*	
usl2pi	*	
extb6b	*	
ws	*	
extb7	*	
l3id	*	
uil3	*	

extb7a1	*	
modl3	*	
spb7a1	*	
extb7a2	*	
usl3pi	*	
extb7b	*	
spb7b	*	
dps	*	
extb7c	*	
pws	*	
Detailed Comments:	Corresponding to the Bearer Services 21 ... 26, 41 ... 46 Unrestrict digital information transfer capability.	

Structured Type Constraint		
Constraint Name:	Llcmp_BS2xOrBS61aOrBS81a_3100	
Structured Type:	LLCMP	
Derivation Path:	Llcmp_BS2xOrBS4x_UDI.	
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.2.2 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iel	('07'O,'08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'10000'B	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb5	'0'B	
l1id	'01'B	
uil1	'00011'B	
extb5a	'0'B	
sb	?	
neg	?	
ur	?	
extb5b1	'0'B	
ir	?	
nictx	?	
nicrx	?	
fctx	?	
fcrx	?	
spb5b1	'0'B	
extb5c	'0'B	
nsb	?	
ndb	?	
pi	?	
extb5d	'1'B	
dplxm	?	
modemt	('010001'B, '010010'B, '010011'B, '010100'B, '010111'B, '011100'B)	
Detailed Comments:	Corresponding to the Bearer Services 21 ... 26, 61 Asynchronous or 81 Asynchronous 3.1 kHz audio ex-PLMN information transfer capability.	

Structured Type Constraint		
Constraint Name:	Llcmp_BS3x_UDI	
Structured Type:	LLCMP	
Derivation Path:	Llcmp_BS2xOrBS4x_UDI.	
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.3.1 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iel	('05'O, '06'O, '07'O, '08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'01000'B	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb5	'0'B	
l1id	'01'B	
uil1	('00001'B, '01001'B)	
extb5a	'0'B	
sb	'0'B	
neg	'0'B	
ur	('00010'B, '00011'B, '00101'B, '01000'B)	
extb5b1	'?'B	
ir	('01'B, '10'B)	
nictx	'?'B	
nicrx	'?'B	
fctx	'?'B	
fcrx	'?'B	
spb5b1	'0'B	
extb5c	'0'B IF_PRESENT	
nsb	*	
ndb	*	
pi	*	
extb5d	'1'B IF_PRESENT	
dplxm	*	
modemt	*	
extb6	*	
l2id	'10'B IF_PRESENT	
uil2	('00110'B, '00111'B) IF_PRESENT	
extb7	*	
l3id	'11'B IF_PRESENT	
uil3	('00110'B, '00111'B) IF_PRESENT	
Detailed Comments:	Corresponding to the Bearer Services 31 ... 34 unrestricted digital information transfer capability.	

Structured Type Constraint		
Constraint Name:	Llcmp_BS3xOrBS61sOrBS81s_3100	
Structured Type:	LLCMP	
Derivation Path:	Llcmp_BS2xOrBS4x_UDI.	
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.3.2 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iel	('03'O,'04'O, '08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'10000'B	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb5	'?'B	
l1id	'01'B	
uil1	'00011'B	
extb5a	'0'B IF_PRESENT	
sb	*	
neg	*	
ur	*	
extb5b1	*	
ir	*	
nictx	*	
nicrx	*	
fctx	*	
fcrx	*	
spb5b1	*	
extb5c	'0'B IF_PRESENT	
nsb	*	
ndb	*	
pi	*	
extb5d	'1'B IF_PRESENT	
dplxm	*	
modemt	('010010'B, '010011'B, '010111'B, '011100'B) IF_PRESENT	
extb6	*	
l2id	'10'B IF_PRESENT	
uil2	('00110'B, '00111'B) IF_PRESENT	
extb7	*	
l3id	'11'B IF_PRESENT	
uil3	('00110'B, '00111'B) IF_PRESENT	
Detailed Comments:	Corresponding to the Bearer Services 31 ... 34, 61 Synchronous or 81 Synchronous 3.1kHz audio information transfer capability.	

Structured Type Constraint		
Constraint Name:	Llcmp_BS5x	
Structured Type:	LLCMP	
Derivation Path:	Llcmp_BS2xOrBS4x_UDI.	
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.5.1 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iel	('06'O,'07'O, '08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'01000'B	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb5	'0'B	
l1id	'01'B	
uil1	'01001'B	
extb5a	'?'B	
sb	'0'B	
neg	'0'B	
ur	('00011'B, '00101'B, '01000'B)	
extb5b1	*	
ir	*	
nictx	*	
nicrx	*	
fctx	*	
fcrx	*	
spb5b1	*	
extb5c	*	
nsb	*	
ndb	*	
pi	*	
extb5d	*	
dplxm	*	
modemt	*	
extb6	?	
l2id	'10'B	
uil2	('00110'B, '00111'B)	
extb7	?	
l3id	'11'B	
uil3	('00110'B, '00111'B)	
Detailed Comments:	Corresponding to the Bearer Services 51 ... 53 unrestricted digital information transfer capability.	

Structured Type Constraint		
Constraint Name:	LocAreaId(mcc, mnc, lac:OCTETSTRING)	
Structured Type:	LAI	
Derivation Path:		
Comments:	Cell A default value for L3 test	
Element Name	**Element Value**	**Comments**
iei	-	
mcc	mcc	
mnc	mnc	
lac	lac	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	LocUpType(locup:B_2)	
Structured Type:	LUT	
Derivation Path:		
Comments:	location updating type	
Element Name	**Element Value**	**Comments**
foreq	'?'B	
sprb	'0'B	
lut	locup	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiImsi_01iei	
Structured Type:	MI	
Derivation Path:		
Comments:	the TSPX_IMSI is the IMSI of the MS under test	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	OC_IntToOct(((LENGTH_OF(TSPX_IMSI))/2 + 1),1)	
idigit_1	OC_FirstDigi(TSPX_IMSI)	
oei	OC_OeBit(TSPX_IMSI)	
toi	'001'B	
idigits_other	OC_OtherDigi(TSPX_IMSI)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiImsi_01	
Structured Type:	MI	
Derivation Path:		
Comments:	the TSPX_IMSI is the IMSI of the MS under test	
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct(((LENGTH_OF(TSPX_IMSI))/2 + 1),1)	
idigit_1	OC_FirstDigi(TSPX_IMSI)	
oei	OC_OeBit(TSPX_IMSI)	
toi	'001'B	
idigits_other	OC_OtherDigi(TSPX_IMSI)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiImsi_02	
Structured Type:	MI	
Derivation Path:		
Comments:	This constraint is used in paging filling message. No mobile is paged (Type of identity is set to '000'B, i.e. no identity).	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'01'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'000'B	
idigits_other	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_01	
Structured Type:	MI	
Derivation Path:		
Comments:	default TMSI for testing.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'100'B	
idigits_other	TSPX_TMSI	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_01iei	
Structured Type:	MI	
Derivation Path:		
Comments:	default TMSI for testing.	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'100'B	
idigits_other	TSPX_TMSI	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_02	
Structured Type:	MI	
Derivation Path:		
Comments:	A new TMSI	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'100'B	
idigits_other	TSPX_TMSI1	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_r02iei	
Structured Type:	MI	
Derivation Path:		
Comments:	default TMSI for testing with toi setting to no id.	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'000'B	
idigits_other	TSPX_TMSI	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_omit	
Structured Type:	MA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_01	
Structured Type:	MA	
Derivation Path:		
Comments:	non hopping	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'00'O	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_01iei	
Structured Type:	MA	
Derivation Path:		
Comments:	non hopping	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'00'O	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_ind5A	
Structured Type:	MA	
Derivation Path:		
Comments:	Indicates 5 frequences (17, 34, 46, 52, 73)	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000000'B	
mac_7n	'00001001'B	
mac_6n	'10010010'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_ind5Ad	
Structured Type:	MA	
Derivation Path:		
Comments:	Indicates 5 frequences (734, 741, 762, 791, 832)	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000000'B	
mac_7n	'10010000'B	
mac_6n	'00100011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_281	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with indicates to the ARFCN_list={73,74,75} of cell allocation A.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000000'B	
mac_7n	'00111000'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_281d	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with indicates to the ARFCN_list={773,775,779} of cell allocation A.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000000'B	
mac_7n	'00000111'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_ad281d	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with indicates to the ARFCN_list={734,762,791} of cell allocation A.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000000'B	
mac_7n	'00010000'B	
mac_6n	'00100001'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_02	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE matching any value.	
Element Name	**Element Value**	**Comments**
ba_used	?	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	?	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	?	
rxlev_nc1	*	
bcchfrq_nc1	*	
bsic_nc1	*	
rxlev_nc2	*	
bcchfrq_nc2	*	
bsic_nc2	*	
rxlev_nc3	*	
bcchfrq_nc3	*	
bsic_nc3	*	
rxlev_nc4	*	
bcchfrq_nc4	*	
bsic_nc4	*	
rxlev_nc5	*	
bcchfrq_nc5	*	
bsic_nc5	*	
rxlev_nc6	*	
bcchfrq_nc6	*	
bsic_nc6	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	NT_BothT_BothNTorT_BothT_BothNT	
Structured Type:	CE	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
TorBothTorBothNT	('00'B, '10'B, '11'B)	
NTorBothTorBothNT	('01'B, '10'B, '11'B)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Pcmd_19(powerlevel:BITSTRING)	
Structured Type:	PCMD	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
sprb	'000'B	
pl	powerlevel	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Pcmd_20(powerlevel:B_5)	
Structured Type:	PCMD	
Derivation Path:		
Comments:	PowerCmd used in synchronized and non synchronized HO cases.	
Element Name	**Element Value**	**Comments**
sprb	'011'B	
pl	powerlevel	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Pm(pgm:B_2)	
Structured Type:	PM	
Derivation Path:		
Comments:	normal paging mode	
Element Name	**Element Value**	**Comments**
sprb	'00'B	
pgm	pgm	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ProgInd_omit	
Structured Type:	PI	
Derivation Path:		
Comments:	Omitted Progress Indicator.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
extb3	-	
cs	-	
spb	-	
loc	-	
extb4	-	
prd	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RachCntrlPara(maxtx:B_2; txint:B_4; re:B_1)	
Structured Type:	RACHCP	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
maxrtx	maxtx	
txint	txint	
cba	'0'B	
re	re	
acc_2	'00000'B	
ec	'0'B	
acc_1	'0000000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RachCntrlPara_r01	
Structured Type:	RACHCP	
Derivation Path:		
Comments:	Call reestablishment is not allowed in the cell.	
Element Name	**Element Value**	**Comments**
maxrtx	'00'B	
txint	'0010'B	
cba	'0'B	
re	'1'B	
acc_2	'00000'B	
ec	'0'B	
acc_1	'0000000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RelTmdDif_omit	
Structured Type:	TDIF	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
value	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Rqr2(Rr: BITSTRING; Fn: FN)	
Structured Type:	RQR	
Derivation Path:		
Comments:	To address the MS under test	
Element Name	**Element Value**	**Comments**
ra	Rr	
fn	Fn	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SI3RestOctetDef	
Structured Type:	SI3RO	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
pi	'0'B	
cbq	'0'B	
cro	'101011'B	
to	'001'B	
pt	'01011'B	
poi	'0'B	
po	'01'B	
ti	'0'B	
ecsc	'1'B	
spr1	'011'B	
spr2	'00101011'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SI4RestOctetDef	
Structured Type:	SI4RO	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
pi	'0'B	
cbq	'0'B	
cro	'101011'B	
to	'001'B	
pt	'01011'B	
poi	'0'B	
po	'01'B	
spr1	'01011'B	
spr2	'00101011'B	
spr3	'00101011'B	
spr4	'00101011'B	
spr5	'00101011'B	
spr6	'00101011'B	
spr7	'00101011'B	
spr8	'00101011'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SI4RestOctet_CBMS	
Structured Type:	SI4RO	
Derivation Path:		
Comments:	Rest octets for SMSCB SysInf4 message.	
Element Name	**Element Value**	**Comments**
pi	'0'B	
cbq	'0'B	
cro	'101011'B	
to	'001'B	
pt	'01011'B	
poi	'0'B	
po	'01'B	
spr1	'01011'B	
spr2	'00101011'B	
spr3	-	
spr4	-	
spr5	-	
spr6	-	
spr7	-	
spr8	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Signal_01	
Structured Type:	SIGNAL	
Derivation Path:		
Comments:	signal value is arbitrarily selected.	
Element Name	**Element Value**	**Comments**
iei	'00110100'B	
sigv	'00000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	StartingTm_omit	
Structured Type:	STRT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
fn	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Synchi(rot:B_1; si:B_2)	
Structured Type:	SYNCHI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'1101'B	
nci	'0'B	
rot	rot	
si	si	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Synchi_omit	
Structured Type:	SYNCHI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
nci	-	
rot	-	
si	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TimingAdv(bits:INTEGER)	
Structured Type:	TA	
Derivation Path:		
Comments:	Time Advance value is pameterized.	
Element Name	**Element Value**	**Comments**
iei	-	
sprb	'00'B	
value	INT_TO_BIT(bits, 6)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TimingAdv_omit	
Structured Type:	TA	
Derivation Path:		
Comments:	Time Advance value is pameterized.	
Element Name	**Element Value**	**Comments**
iei	-	
sprb	-	
value	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_01	
Structured Type:	TI	
Derivation Path:		
Comments:	used by the MS in the transaction initiated by the test system.	
Element Name	**Element Value**	**Comments**
ti_f	'1'B	
ti_v	'000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_02	
Structured Type:	TI	
Derivation Path:		
Comments:	used in the messages sent to the MS in the transaction initiated by test system.	
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	'000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_09	
Structured Type:	TI	
Derivation Path:		
Comments:	used by the MS in the transaction initiated by the MS.	
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	UuInfo_omit	
Structured Type:	UU	
Derivation Path:		
Comments:	Omitted UUInfo.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
uupd	-	
uui	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	V22bisOrV26ter	
Structured Type:	MODEMTYPE	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
V22bisOrV26ter	('00011'B, '00101'B)	
Detailed Comments:		

ASN1 type constraints

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CF all synchronous services
Constraint Value	

```
activateSSComponents      activateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
12,
                                                                      ss_ForBS

{
ss_Code                           '20'H,
basicService       bearerService      '68'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFU all basic services
Constraint Value	

```
activateSSComponents      activateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
12,
                                                                      ss_ForBS

{
ss_Code                           '21'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	BAOC all synchronous services

Constraint Value

```
activateSSComponents      activateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
12,
                                                                      ss_ForBS

{
ss_Code                          '92'H,
basicService          bearerService             '68'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	BIC-Roam

Constraint Value

```
activateSSComponents      activateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
12,
                                                                      ss_ForBS

{
ss_Code                          '9B'H  -- BIC-Roam
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_05
ASN.1 Type:	Component
Derivation Path:	
Comments:	BOIC

Constraint Value

```
activateSSComponents      activateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
12,
                                                                      ss_ForBS

{
ss_Code                          '93'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_06
ASN.1 Type:	Component
Derivation Path:	
Comments:	BAIC
Constraint Value	

activateSSComponents activateSS_InvokeComp
{
invokeID
?,
localValue
12,
ss_ForBS
{
ss_Code '9A'H
}
}	

Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	BldMptySS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	build multiparty request
Constraint Value	

buildMPTYComponents buildMPTY_InvokeComp
{
invokeID
?,
localValue
124
}	

Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	Comps_01(comp:Component)
ASN.1 Type:	Components
Derivation Path:	
Comments:	A received FIE may contain several components, but at least contains "comp".
Constraint Value	
SUPERSET({ comp})	
Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFC for speech

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '28'H,
basicService    teleservice    '10'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRc for all facsimile

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '2B'H,
basicService    teleservice    '60'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	deactivation for barring

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
13,
                                                                  ss_ForBS

{

ss_Code       '90'H,

basicService   teleservice   '10'H

}
                                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	deactivation for barring of outging calls

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
13,
                                                                  ss_ForBS

{

ss_Code       '91'H,

basicService   teleservice   '60'H

}
                                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_05
ASN.1 Type:	Component
Derivation Path:	
Comments:	deactivation for barring of incoming calls

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
13,
                                                                  ss_ForBS

{

ss_Code       '99'H

}
                                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_06
ASN.1 Type:	Component
Derivation Path:	
Comments:	deactivation for BOICExHC

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '94'H
}
                                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	EraseSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFC for all facsimile

Constraint Value

```
eraseSSComponents      eraseSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
11,
                                                                      ss_ForBS

{
ss_Code       '28'H,
basicService   teleservice   '60'H
}
                                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	EraseSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRc for all basic services

Constraint Value

```
eraseSSComponents      eraseSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
11,
                                                                      ss_ForBS

{
ss_Code       '2B'H
}
                                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	EraseSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFU for speech

Constraint Value

```
eraseSSComponents     eraseSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
11,
                                                                      ss_ForBS

{
ss_Code      '21'H,
basicService   teleservice   '10'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	EraseSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRy for all facsimile

Constraint Value

```
eraseSSComponents     eraseSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
11,
                                                                      ss_ForBS

{
ss_Code      '2A'H,
basicService   teleservice   '60'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	FwdChAdvRslt_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
forwardChargeAdviceComponents     forwardChargeAdvice_ReturnResultComp
            {
                                                                      invokeID
0,
                                                                      result
*
               }
```

Detailed Comments:

ASN.1 Type Constraint Declaration

Constraint Name: GetPasswdRslt_01
ASN.1 Type: Component
Derivation Path:
Comments:

Constraint Value

```
getPasswordComponents     getPassword_ReturnResultComp
                                                                    {
                                                                      invokeID
0,
                                                                      result
                                                                              {

localValue          18,

currentPassword   "1234"
                                                                               }
                                                                       }
```

Detailed Comments:

ASN.1 Type Constraint Declaration

Constraint Name: GetPasswdRslt_02
ASN.1 Type: Component
Derivation Path:
Comments:

Constraint Value

```
getPasswordComponents     getPassword_ReturnResultComp
                                                                    {
                                                                      invokeID
0,
                                                                      result
                                                                              {

localValue          18,

currentPassword   "9876"
                                                                               }
                                                                       }
```

Detailed Comments:

ASN.1 Type Constraint Declaration

Constraint Name: GetPasswdRslt_03
ASN.1 Type: Component
Derivation Path:
Comments:

Constraint Value

```
getPasswordComponents     getPassword_ReturnResultComp
                                                                    {
                                                                      invokeID
0,
                                                                      result
                                                                              {

localValue          18,

currentPassword   "9877"
                                                                               }
                                                                       }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFB for all basic services

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp       {
    invokeID                      ?,
    localValue                    14,
    ss_ForBS     {

ss_Code      '29'H
                }
                                                            }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRy for Speech

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
14,
                                                                  ss_ForBS

{
ss_Code      '2A'H, --CNFRy
basicService   teleservice   '10'H
}
                                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRc for all basic services

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
14,
                                                                  ss_ForBS

{
ss_Code      '2B'H
}
                                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFB for all facsimile

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
14,
                                                                  ss_ForBS

{
ss_Code      '29'H,
basicService   teleservice   '60'H
}
                                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_05
ASN.1 Type:	Component
Derivation Path:	
Comments:	BICRoam

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
14,
                                                                  ss_ForBS

{
ss_Code      '9B'H
}
                                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_06
ASN.1 Type:	Component
Derivation Path:	
Comments:	BOIC

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
14,
                                                                  ss_ForBS

{
ss_Code      '93'H
}
                                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_07
ASN.1 Type:	Component
Derivation Path:	
Comments:	BAIC

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
14,
                                                                  ss_ForBS
{
ss_Code      '9A'H
}
                                                                }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_08
ASN.1 Type:	Component
Derivation Path:	
Comments:	BOICExHC

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
14,
                                                                  ss_ForBS
{
ss_Code      '94'H
}
                                                                }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	NotificationSS_07(Invkid: OCTETSTRING)
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
unstructuredSSNotifyComponents     unstructuredSSNotify_ReturnResultComp
                                                                {
                                                                  invokeID
OC_OctToInvokeIDType(Invkid),
                                                                  result
                                                                        {
localValue                    61
                                                                        }
                                                                }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	NotificationSS_09(Invkid: OCTETSTRING)
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
unstructuredSSNotifyComponents     unstructuredSSNotify_ReturnErrorComp

                                                          errorCodes
                                                                    {
                                                                       invokeID
OC_OctToInvokeIDType(Invkid),
                                                                       errorCode
ussd_Busy
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegisterSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRy Speech

Constraint Value

```
registerSSComponents     registerSS_InvokeComp   {
     invokeID                    ?,
     localValue                 10,
registerSS_Arg  {

ss_Code                        '2A'H,

basicService    teleservice    '10'H,

forwardedToNumber         '8100342143'H,
                                       -- Unknown Number/ISDN telephony + '00431234'

noReplyConditionTime       5

}
                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegisterSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFU

Constraint Value

```
registerSSComponents     registerSS_InvokeComp      {
   invokeID                    ?,
   localValue                 10,
   registerSS_Arg  {

ss_Code                        '21'H,

basicService    teleservice    '60'H,

forwardedToNumber         '8100342143'H
                                       -- Unknown Number/ISDN telephony + '00431234'

                }
                                                   }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegisterSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFB

Constraint Value

```
registerSSComponents     registerSS_InvokeComp     {
       invokeID                    ?,
       localValue                 10,
       registerSS_Arg     {

ss_Code                      '29'H,

basicService    bearerService      '60'H,

forwardedToNumber          '8100342143'H
                                    -- Unknown Number/ISDN telephony + '00431234'
                            }
                                                  }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegisterSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	CF for all facsimile

Constraint Value

```
registerSSComponents     registerSS_InvokeComp     {
     invokeID                    ?,
     localValue                 10,
     registerSS_Arg      {

ss_Code                           '20'H,

basicService     teleservice      '60'H,

forwardedToNumber           '8100342143'H
                                       -- Unknown Number/ISDN telephony + '00431234'
                           }
                                                  }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegPasswdSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	All call restriction services.

Constraint Value

```
registerPasswordComponents     registerPassword_InvokeComp
                                                                                {
                                                                                  invokeID

?,
                                                                                  localValue

17,
                                                                                  ss_Code

'90'H
                                                                                   }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ProcessUSSData_01(ussdString: IA5String)
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
processUnstructuredSSDataComponents     processUnstructuredSSData_InvokeComp
                                                                          {
                                                                            invokeID
?,
                                                                            localValue
19,
                                                                            ss_UserData
ussdString
                                                                          }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ProcessUSSDReq_01(ussdString: IA5String)
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
processUnstructuredSSRequestComponents     processUnstructuredSSRequest_InvokeComp
                                                                          {
                                                                            invokeID
?,
                                                                            localValue
59,
                                                                            ussd_Arg

{
ussd_DataCodingScheme      'F0'O,
ussd_String            OC_CodingOfUssdString(ussdString)
}
                                                                           }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	USSDReq_04(Invkid :OCTETSTRING; ussdString: IA5String)
ASN.1 Type:	Component
Derivation Path:	
Comments:	ReturnResult for Unstructured SS request

Constraint Value

```
unstructuredSSRequestComponents     unstructuredSSRequest_ReturnResultComp
                                                                          {
                                                                            invokeID
OC_OctToInvokeIDType(Invkid),
                                                                            result
                                                                                     {
localValue                     60,
ussd_Res
{
ussd_DataCodingScheme      'F0'O,
ussd_String OC_CodingOfUssdString(ussdString)
}
                                                                                     }
                                                                           }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	USSDReq_05(Invkid :OCTETSTRING)
ASN.1 Type:	Component
Derivation Path:	
Comments:	Return Error for UnstructuredSS-Request with the error code USSD Busy
Constraint Value	

unstructuredSSRequestComponents	unstructuredSSRequest_ReturnErrorComp
	errorCodes
	{
	invokeID
	OC_OctToInvokeIDType(Invkid),
	errorCode
	ussd_Busy
	}

Detailed Comments:	

ASP constraint declarations

TTCN ASP constraint declarations

ASP Constraint Declaration		
Constraint Name:	AlertRcv(pdu: ALERT_PDU)	
ASP Type:	DL_DatInAlert	
Derivation Path:		
Comments:	To receive an ALERTING message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AlertSnd(ch:LOGICCH; pdu: ALERT_PDU)	
ASP Type:	DL_DatRqAlert	
Derivation Path:		
Comments:	To send an ALERTING message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AssCmd(ch:LOGICCH; pdu: ASS_CMD_PDU)	
ASP Type:	DL_DatRqAssCmd	
Derivation Path:		
Comments:	To send an ASSIGNMENT COMMAND message which is assigned in send statement.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AssCmp(ch:LOGICCH; pdu:ASS_COM_PDU)	
ASP Type:	DL_DatInAssCom	
Derivation Path:		
Comments:	To match any received ASSIGNMENT COMPLETE message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AssFl_any_cau(ch: LOGICCH)	
ASP Type:	DL_DatInAssfl	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	AssgnFl_01	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AuthReq(ch: LOGICCH; pdu: AUTH_RQ_PDU)	
ASP Type:	DL_DatRqAuthRq	
Derivation Path:		
Comments:	To send an AUTHENTICATION REQUEST message with default values.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AuthRes(pdu: AUTH_RES_PDU)	
ASP Type:	DL_DatInAuthRes	
Derivation Path:		
Comments:	To match any received AUTHENTICATION RESPONSE message which contains any SRES.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CallCfm(pdu: CALL_CO_PDU)	
ASP Type:	DL_DatInCallCo	
Derivation Path:		
Comments:	To receive a CC CALL CONFIRMED message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CallProc(ch: LOGICCH; pdu: CALL_PROC_PDU)	
ASP Type:	DL_DatRqCallProc	
Derivation Path:		
Comments:	To send a CALL PROCEEDING message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChmmoAckRcv(ch:LOGICCH; msg:CHMMO_ACK_PDU)	
ASP Type:	DL_DatInChmmoAck	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChmmoReqSnd(ch:LOGICCH; msg:CHMMO_PDU)	
ASP Type:	DL_DatRqChmmo	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChRel(ch: LOGICCH; pdu: CH_REL_PDU)	
ASP Type:	DL_DatRqChRel	
Derivation Path:		
Comments:	To send a CHANNEL RELEASE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChReq(pdu: CH_RQ_PDU)	
ASP Type:	DL_RacInChRq	
Derivation Path:		
Comments:	To receive a primitive containing a CHANNEL REQUEST message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
fn	?	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CMSerReq(pdu: CMS_RQ_PDU)	
ASP Type:	DL_EstInCmsRq	
Derivation Path:		
Comments:	To receive a CM SERVICE REQUEST message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ConnAckRcv(pdu: CONN_ACK_PDU)	
ASP Type:	DL_DatInConnAck	
Derivation Path:		
Comments:	To receive a CONNECT ACKNOWLEDGE message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ConnAckSnd(ch:LOGICCH; pdu: CONN_ACK_PDU)	
ASP Type:	DL_DatRqConnAck	
Derivation Path:		
Comments:	To send a CONNECT ACKNOWLEDGE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ConnRcv(pdu: CONN_PDU)	
ASP Type:	DL_DatInConn	
Derivation Path:		
Comments:	To receive a CC CONNECT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ConnSnd(ch: LOGICCH; pdu: CONN_PDU)	
ASP Type:	DL_DatRqConn	
Derivation Path:		
Comments:	To send a CC CONNECT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CphCmd(ch: LOGICCH; pdu: CPHM_CMD_PDU)	
ASP Type:	DL_DatRqCphmCmd	
Derivation Path:		
Comments:	To send a CIPHERING MODE COMMAND message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CphCom(pdu: CPHM_COM_PDU)	
ASP Type:	DL_DatInCphmCom	
Derivation Path:		
Comments:	To receive a CIPHERING MODE COMPLETE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DiscRcv(ch:LOGICCH; pdu: DISC_PDU)	
ASP Type:	DL_DatInDisc	
Derivation Path:		
Comments:	To match a received DISCONNECT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DiscSnd(ch:LOGICCH; pdu: DISC_PDU)	
ASP Type:	DL_DatRqDisc	
Derivation Path:		
Comments:	To send a DISCONNECT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLEstInd(ch :LOGICCH)	
ASP Type:	DL_EstIn	
Derivation Path:		
Comments:	Indication of a layer 2 connection establishment	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
establish_mode	?	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLRelInd_01	
ASP Type:	DL_RelIn	
Derivation Path:		
Comments:	Layer 2 indication of the layer 2 connection has been released.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
release_mode	?	
outstanding_indicator	?	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLRelInd_02	
ASP Type:	DL_RelIn	
Derivation Path:		
Comments:	Layer 2 indication of the layer 2 connection has been released.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
release_mode	C_LocEndRel	
outstanding_indicator	?	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ESetupRcv(pdu: ESETUP_PDU)	
ASP Type:	DL_DatInESetup	
Derivation Path:		
Comments:	To receive an emergency call SETUP message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvCmpRcv(ch:LOGICCH; msg:HO_COM_PDU)	
ASP Type:	DL_DatInHoCom	
Derivation Path:		
Comments:	To receive any HANDOVER COMPLETE message on channel ch.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvSnd(ch:LOGICCH; msg:HO_CMD_PDU)	
ASP Type:	DL_DatRqHoCmd	
Derivation Path:		
Comments:	To send a HANDOVER COMMAND indicating a finely synchronised intra cell handover.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvAcc_01	
ASP Type:	DL_RacInHoacc	
Derivation Path:		
Comments:	To received any HANDOVER ACCESS message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	HandOverAcc_01	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvAccRcv(ch: LOGICCH; msg:HOACC_PDU)	
ASP Type:	DL_RacInHoacc	
Derivation Path:		
Comments:	To received any HANDOVER ACCESS message on channel `ch`.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvFlRcv(ch:LOGICCH; msg :HOFL_PDU)	
ASP Type:	DL_DatInHofl	
Derivation Path:		
Comments:	To match a received HANDOVER FAILURE message containing any RR cause.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ImmAss(ch: LOGICCH; pdu: IMMASS_PDU)	
ASP Type:	DL_UdatRqImmass	
Derivation Path:		
Comments:	To send an IMMEDIATE ASSIGNMENT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ImsiDet(pdu:IMSID_IN_PDU)	
ASP Type:	DL_EstInImsidIn	
Derivation Path:		
Comments:	To match a received IMSI DETACH INDICATION message matching any MS classmark1 value and any mobile identity value	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	LocAcp(ch:LOGICCH; pdu:LUP_ACP_PDU)	
ASP Type:	DL_DatRqLupAcp	
Derivation Path:		
Comments:	To send a LOCATION UPDATING ACCEPT message with a new mobile identity TMSI and location area.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	LocUp(ch: LOGICCH; pdu:LUP_RQ_PDU)	
ASP Type:	DL_EstInLupRq	
Derivation Path:		
Comments:	To match a received LOCATION UPDATING REQUEST message containing location updating type = IMSI attach.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	LocRej(ch: LOGICCH; pdu:LUP_REJ_PDU)	
ASP Type:	DL_DatRqLupRej	
Derivation Path:		
Comments:	To send a LOCATION UPDATING REJECT message containing the reject cause IMSI unknown in HLR. Used var's: ch	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	MDLRelReq(ch :LOGICCH)	
ASP Type:	MDL_RelRq	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
release_mode	C_LocEndRel	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	MsrRept(pdu:MSR_RPT_PDU)	
ASP Type:	DL_UdatInMsrRpt	
Derivation Path:		
Comments:	To receive a measurement report without measurement results.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PgReq1(ch:LOGICCH; pgg:PGG; pg1_req_pdu: PG1_RQ_PDU)	
ASP Type:	DL_UdatRqPg1Rq	
Derivation Path:		
Comments:	To send a PAGING REQUEST TYPE 1 message to the paging group indicated by the parameter pgg which is derived from system parameters. The PAGING REQUEST TYPE1 message requests normal paging mode, any channel and for the MS identity TSPX_TMSI.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
pgg	pgg	
msg	pg1_req_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PagingRes(pdu: PG_RES_PDU)	
ASP Type:	DL_EstInPgRes	
Derivation Path:		
Comments:	To receive a PAGING RESPONSE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PhyInfo_01(ch: LOGICCH; ta:TA)	
ASP Type:	DL_DatRqPhyinfo	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	Phyinfor_01(ta)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ReleaseRcv(pdu: REL_PDU)	
ASP Type:	DL_DatInRel	
Derivation Path:		
Comments:	To receive a CC RELEASE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ReleaseSnd(ch: LOGICCH; pdu: REL_PDU)	
ASP Type:	DL_DatRqRel	
Derivation Path:		
Comments:	To send a RELEASE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	RelComRcv(pdu: REL_COM_PDU)	
ASP Type:	DL_DatInRelCmp	
Derivation Path:		
Comments:	To receive a RELEASE COMPLETE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	RelComSnd(ch:LOGICCH; pdu: REL_COM_PDU)	
ASP Type:	DL_DatRqRelCmp	
Derivation Path:		
Comments:	To send a RELEASE COMPLETE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SetupRcv(pdu: SETUP_MO_PDU)	
ASP Type:	DL_DatInSetup	
Derivation Path:		
Comments:	To receive a SETUP message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SetupSnd(ch: LOGICCH; setup:SETUP_MT_PDU)	
ASP Type:	DL_DatRqSetup	
Derivation Path:		
Comments:	To send a SETUP message which is the input parameter of this ASP.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	setup	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SyncInfo(ch:LOGICCH; bcc:BCC; ncc:NCC)	
ASP Type:	DL_UdatRqSchinfo	
Derivation Path:		
Comments:	To send SYNCHRONIZATION INFORMATION message with default parameters.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SyncInfor(bcc, ncc)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo1(ch:LOGICCH; cchd:CCHD; maxtx:B_2; txint:B_4; re:B_1)	
ASP Type:	DL_UdatRqSysinfo1	
Derivation Path:		
Comments:	To send SYSTEM INFORMATION TYPE 1 in cell A for RR testing of GSM900.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf1(cchd, maxtx, txint, re)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo1_nh(ch:LOGICCH; ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
ASP Type:	DL_UdatRqSysinfo1_nh	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf3(ci, mcc, mnc, lac, ccd, co, crh, mtmc, neci, maxtx, txint, re)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo2(ch: LOGICCH; bcchfl: NCD; maxtx:B_2; txint:B_4; re:B_1; nccp:NCCP)	
ASP Type:	DL_UdatRqSysinfo2	
Derivation Path:		
Comments:	To send the default SYSTEM INFORMATION TYPE 2 message containing default neighbour cells description in cell B for GSM900.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf2(bcchfl, maxtx, txint, re, nccp)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo2bis(ch: LOGICCH; bcchflegsm: NCD)	
ASP Type:	DL_UdatRqSysinfo2bis	
Derivation Path:		
Comments:	SYSTEM INFORMATION 2bis in cell A under EGSM with the ARFCN list = {988, 990, 1003}.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf2bis(bcchflegsm)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo3(ch:LOGICCH; ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
ASP Type:	DL_UdatRqSysinfo3	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf3(ci, mcc, mnc, lac, ccd, co, crh, mtmc, neci, maxtx, txint, re)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo4(ch:LOGICCH; mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
ASP Type:	DL_UdatRqSysinfo4	
Derivation Path:		
Comments:	To send a default SYSTEM INFORMATION TYPE 4 message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf4(mcc, mnc, lac, crh, mtmc, neci, maxtx, txint, re)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo4_CBMS(ch:LOGICCH; mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; cbchd:CHD)	
ASP Type:	DL_UdatRqSysinfo4	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 4 message for SMSCB with the values or GSM 11.10, 34.3	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf4_CBMS(mcc, mnc, lac, crh, mtmc, neci, maxtx,txint, re, cbchd)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo5(sacch: LOGICCH; sysinfo5_pdu:SYSINFO5_PDU)	
ASP Type:	DL_UdatRqSysinfo5	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 5 message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	sacch	
msg	sysinfo5_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo5bis(sacch: LOGICCH; sysinfo5bis_pdu:SYSINFO5bis_PDU)	
ASP Type:	DL_UdatRqSysinfo5bis	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 5bis message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	sacch	
msg	sysinfo5bis_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo6(sacch: LOGICCH; sysinfo6_pdu:SYSINFO6_PDU)	
ASP Type:	DL_UdatRqSysinfo6	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 6 message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	sacch	
msg	sysinfo6_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	TmsiReallocSnd(par:MI; ch:LOGICCH; mcc, mnc, lac:OCTETSTRING)	
ASP Type:	DL_DatRqTmsireCmd	
Derivation Path:		
Comments:	To send a TMSI REALLOCATION COMMAND message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	TmsiReallocCmd(par, mcc, mnc, lac)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	TmsiReallocCmp(ch:LOGICCH)	
ASP Type:	DL_DatInTmsireCom	
Derivation Path:		
Comments:	To receive a TMSI REALLOCATION COMPLETE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	TmsiReallocComp	
fn	?	
Detailed Comments:		

PDU constraint declarations

TTCN PDU constraint declarations

PDU Constraint Declaration		
Constraint Name:	Alerting_01(Ti :TI)	
PDU Type:	ALERT_PDU	
Derivation Path:		
Comments:	An ALERTING message containing mandatory IE's only to be sent to the MS.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000001'B	
fie	-	
pi	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AlertingInd_01(ti:TI)	
PDU Type:	ALERT_PDU	
Derivation Path:		
Comments:	To match an received ALERTING message	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?000001'B	
fie	*	
pi	-	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_fh(slot:SN; tsc:TSC; chtype:CH_TDMA; par_pwlvl:INTEGER; maio:MAIO; hsn:HSN; frql:FRQL; Cchd:CCHD; chmod:CHMOD; ma:MA; Cphms:CPHMS)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message for hopping without starting time.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_fh(chtype, slot, tsc, maio, hsn)	
pcmd	Pcmd_19(INT_TO_BIT(par_pwlvl, 5))	
frql_at	frql	
cchd	Cchd	
ch1mod	chmod	
ch2d_at	-	
ch2mod	-	
ma_at	ma	
strt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	Cphms	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_nfh(type:CH_TDMA; slot:SN; tsc:TSC; par_pwlvl, arfcn:INTEGER; cchd:CCHD; chmod:CHMOD; strt:STRT; cphms:CPHMS)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message. The channel mode and type are specified as parameters.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_nfh(type, slot, tsc, arfcn)	
pcmd	Pcmd_19(INT_TO_BIT(par_pwlvl, 5))	
frql_at	-	
cchd	cchd	
ch1mod	chmod	
ch2d_at	-	
ch2mod	-	
ma_at	-	
strt	strt	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	cphms	
Detailed Comments:	used iin CC testing	

PDU Constraint Declaration		
Constraint Name:	AsgnCmp_02	
PDU Type:	ASS_COM_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMPLETE message containing any cause.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101001'B	
rrcau	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AssgnFl_01	
PDU Type:	ASSFL_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101111'B	
rrcau	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AuthRequest(cksn: B_3; rand:RAND)	
PDU Type:	AUTH_RQ_PDU	
Derivation Path:		
Comments:	An AUTHENTICATION REQUEST message containing default ciphering key sequence number and default challenge RAND from PIXIT.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00010010'B	
shoct	'0000'B	
cphksn	CphKeySN_07(cksn)	
rand	rand	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AuthResponse	
PDU Type:	AUTH_RES_PDU	
Derivation Path:		
Comments:	An AUTHENTICATION RESPONSE message matching any SRES.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?010100'B	
sres	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallConfirm_01(ti:TI)	
PDU Type:	CALL_CO_PDU	
Derivation Path:		
Comments:	a CALL CONFIRMED message with TI parameterized	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?001000'B	
bcri	*	
bcap1	*	
bcap2	*	
cau	-	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallConfirm_04(ti:TI; Bcap1:BCAP)	
PDU Type:	CALL_CO_PDU	
Derivation Path:		
Comments:	a CALL CONFIRMED message with TI and bcap1 parameterized	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?001000'B	
bcri	-	
bcap1	Bcap1	
bcap2	-	
cau	-	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallConfirm_20(ti:TI)	
PDU Type:	CALL_CO_PDU	
Derivation Path:		
Comments:	a CALL CONFIRMED message without bearer capability information elements	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?001000'B	
bcri	-	
bcap1	-	
bcap2	-	
cau	-	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallConfirm_Efr_20(ti:TI)	
PDU Type:	CALL_CO_PDU	
Derivation Path:		
Comments:	a CALL CONFIRMED message for EFR Mobiles	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?001000'B	
bcri	-	
bcap1	*	
bcap2	-	
cau	-	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallProced_03	
PDU Type:	CALL_PROC_PDU	
Derivation Path:		
Comments:	An CALL PROCEEDING message templet used for CallProcGen	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'00000010'B	
bcri	-	
bcap1	-	
bcap2	-	
fie	-	
pi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChmomoAck_01(chmd:CHMOD; chd: CHD)	
PDU Type:	CHMMO_ACK_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010111'B	
chd	chd	
chmod	chmd	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChmomoReq_01(chmd:CHMOD; chd:CHD)	
PDU Type:	CHMMO_PDU	
Derivation Path:		
Comments:	the channel being modified is default full rate traffic channel.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010000'B	
chd	chd	
chmod	chmd	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRelease_01	
PDU Type:	CH_REL_PDU	
Derivation Path:		
Comments:	A CHANNEL RELEASE message with RR cause = normal event	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00001101'B	
rrcau	'00000000'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_02	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match any received CHANNEL REQUEST message	
Field Name	**Field Value**	**Comments**
ecau_rrf	'????????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_04	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message which originates a call (establishment cause = '111'B), "Originating call and TCH/F is needed, or IMSI detach, SMS or SS procedures that can be completed with an SDCCH and NECI set to 0".	
Field Name	**Field Value**	**Comments**
ecau_rrf	'111?????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_15	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '111'B or '0100'B or '0101'B --- initiate outgoing call or SDCCH other procedures.	
Field Name	**Field Value**	**Comments**
ecau_rrf	('111?????'B, '0100????'B, '0101????'B, '101?????'B, '0001????'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_16	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '101'B for emergency call.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'101?????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_17	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	A CHANNEL REQUEST message containing establishment cause = 100, 0010, 0011, 0001	
Field Name	**Field Value**	**Comments**
ecau_rrf	('100?????'B, '0010????'B, '0011????'B, '0001????'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_01	
PDU Type:	CMS_RQ_PDU	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?100100'B	
cphksn	?	
svtype	?	
msclm	?	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_04	
PDU Type:	CMS_RQ_PDU	
Derivation Path:	CMServiceReq_01.	
Comments:	To match a received CM SERVICE REQUEST message containing CM service type = "Mobile originating call establishment or packet mode connection establishment" or " emergency call establishment".	
Field Name	**Field Value**	**Comments**
svtype	('0001'B, '0010'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_06(cksn:B_3)	
PDU Type:	CMS_RQ_PDU	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message for emergency call with TMSI and correct CKSN.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?100100'B	
cphksn	CphKeySN_01(cksn)	
svtype	C_CMServiceTypeE	
msclm	?	
mi	MiTmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Connect_01(ti:TI)	
PDU Type:	CONN_PDU	
Derivation Path:		
Comments:	CC CONNECT message matching any received value	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?000111'B	
fie	*	
pi	-	
cnn	-	
cns	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Connect_02(Ti:TI)	
PDU Type:	CONN_PDU	
Derivation Path:		
Comments:	CC CONNECT message containing mandatory IE's only.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000111'B	
fie	-	
pi	-	
cnn	-	
cns	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ConnectAck_01(ti:TI)	
PDU Type:	CONN_ACK_PDU	
Derivation Path:		
Comments:	A CONNECT ACKNOWLEDGE message n -> ms	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00001111'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ConnectAck_02(Ti:TI)	
PDU Type:	CONN_ACK_PDU	
Derivation Path:		
Comments:	A CONNECT ACKNOWLEDGE message	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?001111'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CphModeCmd_01	
PDU Type:	CPHM_CMD_PDU	
Derivation Path:		
Comments:	CIPHERING MODE COMMAND message, the ciphering algorithm is specified by PIXIT.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00110101'B	
cph_res	CiphRes_01	
cphms	CphMod_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CphModeCmp_01	
PDU Type:	CPHM_COM_PDU	
Derivation Path:		
Comments:	To match any received CIPHERING MODE COMPLETE message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00110010'B	
mei	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	DisconnS(Ti:TI; cau:CAU; progind:PI; uuinf:UU)	
PDU Type:	DISC_PDU	
Derivation Path:		
Comments:	A DISCONNECT message that contains cause = normal clearing and progress indicator = #8.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00100101'B	
cau	cau	
fie	-	
pi	progind	
uu	uuinf	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	DisconnR(Ti:TI; cau:CAU)	
PDU Type:	DISC_PDU	
Derivation Path:		
Comments:	A default DISCONNECT message matching parametrised value ms -> n.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?100101'B	
cau	cau	
fie	*	
pi	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverAcc_01	
PDU Type:	HOACC_PDU	
Derivation Path:		
Comments:	To match any received HANDOVER ACCESS message	
Field Name	**Field Value**	**Comments**
horf	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverAcc_02(horef:HORF)	
PDU Type:	HOACC_PDU	
Derivation Path:		
Comments:	To match a received HANDOVER ACCESS message with handover reference ` horef`.	
Field Name	**Field Value**	**Comments**
horf	horef	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverCmd_fh(type:CH_TDMA; slot:SN; tsc:TSC; ncc, bcc:B_3; sdccharfcn, mod, hsn:INTEGER; horf:HORF; pwlvl:B_5; synchi:SYNCHI; frql1, frql2:FRQL; cchd:CCHD; chmod:CHMOD; frqchs:FRQCHS; ma:MA; rtdif:TDIF; ta:TA; cphms:CPHMS)	
PDU Type:	HO_CMD_PDU	
Derivation Path:		
Comments:	A derived HANDOVER COMMAND containing TCH/F_FH in non synchronized new CELL A.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101011'B	
cd	CellDescrp(ncc, bcc, sdccharfcn)	
ch1d_at	ChDescrp_fh(type, slot, tsc, INT_TO_BIT((TSPX_MAIO MOD mod), 6), INT_TO_BIT(hsn, 6))	
horf	horf	
pcmd	Pcmd_20(pwlvl)	
synchi	synchi	
frqsl_at	frql1	
frql_at	frql2	
cchd	cchd	
ch1mod	chmod	
ch2d_at	-	
ch2mod	-	
frqchs_at	frqchs	
ma_at	ma	
strt	-	
rtdif	rtdif	
ta	ta	
frqsl_bt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	cphms	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverCmd_nfh(type:CH_TDMA; slot:SN; tsc:TSC; ncc, bcc:B_3; sdccharfcn, tcharfcn:INTEGER; horf:HORF; pwlvl:B_5; synchi:SYNCHI; chmod:CHMOD; rtdif:TDIF; ta:TA; cphms:CPHMS)	
PDU Type:	HO_CMD_PDU	
Derivation Path:		
Comments:	An HANDOVER COMMAND indicating finely synchronised intra cell handover.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101011'B	
cd	CellDescrp(ncc, bcc, sdccharfcn)	
ch1d_at	ChDescrp_nfh(type, slot, tsc, tcharfcn)	
horf	horf	
pcmd	Pcmd_20(pwlvl)	
synchi	synchi	
frqsl_at	-	
frql_at	-	
cchd	-	
ch1mod	chmod	
ch2d_at	-	
ch2mod	-	
frqchs_at	-	
ma_at	-	
strt	-	
rtdif	rtdif	
ta	ta	
frqsl_bt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	cphms	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverCmp_20	
PDU Type:	HO_COM_PDU	
Derivation Path:		
Comments:	A basic received constraint for HANDOVER COMPLETE message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101100'B	
rrcau	'00000000'B	
motdif	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOvFail_01	
PDU Type:	HOFL_PDU	
Derivation Path:		
Comments:	A HANDOVER FAILURE message matching any RR cause.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101000'B	
rrcau	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_01Def(Rr: BITSTRING; Fn: FN; sub:B_2; slot:SN; tsc:TSC; par_arfcn:INTEGER; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/4 channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(sub)), 5), slot, tsc, par_arfcn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_221(Rr:BITSTRING; Fn:FN; ts_ccch:SN; tsc:TSC; par_chtype:CH_TDMA; maio:MAIO; hsn:HSN; ta:TA; ma:MA; iel:LENGTH)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((11 + OC_OctToInt(iel)) * 4) + 1, 1)	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_fh(par_chtype, ts_ccch, tsc, maio, hsn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	ma	
strt	-	
iaroct	IaRestOct(iel)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_nfh(Rr: BITSTRING; Fn: FN; type: CH_TDMA; slot:SN; tsc:TSC; ta:TA; arfcn: INTEGER; pgm:B_2)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign TCH/F or TCH/H channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(pgm)	
chd	ChDescrp_nfh(type, slot, tsc, arfcn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r13(Rr: BITSTRING; Fn: FN; chd:CHD; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign TCH/F channel with a time advance of 30 bit periods.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	chd	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImsiDetach_30(par:MI)	
PDU Type:	IMSID_IN_PDU	
Derivation Path:		
Comments:	IMSI DETACH INDICATION message matching any MS classmark1 value and given IMSI.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?000001'B	
msclm	?	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtAcp_01(newmi: MI; mcc, mnc, lac:OCTETSTRING)	
PDU Type:	LUP_ACP_PDU	
Derivation Path:		
Comments:	To assign a new TMSI with or without mobile identity.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00000010'B	
lai	LocAreaId(mcc, mnc, lac)	
mi	newmi	
fop	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtReq_01(locup:B_2)	
PDU Type:	LUP_RQ_PDU	
Derivation Path:		
Comments:	A LOCATION UPDATING REQUEST message containing location updating type.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?001000'B	
cphksn	?	
lutype	LocUpType(locup)	
lai	?	
msclm	?	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtRej_01(par:REJCAU)	
PDU Type:	LUP_REJ_PDU	
Derivation Path:		
Comments:	A LOCATION UPDATING REJECT message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00000100'B	
rejcau	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_02	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report match any received MSR_RPT_PDU.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_02	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PagingRes_01	
PDU Type:	PG_RES_PDU	
Derivation Path:		
Comments:	To match any received PAGING RESPONSE message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00100111'B	
shoct	'0000'B	
cphksn	?	
msclm	?	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PagingRes_03(cksn:B_3)	
PDU Type:	PG_RES_PDU	
Derivation Path:		
Comments:	To match a received PAGING RESPONSE message with default TMSI, CKSN and classmark2.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00100111'B	
shoct	'0000'B	
cphksn	CphKeySN_01(cksn)	
msclm	ClassMark2	
mi	MiTmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PagingRes_r02	
PDU Type:	PG_RES_PDU	
Derivation Path:	PagingRes_01.	
Comments:	To match the received PAGING RESPONSE message containing IMSI of the IUT for RR tests.	
Field Name	**Field Value**	**Comments**
mi	MiImsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_01	
PDU Type:	PG1_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE1 message requesting any channel with normal paging mode, the mobile identity is TMSI.	
Field Name	**Field Value**	**Comments**
l2_pl	'25'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100001'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	MiTmsi_01	
mi2	-	
p1roct	'2B2B2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_05	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	PAGING REQUEST TYPE1 message with mobile identity 1 being IMSI of the MS.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((3 + OC_LengthOf(MiImsi_01)) * 4) + 1, 1)	
mi1	MiImsi_01	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O, 19 - OC_LengthOf(MiImsi_01))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1Norm	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	PAGING REQUEST TYPE1 message containing normal paging mode, used as paging filling message.	
Field Name	**Field Value**	**Comments**
l2_pl	'15'O	
mi1	MiImsi_02	
p1roct	'2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1Reorg	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	PAGING REQUEST TYPE1 message containing paging reorganisation mode, used as paging filling message.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((3 + OC_LengthOf(MiImsi_02)) * 4) + 1, 1)	
pm	Pm(C_reorg_paging)	
mi1	MiImsi_02	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O, 19 - OC_LengthOf(MiImsi_02))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Phyinfor_01(ta:TA)	
PDU Type:	PHYINFO_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101101'B	
ta	ta	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Release_08(Ti:TI)	
PDU Type:	REL_PDU	
Derivation Path:		
Comments:	A CC RELEASE message with mandatory IE's only used in structured procedures test .	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00101101'B	
cau	-	
cau2	-	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Release_10(Ti:TI)	
PDU Type:	REL_PDU	
Derivation Path:		
Comments:	A CC RELEASE message used to match any received CC RELEASE message with a controllable TI.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?101101'B	
cau	*	
cau2	*	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_03(Ti:TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	To match any received RELEASE message.	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'0?101010'B	
cau	*	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_08(Ti :TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	A RELEASE COMPLETE message containing mandatory IEs only. n -> ms	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'00101010'B	
cau	-	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_01(ti:TI)	
PDU Type:	SETUP_MT_PDU	
Derivation Path:		
Comments:	A SETUP message containing speech bearer capability. n -> ms.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00000101'B	
bcri	-	
bcap1	EFR_Bcap_Speech_MT	
bcap2	-	
fie	-	
pi	-	
sig	-	
cgpn	-	
cgps	-	
cdpn	-	
cdps	-	
llcri	-	
llcmp1	-	
llcmp2	-	
hlcri	-	
hlcmp1	-	
hlcmp2	-	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_02(bcap: BCAP)	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	A SETUP message containing speech bearer capability. ms -> n	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	-	
bcap1	bcap	
bcap2	-	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	*	
llcmp1	*	
llcmp2	*	
hlcri	*	
hlcmp1	*	
hlcmp2	*	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_03(ti:TI)	
PDU Type:	SETUP_MT_PDU	
Derivation Path:		
Comments:	A SETUP message with mandatory IE's only. This is default for BIBO testing.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00000101'B	
bcri	-	
bcap1	-	
bcap2	-	
fie	-	
pi	-	
sig	-	
cgpn	-	
cgps	-	
cdpn	-	
cdps	-	
llcri	-	
llcmp1	-	
llcmp2	-	
hlcri	-	
hlcmp1	-	
hlcmp2	-	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ESetupBcap(bcap: BCAP)	
PDU Type:	ESETUP_PDU	
Derivation Path:		
Comments:	A Emergency call SETUP message with speech bearer capability. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?001110'B	
bcap	bcap	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ESetupNoBcap	
PDU Type:	ESETUP_PDU	
Derivation Path:		
Comments:	A Emergency call SETUP message without speech bearer capability. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?001110'B	
bcap	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_10_UDI(bcap: BCAP; llc1:LLCMP; hlc1:HLCMP)	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	A SETUP message containing bearer capability supported by the mobile station, used for Unrestricted digital information transfer capability. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	-	
bcap1	bcap	
bcap2	-	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	-	
llcmp1	llc1	
llcmp2	-	
hlcri	-	
hlcmp1	hlc1 IF_PRESENT	
hlcmp2	-	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	CallCntrlCap IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_10_3100(bcap: BCAP; llc1:LLCMP; hlc1:HLCMP)	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	A SETUP message containing bearer capability supported by the mobile station. used for 3.1 kHz audio information transfer capability. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	-	
bcap1	bcap	
bcap2	-	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	-	
llcmp1	llc1 IF_PRESENT	
llcmp2	-	
hlcri	-	
hlcmp1	hlc1 IF_PRESENT	
hlcmp2	-	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	CallCntrlCap IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_11_3100(bcri1: RPI; bcap, bcap2nd: BCAP; llc1, llc2:LLCMP; llcri1, hlcri1:RPI; hlc1, hlc2:HLCMP)	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	A SETUP message containing two bearer capabilities supported by the mobile station, used for 3.1 kHz audio information transfer capability. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	bcri1	
bcap1	bcap	
bcap2	bcap2nd	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	llcri1 IF_PRESENT	
llcmp1	llc1 IF_PRESENT	
llcmp2	llc2 IF_PRESENT	
hlcri	hlcri1 IF_PRESENT	
hlcmp1	hlc1 IF_PRESENT	
hlcmp2	hlc2 IF_PRESENT	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	CallCntrlCap IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_20(par: BCAP; ti:TI)	
PDU Type:	SETUP_MT_PDU	
Derivation Path:		
Comments:	A SETUP message containing bearer capability supported by the mobile station.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00000101'B	
bcri	-	
bcap1	par	
bcap2	-	
fie	-	
pi	-	
sig	-	
cgpn	-	
cgps	-	
cdpn	-	
cdps	-	
llcri	-	
llcmp1	-	
llcmp2	-	
hlcri	-	
hlcmp1	-	
hlcmp2	-	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_21(par1:RPI; par2, par3 :BCAP; ti:TI)	
PDU Type:	SETUP_MT_PDU	
Derivation Path:		
Comments:	A SETUP message containing two bearer capabilities supported by the mobile station.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00000101'B	
bcri	par1	
bcap1	par2	
bcap2	par3	
fie	-	
pi	-	
sig	-	
cgpn	-	
cgps	-	
cdpn	-	
cdps	-	
llcri	-	
llcmp1	-	
llcmp2	-	
hlcri	-	
hlcmp1	-	
hlcmp2	-	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SetupInd_01	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	To match any received SETUP message	
Field Name	**Field Value**	**Comments**
ti	TI_09	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	*	
bcap1	?	
bcap2	*	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	*	
llcmp1	*	
llcmp2	*	
hlcri	*	
hlcmp1	*	
hlcmp2	*	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	CallCntrlCap IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SyncInfor(bcc:BCC; ncc:NCC)	
PDU Type:	SCHINFO_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ncc	ncc	
bcc	bcc	
t1	?	
t2	?	
t3_	?	
Detailed Comments:	1. The values of t1, t2 and t3_ shall be correctly set by the L2 simulation module when this message is sent to air on the synchronization channel. It is assumed that there is a set of timebase counters in the L2 module, these counters run continuously and keep the correct timeslot number SN and TDMA frame number FN as long as the test system is poweron , the T1, T2, T3' (values for t1, t2 and t3_) can be derived from these counters.	

PDU Constraint Declaration		
Constraint Name:	SysInf1(cchd:CCHD; maxtx:B_2; txint:B_4; re:B_1)	
PDU Type:	SYSINFO1_PDU	
Derivation Path:		
Comments:	Default parameters for cell A in RR testing of GSM 900	
Field Name	**Field Value**	**Comments**
l2_pl	'55'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011001'B	
cchd	cchd	
rachcp	RachCntrlPara(maxtx, txint, re)	
si1roct	'2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf2(bcchfl: NCD; maxtx:B_2; txint:B_4; re:B_1; nccp:NCCP)	
PDU Type:	SYSINFO2_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION TYPE2 containing default neighbour cells description	
Field Name	**Field Value**	**Comments**
l2_pl	'59'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011010'B	
bcchfl	bcchfl	
nccp	nccp	
rachcp	RachCntrlPara(maxtx, txint, re)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf2bis(bcchflegsm: NCD)	
PDU Type:	SYSINFO2bis_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION 2bis in cell A under EGSM with the ARFCN list = {988, 990, 1003}.	
Field Name	**Field Value**	**Comments**
l2_pl	'55'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00000010'B	
xbcchfl	bcchflegsm	
rachcp	RachCntrlPara_r01	
si2bisroct	'00'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf3(ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
PDU Type:	SYSINFO3_PDU	
Derivation Path:		
Comments:	CCCH combined or not with SDCCH	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011011'B	
ci	ci	
lai	LocAreaId(mcc, mnc, lac)	
ccd	ccd	
co	co	
csp	CellSelPara(crh, mtmc, neci)	
rachcp	RachCntrlPara(maxtx, txint, re)	
si3roct	SI3RestOctetDef	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf4(mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
PDU Type:	SYSINFO4_PDU	
Derivation Path:		
Comments:	A SYSTEM INFORMATION TYPE 4 message containing default values	
Field Name	**Field Value**	**Comments**
l2_pl	'31'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011100'B	
lai	LocAreaId(mcc, mnc, lac)	
csp	CellSelPara(crh, mtmc, neci)	
rachcp	RachCntrlPara(maxtx, txint, re)	
cbchd	-	
cbchma	-	
si4roct	SI4RestOctetDef	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf4_CBMS(mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; cbchd:CHD)	
PDU Type:	SYSINFO4_PDU	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 4 message for SMSCB with the values or GSM 11.10, 34.3	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011100'B	
lai	LocAreaId(mcc, mnc, lac)	
csp	CellSelPara(crh, mtmc, neci)	
rachcp	RachCntrlPara(maxtx, txint, re)	
cbchd	cbchd	
cbchma	MoblAllc_01iei	
si4roct	SI4RestOctet_CBMS	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf5(bcchfl: NCD)	
PDU Type:	SYSINFO5_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION TYPE 5 containing default neighbour cells description for GSM900	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00011101'B	
bcchfl	bcchfl	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf5bis(bcchfl: NCD)	
PDU Type:	SYSINFO5bis_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION TYPE 5bis containing partial neighbour cells description with 1 frequency for DCS1800.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00000101'B	
xbcchfl	bcchfl	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf6(ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; nccp:NCCP)	
PDU Type:	SYSINFO6_PDU	
Derivation Path:		
Comments:	A SYSTEM INFORMATION TYPE 6 message containing default parameters for RR testing.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00011110'B	
ci	ci	
lai	LocAreaId(mcc, mnc, lac)	
co	co	
nccp	nccp	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	TmsiReallocCmd(par:MI; mcc, mnc, lac: OCTETSTRING)	
PDU Type:	TMSIRE_CMD_PDU	
Derivation Path:		
Comments:	TMSI REALLOCATION COMMAND message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00011010'B	
lai	LocAreaId(mcc, mnc, lac)	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	TmsiReallocComp	
PDU Type:	TMSIRE_COM_PDU	
Derivation Path:		
Comments:	TMSI REALLOCATION COMPLETE message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?011011'B	
Detailed Comments:		

Dynamic Part

Test Cases

Test Group EFR

Test Case Dynamic Behaviour					
Test Case Name:	TC_26_12_1				
Group:	GSM_MS_EFR_4220/EFR/				
Purpose:	To verify that the MS (in an RR connected state) with a TCH/F allocated acknowledging a CHANNEL MODE MODIFY message by sending a CHANNEL MODE MODIFY ACKNOWLEDGEMENT message specifying and switching to the correct mode. - the new mode if that mode is supported - the old mode if the new mode is not supported. This shall be verified for the channel modes - speech full rate versions 1/2 and 3 - data 9.6 Kb/s - data 4.8 Kb/s full rate - data 2.4 Kb/s full rate.				
Default:	OtherEventsFail				
Comments:	apply only to the MS supporting TCH/F (EFR).				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		START T_guard(300)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_02, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			1.
3		+gsmOrDcs			
4		+FullRateCh_A_1_nociph(C_Ass, TSPX_TmSltDef, TSPX_TscDef, ChMod_speech, FreqTCH(C_arfcn_tchA), FreqTCH(C_arfcn_tchAd), TimingAdv(30), C_BABR_0, C_cch_1Comb, C_BPM_3)			
5		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_05)		
6		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
7		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
8		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_r13(TCV_Rr, TCV_Fn, TCV_chd1, TimingAdv(30)))		2.
9		L?DL_EstInPgRes	PagingRes(PagingRes_r02)		
10		ACTIVATE(OtherEventsFail)			Restore Normal default
11		+test_channel_modes			
		test_channel_modes			
12		+set_FullSpeech_v2			
13		+test_24DataF			
14		+set_FullSpeech_v2			
15		+test_48DataF			
16		+set_FullSpeech_v2			
17		+test_96DataF			

18		+set_FullSpeech_v2			
19		+test_FullSpeech_v1			
20		+set_FullSpeech_v2			
21		+test_FullSpeech_v3			
22		+PostMainLinkRel(TCV_chTch)			
		set_FullSpeech_v2			
23		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(ChMod_EFR2, TCV_chd1))		
24	E_0001	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_EFR2, TCV_chd1))	(P)	
		test_FullSpeech_v3			
25		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(ChMod_EFR3, TCV_chd1))		
26		[TSPC_EFR_Speech_v3]			
27	E_0002	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_EFR3, TCV_chd1))	(P)	
28		[NOT TSPC_EFR_Speech_v3]			
29	E_0003	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_EFR2, TCV_chd1))	(P)	
		test_FullSpeech_v1			
30		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(ChMod_speech, TCV_chd1))		
31	E_0004	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_speech, TCV_chd1))	(P)	
		test_24DataF			
32		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(ChMod_3k, TCV_chd1))		
33		[TSPC_24DataF]			
34	E_0005	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_3k, TCV_chd1))	(P)	
35		[NOT TSPC_24DataF]			
36	E_0006	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_EFR2, TCV_chd1))	(P)	
		test_48DataF			
37		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(		

			ChMod_6k, TCV_chd1))		
38		[TSPC_48DataF]			
39	E_0007	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_6k, TCV_chd1))	(P)	
40		[NOT TSPC_48DataF]			
41	E_0008	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_EFR2, TCV_chd1))	(P)	
		test_96DataF			
42		L!DL_DatRqChmmo	ChmmoReqSnd(TCV_chTch, ChmomoReq_01(ChMod_12k, TCV_chd1))		
43		[TSPC_96Data]			
44	E_0009	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_12k, TCV_chd1))	(P)	
45		[NOT TSPC_96Data]			
46	E_0010	L?DL_DatInChmmoAck	ChmmoAckRcv(TCV_chTch, ChmomoAck_01(ChMod_EFR2, TCV_chd1))	(P)	
		gsmOrDcs			
47		[TSPC_PGSM OR TSPC_EGSM]			
48		(TCV_chd1 := ChDescrp_nfh('00001'B, TSPX_TmSltDef, TSPX_TscDef, C_arfcn_tchA))			
49		[TSPC_DCS]			
50		(TCV_chd1 := ChDescrp_nfh('00001'B, TSPX_TmSltDef, TSPX_TscDef, C_arfcn_tchAd))			

Detailed Comments: 1. Default system infirmations for RR testing.
2. TCH/F channel.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_1

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/F, sv2, without frequency hopping towards a TCH/F,sv2, without frequency hopping and,

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := TSPX_TimadvA, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_201_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell B is 100 in advance of frame number in cell A.
6		+FullRateCh_A_1(C_Ass, TCV_asscmd_ts, TCV_tsc, ChMod_EFR2, FreqTCH_omit, FreqTCH_omit, TSPX_TCHcarrierA_ho, TSPX_TCHcarrierA_hod, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, NonFH on cellA
7		+Est_MT_CallNonFH(TimingAdv(TCV_T), C_Full_v2, TSPX_TCHHSubDef, TSPX_RANDDef)			To establish a MT call with non hopping channel.
8	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
9		(TCV_n := 500, TCV_Cntref := TSPX_NoOfHoAccA, TCV_T := 20, TCV_Horf:= TSPX_HoRefA, TCV_ts := TSPX_TmSltB, TCV_tsc := TSPX_TscA, TCV_chtype := '00001'B)			Parameters initialization for Handover procedure.
10		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, nonFH on cellB
11		+ltree_arfcnB			
12		(TCV_HoCMD := HandOverCmd_nfh(TCV_chtype, TCV_ts, TCV_tsc,			

			C_NCC, C_BCC, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
13			(TCV_ch := TCV_chTch)			
14			(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
15			L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch1, TCV_HoCMD)		HO command
16			L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17			+RR_hocomp1(TCV_n, TCV_chTch1)			
18			+ChanRel_end(TCV_ch)			
		ltree_arfcnB				
19		[TSPC_PGSM OR TSPC_EGSM]				
20		(TCV_chdescr_arfcn := C_BCCHcarrierB_ho, TCV_tch_arfcn := TSPX_TCHcarrierB_ho)				
21		[TSPC_DCS]				
22		(TCV_chdescr_arfcn := C_BCCHcarrierB_hod, TCV_tch_arfcn := TSPX_TCHcarrierB_hod)				

Detailed Comments: IUT enters state U10 with TCH/F_nonFH in cell A and then HandOver from TCH/F_nonFH sv2 of cell A to TCH/F_nonFH sv2 in cell B

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_2

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/F sv2, without frequency hopping on cell B towards a TCH/F, sv2, with frequency hopping on cell A

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := 20, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Bman, CellChDes_201_Bd, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellA_ho(C_E_neighbourdefault, C_BCCHcarrierA_ho, C_BCCHcarrierA_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Aman, CellChDes_201_Ad, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell A is 100 in advance of frame number in cell B.
6		+FullRateCh_B_1(C_Asynho, TCV_asscmd_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, NonFH on cellB
7		+ltree_arfcnB			
8		+Est_MT_CallNonFH(TimingAdv(TCV_T), C_Full_v2, TSPX_TCHHSubDef, TSPX_RANDDef)			To establish a MTcall with non hopping channel.
9	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
10		(TCV_n := 500, TCV_Cntref := TSPX_NoOfHoAccB, TCV_T := TSPX_TimadvA, TCV_ts := TSPX_TmSltNotZero, TCV_Horf:= TSPX_HoRefB, TCV_tsc := TSPX_TscB, TCV_chtype := '00001'B)			Parameters initialization for Handover procedure.
11		+FullRateCh_A_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCHa_hof1, FreqTCHa_hof1d, C_Noarfcn, C_Noarfcn, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, FH on cellA
12		(TCV_ch := TCV_chTch)			
13		+ltree_BuildHoCMD			
14		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
15		L!DL_DatRqHoCmd	HndOvSnd(		HO command

			TCV_chTch1, TCV_HoCMD)		
16		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17		+RR_hocomp1(TCV_n, TCV_chTch1)			
18		+ChanRel_end(TCV_ch)			
	ltree_arfcnB				
19	[TSPC_PGSM OR TSPC_EGSM]				
20	(TCV_tch_arfcn := TSPX_TCHcarrierB_ho)				
21	[TSPC_DCS]				
22	(TCV_tch_arfcn := TSPX_TCHcarrierB_hod)				
	ltree_BuildHoCMD				
23	[TSPC_PGSM OR TSPC_EGSM]				
24	(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnA, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_22, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				
25	[TSPC_DCS]				
26	(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnAd_4, 9, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_20_Ad, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				

Detailed Comments: HO from TCH/F_nonFH sv2 of cell A to TCH/F_nonFH sv2 in cell B

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_3

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/F, sv2, with frequency hopping towards a TCH/F, sv2, without frequency hopping and,

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := TSPX_TimadvA, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_201_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell B is 100 in advance of frame number in cell A.
6		+FullRateCh_A_1(C_Asynho, TCV_asscmd_ts, TCV_tsc, ChMod_EFR2, FreqTCHa_hof1, FreqTCHa_hof1d, C_Noarfcn, C_Noarfcn, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, FH on cellA
7		+Est_MT_Call_FH(TimingAdv(TCV_T), C_Full_v2, INT_TO_BIT(TSPX_MAIO, 6), TSPX_HSN, TSPX_RANDDef, TCV_asscmd_ts, TSPX_TscDef, TSPX_TCHHSubDef)			To establish a MTcall with hopping channel.
8	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
9		(TCV_n := 500, TCV_Cntref := TSPX_NoOfHoAccC, TCV_T := 20, TCV_ts := TSPX_TmSltC, TCV_Horf:= TSPX_HoRefC, TCV_tsc := TSPX_TscC, TCV_chtype := '00001'B)			Parameters initialization for Handover procedure.
10		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, NonFH on cellB
11		+ltree_arfcnB			
12		(TCV_HoCMD := HandOverCmd_nfh(TCV_chtype, TCV_ts, TCV_tsc,			

			C_NCC, C_BCC, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_report_otd, C_non_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
13			(TCV_ch := TCV_chTch)			
14			(TCV_Fn := OM_ComingFn(TCV_chTch1), TCV_Null:= OM_SendNextOn(TCV_chTch1, TCV_Fn), TCV_Strt := OC_StartTime(TCV_Fn,(C_StartingTimeHO + 100), 1), TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			starting time is set to 1.1 s plus 100 frames
15			L!DL_DatRqHoCmd(DL_DatRqHoCmd.msg.strt:= TCV_Strt)	HndOvSnd(TCV_chTch1, TCV_HoCMD)		HO command
16			L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17			+RR_hocomp1(TCV_n, TCV_chTch1)			
18			+ltree_FirstBurstOnTimeCHK			
19			+ChanRel_end(TCV_ch)			
		ltree_arfcnB				
20		[TSPC_PGSM OR TSPC_EGSM]				
21		(TCV_chdescr_arfcn := C_BCCHcarrierB_ho, TCV_tch_arfcn := TSPX_TCHcarrierB_ho)				
22		[TSPC_DCS]				
23		(TCV_chdescr_arfcn := C_BCCHcarrierB_hod, TCV_tch_arfcn := TSPX_TCHcarrierB_hod)				
		ltree_FirstBurstOnTimeCHK				
24		+FnArith(TCV_FnBurst, TCV_Strt.fn)				
25	E_0126	[(TCV_Time > 5) OR(TCV_Time < 0)]			(F)	
26	E_0127	[(TCV_Time >= 0) AND(TCV_Time < 5)]			(P)	

Detailed Comments: HO from TCH/F_FH sv2 of cell A to TCH/F_nonFH sv2 in cell B

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_4

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/F, sv2, without frequency hopping towards a TCH/F, sv1, without frequency hopping and,

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := 20, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Bman, CellChDes_201_Bd, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellA_ho(C_E_neighbourdefault, C_BCCHcarrierA_ho, C_BCCHcarrierA_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Aman, CellChDes_201_Ad, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell A is 100 in advance of frame number in cell B.
6		+FullRateCh_B_1(C_Asynho, TCV_asscmd_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, NonFH on cellB
7		+ltree_arfcnB			
8		+Est_MT_CallNonFH(TimingAdv(TCV_T), C_Full_v2, TSPX_TCHHSubDef, TSPX_RANDDef)			To establish a MTcall with non hopping channel.
9	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
10		(TCV_n := 500, TCV_Cntref := TSPX_NoOfHoAccA, TCV_T := 20, TCV_ts := TSPX_TmSltD, TCV_Horf:= TSPX_HoRefD, TCV_tsc := TSPX_TscD, TCV_chtype := '00001'B)			Parameters initialization for Handover procedure.
11		+FullRateCh_A_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_speech, FreqTCH_omit, FreqTCH_omit, TSPX_TCHcarrierA_ho, TSPX_TCHcarrierA_hod, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv1, NonFH on cellA
12		+ltree_arfcnA			
13		(TCV_HoCMD := HandOverCmd_nfh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC,			

		TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), ChMod_speech_iei, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
14		(TCV_ch := TCV_chTch)			
15		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
16		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch1, TCV_HoCMD)		HO command
17		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
18		+RR_hocomp1(TCV_n, TCV_chTch1)			
19		+ChanRel_end(TCV_ch)			
	ltree_arfcnA				
20	[TSPC_PGSM OR TSPC_EGSM]				
21	(TCV_chdescr_arfcn := C_arfcnA, TCV_tch_arfcn := TSPX_TCHcarrierA_ho)				
22	[TSPC_DCS]				
23	(TCV_chdescr_arfcn := C_arfcnAd_4, TCV_tch_arfcn := TSPX_TCHcarrierA_hod)				
	ltree_arfcnB				
24	[TSPC_PGSM OR TSPC_EGSM]				
25	(TCV_tch_arfcn := TSPX_TCHcarrierB_ho)				
26	[TSPC_DCS]				
27	(TCV_tch_arfcn := TSPX_TCHcarrierB_hod)				

Detailed Comments: HO from TCH/F_nonFH sv2 of cell B to TCH/F_nonFH sv1 in cell A

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_5

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/F, sv2, without frequency hopping towards a TCH/F without frequency hopping

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := 20, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_201_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell B is 100 in advance of frame number in cell A.
6		+FullRateCh_A_1(C_Asynho, TCV_asscmd_ts, TCV_tsc, ChMod_speech, FreqTCH_omit, FreqTCH_omit, TSPX_TCHcarrierA_ho, TSPX_TCHcarrierA_hod, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv1, NonFH on cellA
7		+Est_MT_CallNonFH(TimingAdv(TCV_T), C_Full, TSPX_TCHHSubDef, TSPX_RANDDef)			To establish a MT call with non hopping channel.
8	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
9		(TCV_n := 500, TCV_Cntref := TSPX_NoOfHoAccB, TCV_T := TSPX_TimadvB, TCV_ts := TSPX_TmSltE, TCV_Horf:= TSPX_HoRefE, TCV_tsc := TSPX_TscE, TCV_chtype := '00001'B)			Parameters initialization for Handover procedure.
10		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, nonFH on cellB
11		+ltree_arfcnB			
12		(TCV_HoCMD := HandOverCmd_nfh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC,			

			TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
13			(TCV_ch := TCV_chTch)			
14			(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
15			L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch1, TCV_HoCMD)		HO command
16			L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17			+RR_hocomp1(TCV_n, TCV_chTch1)			
18			+ChanRel_end(TCV_ch)			
		ltree_arfcnB				
19		[TSPC_PGSM OR TSPC_EGSM]				
20		(TCV_chdescr_arfcn := C_BCCHcarrierB_ho, TCV_tch_arfcn := TSPX_TCHcarrierB_ho)				
21		[TSPC_DCS]				
22		(TCV_chdescr_arfcn := C_BCCHcarrierB_hod, TCV_tch_arfcn := TSPX_TCHcarrierB_hod)				

Detailed Comments: HandOver from TCH/F_nonFH sv1 of cell A to TCH/F_nonFH sv2 in cell B

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_6

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/F sv2, without frequency hopping on cell B towards a TCH/F, sv1, with frequency hopping on cell A

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := TSPX_TimadvB, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Bman, CellChDes_201_Bd, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellA_ho(C_E_neighbourdefault, C_BCCHcarrierA_ho, C_BCCHcarrierA_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Aman, CellChDes_201_Ad, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell A is 100 in advance of frame number in cell B.
6		+FullRateCh_B_1(C_Asynho, TCV_asscmd_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, NonFH on cellB
7		+ltree_arfcnB			
8		+Est_MT_CallNonFH(TimingAdv(TCV_T), C_Full_v2, TSPX_TCHHSubDef, TSPX_RANDDef)			To establish MTcall with NonFH channel.
9	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
10		(TCV_n := 500, TCV_Cntref := TSPX_NoOfHoAccC, TCV_T := TSPX_TimadvC, TCV_ts := TSPX_TmSltNotZero, TCV_Horf:= TSPX_HoRefF, TCV_tsc := TSPX_TscF, TCV_chtype := '00001'B)			
11		+FullRateCh_A_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_speech, FreqTCHa_hof1, FreqTCHa_hof1d, C_Noarfcn, C_Noarfcn, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			
12		(TCV_ch := TCV_chTch)			
13		+ltree_BuildHoCMD			
14		(TCV_Null := OM_SendNextOnHO(TCV_			

		ch, TCV_Cntref))			
15		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch1, TCV_HoCMD)		HO command
16		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17		+RR_hocomp1(TCV_n, TCV_chTch1)			
18		+ChanRel_end(TCV_ch)			
	ltree_arfcnB				
19	[TSPC_PGSM OR TSPC_EGSM]				
20	(TCV_tch_arfcn := TSPX_TCHcarrierB_ho)				
21	[TSPC_DCS]				
22	(TCV_tch_arfcn := TSPX_TCHcarrierB_hod)				
	ltree_BuildHoCMD				
23	[TSPC_PGSM OR TSPC_EGSM]				
24	(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnA, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_Omit, ChMod_speech_iei, Freqchseq_22, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				
25	[TSPC_DCS]				
26	(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnAd_4, 9, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_20_Ad, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				

Detailed Comments: HO from TCH/F_nonFH sv2 of cell B to TCH/F_FH sv1 in cell A

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_7

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/F sv1, with frequency hopping on cell A towards a TCH/F, sv2, with frequency hopping on cell B

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := TSPX_TimadvA, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_201_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell B is 100 in advance of frame number in cell A.
6		+FullRateCh_A_1(C_Asynho, TCV_asscmd_ts, TCV_tsc, ChMod_speech, FreqTCHa_hof1, FreqTCHa_hof1d, C_Noarfcn, C_Noarfcn, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv1, FH on cellA
7		+Est_MT_Call_FH(TimingAdv(TCV_T), C_Full, INT_TO_BIT(TSPX_MAIO, 6), TSPX_HSN, TSPX_RANDDef, TCV_asscmd_ts, TSPX_TscDef, TSPX_TCHHSubDef)			To establish a MTcall with hopping channel.
8	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
9		(TCV_n := 750, TCV_Cntref := TSPX_NoOfHoAccA, TCV_T := 20, TCV_ts := TSPX_TmSltNotZero1, TCV_Horf:= TSPX_HoRefG, TCV_tsc := TSPX_TscG, TCV_chtype := '00001'B)			Parameters initialization for Handover procedure.
10		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCHb_hof1, FreqTCHb_hof1d, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, FH on cellB
11		(TCV_ch := TCV_chTch)			
12		+ltree_BuildHoCMD			
13		(TCV_Fn := OM_ComingFn(TCV_chTch1), TCV_Null:= OM_SendNextOn(TCV_chTch1, TCV_Fn),			starting time is set to 1.1 s plus 100 frames

		TCV_Strt := OC_StartTime(TCV_Fn,(C_StartingTimeHO + 100), 1))			
14		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
15		L!DL_DatRqHoCmd(DL_DatRqHoCmd.msg.strt:= TCV_Strt)	HndOvSnd(TCV_chTch1, TCV_HoCMD)		HO command
16		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17		+RR_hocomp1(TCV_n, TCV_chTch1)			
18		+ltree_FirstBurstOnTimeCHK			
19		+ChanRel_end(TCV_ch)			
	ltree_BuildHoCMD				
20	[TSPC_PGSM OR TSPC_EGSM]				
21	(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_BCCHcarrierB_ho, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_20_B3, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				
22	[TSPC_DCS]				
23	(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_BCCHcarrierB_hod, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_20_B3d, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				
	ltree_FirstBurstOnTimeCHK				
24	+FnArith(TCV_FnBurst, TCV_Strt.fn)				
25	E_0136	[(TCV_Time > 5) OR(TCV_Time < 0)]		(F)	
26	E_0137	[(TCV_Time >= 0) AND(TCV_Time < 5)]		(P)	

Detailed Comments: HO from TCH/F_FH sv1 of cell A to TCH/F_FH sv2 in cell B

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_8

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/F, sv2, without frequency hopping towards a TCH/F, sv1, with frequency hopping

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := 20, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Bman, CellChDes_201_Bd, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellA_ho(C_E_neighbourdefault, C_BCCHcarrierA_ho, C_BCCHcarrierA_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Aman, CellChDes_201_Ad, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell A is 100 in advance of frame number in cell B.
6		+FullRateCh_B_1(C_Asynho, TCV_asscmd_ts, TCV_tsc, ChMod_EFR2, FreqTCHb_hof1, FreqTCHb_hof1d, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, FH on cellB
7		+Est_MT_Call_FH(TimingAdv(TCV_T), C_Full_v2, INT_TO_BIT(TSPX_MAIO, 6), TSPX_HSN, TSPX_RANDDef, TCV_asscmd_ts, TSPX_TscDef, TSPX_TCHHSubDef)			To establish a MTcall with hopping channel.
8	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
9		(TCV_n := 750, TCV_Cntref := TSPX_NoOfHoAccB, TCV_T := TSPX_TimadvA, TCV_ts := TSPX_TmSltNotZero, TCV_Horf:= TSPX_HoRefH, TCV_tsc := TSPX_TscA, TCV_chtype := '00001'B)			Parameters initialization for Handover procedure.
10		+FullRateCh_A_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_speech, FreqTCHa_hof1, FreqTCHa_hof1d, C_Noarfcn, C_Noarfcn, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv1, NonFH on cellA
11		(TCV_ch := TCV_chTch)			
12		+ltree_BuildHoCMD			
13		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
14		L!DL_DatRqHoCmd	HndOvSnd(		HO command

			TCV_chTch1, TCV_HoCMD)		
15		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
16		+RR_hocomp1(TCV_n, TCV_chTch1)			
17		+ChanRel_end(TCV_ch)			
		ltree_BuildHoCMD			
18		[TSPC_PGSM OR TSPC_EGSM]			
19		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnA, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_Omit, ChMod_speech_iei, Freqchseq_22, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
20		[TSPC_DCS]			
21		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnAd_4, 9, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_20_Ad, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			

Detailed Comments: HandOver from TCH/F_FH sv2 of cell B to TCH/F_FH sv1 in cell A

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_9

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/F, sv1, with frequency hopping towards a TCH/F, sv2, without frequency hopping

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := TSPX_TimadvA, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_201_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell B is 100 in advance of frame number in cell A.
6		+FullRateCh_A_1(C_Asynho, TCV_asscmd_ts, TCV_tsc, ChMod_speech, FreqTCHa_hof1, FreqTCHa_hof1d, C_Noarfcn, C_Noarfcn, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv1, FH on cellA
7		+Est_MT_Call_FH(TimingAdv(TCV_T), C_Full, INT_TO_BIT(TSPX_MAIO, 6), TSPX_HSN, TSPX_RANDDef, TCV_asscmd_ts, TSPX_TscDef, TSPX_TCHHSubDef)			To establish a MTcall with hopping channel.
8	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
9		(TCV_n := 750, TCV_Cntref := TSPX_NoOfHoAccC, TCV_T := TSPX_TimadvB, TCV_ts := TSPX_TmSltF, TCV_Horf:= TSPX_HoRefI, TCV_tsc := TSPX_TscB, TCV_chtype := '00001'B)			Parameters initialization for Handover procedure.
10		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, nonFH on cellB
11		+ltree_arfcnB			
12		(TCV_HoCMD := HandOverCmd_nfh(			

			TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
13			(TCV_ch := TCV_chTch)			
14			(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
15			L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch1, TCV_HoCMD)		HO command
16			L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17			+RR_hocomp1(TCV_n, TCV_chTch1)			
18			+ChanRel_end(TCV_ch)			
		ltree_arfcnB				
19		[TSPC_PGSM OR TSPC_EGSM]				
20		(TCV_chdescr_arfcn := C_BCCHcarrierB_ho, TCV_tch_arfcn := TSPX_TCHcarrierB_ho)				
21		[TSPC_DCS]				
22		(TCV_chdescr_arfcn := C_BCCHcarrierB_hod, TCV_tch_arfcn := TSPX_TCHcarrierB_hod)				

Detailed Comments: HandOver from TCH/F_nonFH sv1 of cell A to TCH/F_nonFH sv2 in cell B

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_10

Group: GSM_MS_EFR_4220/EFR/

Purpose: o test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/F, sv2, without frequency hopping towards a TCH/H, sv1, with frequency hopping

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := TSPX_TimadvB, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Bman, CellChDes_201_Bd, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellA_ho(C_E_neighbourdefault, C_BCCHcarrierA_ho, C_BCCHcarrierA_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Aman, CellChDes_201_Ad, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell A is 100 in advance of frame number in cell B.
6		+FullRateCh_B_1(C_Asynho, TCV_asscmd_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, NonFH on cellB
7		+ltree_arfcnB			
8		+Est_MT_CallNonFH(TimingAdv(TCV_T), C_Full_v2, TSPX_TCHHSubDef, TSPX_RANDDef)			To establish MTcall with NonFH channel.
9	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
10		(TCV_n := 750, TCV_Cntref := TSPX_NoOfHoAccD, TCV_T := TSPX_TimadvC, TCV_ts := TSPX_TmSltNotZero1, TCV_Horf:= TSPX_HoRefJ, TCV_tsc := TSPX_TscC, TCV_chtype := INT_TO_BIT((2+BIT_TO_INT(TSPX_TCHHSubA)),5))			
11		+HalfRateCh_A_1(TSPX_TCHHSubA, C_Asynho, TCV_ts, TCV_tsc, ChMod_speech, FreqTCHa_hof1, FreqTCHa_hof1d, C_Noarfcn, C_Noarfcn, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			
12		(TCV_ch := TCV_chTch)			

13		+ltree_BuildHoCMD			
14		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
15		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch1, TCV_HoCMD)		HO command
16		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17		+RR_hocomp1(TCV_n, TCV_chTch1)			
18		+ChanRel_end(TCV_ch)			
	ltree_arfcnB				
19	[TSPC_PGSM OR TSPC_EGSM]				
20	(TCV_tch_arfcn := TSPX_TCHcarrierB_ho)				
21	[TSPC_DCS]				
22	(TCV_tch_arfcn := TSPX_TCHcarrierB_hod)				
	ltree_BuildHoCMD				
23	[TSPC_PGSM OR TSPC_EGSM]				
24	(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnA, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_Omit, ChMod_speech_iei, Freqchseq_22, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				
25	[TSPC_DCS]				
26	(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnAd_4, 9, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_20_Ad, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				

Detailed Comments: HandOver from TCH/F_non sv2 of cell B to TCH/H_FH sv1 in cell A

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_11

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/H, sv1, with frequency hopping towards a TCH/F, sv2, with frequency hopping

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := TSPX_TimadvA, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_201_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell B is 100 in advance of frame number in cell A.
6		+HalfRateCh_A_1(TSPX_TCHHSubA, C_Asynho, TCV_asscmd_ts, TCV_tsc, ChMod_speech, FreqTCHa_hof1, FreqTCHa_hof1d, C_Noarfcn, C_Noarfcn, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/H, sv1, FH on cellA
7		+Est_MT_Call_FH(TimingAdv(TCV_T), C_Half, INT_TO_BIT(TSPX_MAIO, 6), TSPX_HSN, TSPX_RANDDef, TCV_asscmd_ts, TSPX_TscDef, TSPX_TCHHSubDef)			To establish a MTcall with hopping channel.
8	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
9		(TCV_n := 750, TCV_Cntref := TSPX_NoOfHoAccA, TCV_T := 20, TCV_ts := TSPX_TmSltNotZero, TCV_Horf:= TSPX_HoRefA, TCV_tsc := TSPX_TscD, TCV_chtype := '00001'B)			Parameters initialization for Handover procedure.
10		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCHb_hof1, FreqTCHb_hof1d, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, FH on cellB
11		(TCV_ch := TCV_chTch)			
12		+Itree_BuildHoCMD			
13		(TCV_Fn := OM_ComingFn(TCV_chTch1), TCV_Null:= OM_SendNextOn(			starting time is set to 1.1 s plus 100 frames

		TCV_chTch1, TCV_Fn), TCV_Strt := OC_StartTime(TCV_Fn,(C_StartingTimeHO + 100), 1))			
14		(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
15		L!DL_DatRqHoCmd(DL_DatRqHoCmd.msg.strt:= TCV_Strt)	HndOvSnd(TCV_chTch1, TCV_HoCMD)		HO command
16		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17		+RR_hocomp1(TCV_n, TCV_chTch1)			
18		+ChanRel_end(TCV_ch)			
		ltree_BuildHoCMD			
19		[TSPC_PGSM OR TSPC_EGSM]			
20		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_BCCHcarrierB_ho, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_20_B3, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
21		[TSPC_DCS]			
22		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_BCCHcarrierB_hod, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_20_B3d, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
		ltree_FirstBurstOnTimeCHK			
23		+FnArith(TCV_FnBurst, TCV_Strt.fn)			
24	E_0146	[(TCV_Time > 5) OR(TCV_Time < 0)]		(F)	
25	E_0147	[(TCV_Time >= 0) AND(TCV_Time < 5)]		(P)	

Detailed Comments: HandOver from TCH/H_FH sv1 of cell A to TCH/F_FH sv2 in cell B

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_12

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/F, sv2, with frequency hopping towards a TCH/H, sv1, with frequency hopping

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := 20, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Bman, CellChDes_201_Bd, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellA_ho(C_E_neighbourdefault, C_BCCHcarrierA_ho, C_BCCHcarrierA_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Aman, CellChDes_201_Ad, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell A is 100 in advance of frame number in cell B.
6		+FullRateCh_B_1(C_Asynho, TCV_asscmd_ts, TCV_tsc, ChMod_EFR2, FreqTCHb_hof1, FreqTCHb_hof1d, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, FH on cellB
7		+Est_MT_Call_FH(TimingAdv(TCV_T), C_Full_v2, INT_TO_BIT(TSPX_MAIO, 6), TSPX_HSN, TSPX_RANDDef, TCV_asscmd_ts, TSPX_TscDef, TSPX_TCHHSubDef)			To establish a MTcall with hopping channel.
8	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
9		(TCV_n := 750, TCV_Cntref := TSPX_NoOfHoAccE, TCV_T := TSPX_TimadvA, TCV_ts := TSPX_TmSltNotZero1, TCV_Horf:= TSPX_HoRefB, TCV_tsc := TSPX_TscD, TCV_chtype := INT_TO_BIT((2+BIT_TO_INT(TSPX_TCHHSubDef)),5))			Parameters initialization for Handover procedure.
10		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Asynho, TCV_ts, TCV_tsc, ChMod_speech, FreqTCHa_hof1, FreqTCHa_hof1d, C_Noarfcn, C_Noarfcn, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/H, sv1, FH on cellA
11		(TCV_ch := TCV_chTch)			
12		+Itree_BuildHoCMD			
13		(TCV_Null := OM_SendNextOnHO(TCV_ch			

		, TCV_Cntref))			
14		L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch1, TCV_HoCMD)		HO command
15		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
16		+RR_hocomp1(TCV_n, TCV_chTch1)			
17		+ChanRel_end(TCV_ch)			
		ltree_BuildHoCMD			
18		[TSPC_PGSM OR TSPC_EGSM]			
19		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnA, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_Omit, ChMod_speech_iei, Freqchseq_22, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
20		[TSPC_DCS]			
21		(TCV_HoCMD := HandOverCmd_fh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, C_arfcnAd_4, 9, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_20_Ad, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			

Detailed Comments: HandOver from TCH/F_FH sv2 of cell B to TCH/H_FH sv1 in cell A

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_13

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/H, sv1, with frequency hopping towards a TCH/F,sv2, without frequency hopping

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := TSPX_TimadvA, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_201_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell B is 100 in advance of frame number in cell A.
6		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Asynho, TCV_asscmd_ts, TCV_tsc, ChMod_speech, FreqTCHa_hof1, FreqTCHa_hof1d, C_Noarfcn, C_Noarfcn, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/H, sv1, FH on cellA
7		+Est_MT_Call_FH(TimingAdv(TCV_T), C_Half, INT_TO_BIT(TSPX_MAIO, 6), TSPX_HSN, TSPX_RANDDef, TCV_asscmd_ts, TSPX_TscDef, TSPX_TCHHSubDef)			To establish a MTcall with hopping channel.
8	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
9		(TCV_n := 750, TCV_Cntref := TSPX_NoOfHoAccB, TCV_T := 20, TCV_ts := TSPX_TmSltG, TCV_Horf:= TSPX_HoRefC, TCV_tsc := TSPX_TscD, TCV_chtype := '00001'B)			Parameters initialization for Handover procedure.
10		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, nonFH on cellB
11		+ltree_arfcnB			
12		(TCV_HoCMD := HandOverCmd_nfh(TCV_chtype, TCV_ts, TCV_tsc,			

			C_NCC, C_BCC, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
13			(TCV_ch := TCV_chTch)			
14			(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
15			L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch1, TCV_HoCMD)		HO command
16			L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17			+RR_hocomp1(TCV_n, TCV_chTch1)			
18			+ChanRel_end(TCV_ch)			
		ltree_arfcnB				
19		[TSPC_PGSM OR TSPC_EGSM]				
20		(TCV_chdescr_arfcn := C_BCCHcarrierB_ho, TCV_tch_arfcn := TSPX_TCHcarrierB_ho)				
21		[TSPC_DCS]				
22		(TCV_chdescr_arfcn := C_BCCHcarrierB_hod, TCV_tch_arfcn := TSPX_TCHcarrierB_hod)				

Detailed Comments: HandOver from TCH/H_FH sv1 of cell A to TCH/F_nonFH sv2 in cell B

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_14

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/F sv2, without frequency hopping on cell B towards a TCH/H, sv1, with no frequency hopping on cell A

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := 20, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellB, C_SCH_B, C_BCCH_B_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Bman, CellChDes_201_Bd, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnB, C_arfcnBd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellA_ho(C_E_neighbourdefault, C_BCCHcarrierA_ho, C_BCCHcarrierA_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Aman, CellChDes_201_Ad, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell A is 100 in advance of frame number in cell B.
6		+FullRateCh_B_1(C_Asynho, TCV_asscmd_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, NonFH on cellB
7		+ltree_arfcnB			
8		+Est_MT_CallNonFH(TimingAdv(TCV_T), C_Full_v2, TSPX_TCHHSubDef, TSPX_RANDDef)			To establish a MTcall with non hopping channel.
9	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
10		(TCV_n := 750, TCV_Cntref := TSPX_NoOfHoAccF, TCV_T := 20, TCV_ts := TSPX_TmSltA, TCV_Horf:= TSPX_HoRefD, TCV_tsc := TSPX_TscE, TCV_chtype := INT_TO_BIT((2+BIT_TO_INT(TSPX_TCHHSubA)),5))			Parameters initialization for Handover procedure.
11		+HalfRateCh_A_1(TSPX_TCHHSubA, C_Asynho, TCV_ts, TCV_tsc, ChMod_speech, FreqTCH_omit, FreqTCH_omit, TSPX_TCHcarrierA_ho, TSPX_TCHcarrierA_hod, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/H, sv1,nonFH on cellA
12		+ltree_arfcnA			
13		(TCV_HoCMD :=			

			HandOverCmd_nfh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), ChMod_speech_iei, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
14			(TCV_ch := TCV_chTch)			
15			(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
16			L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch1, TCV_HoCMD)		HO command
17			L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
18			+RR_hocomp1(TCV_n, TCV_chTch1)			
19			+ChanRel_end(TCV_ch)			
		ltree_arfcnA				
20		[TSPC_PGSM OR TSPC_EGSM]				
21		(TCV_chdescr_arfcn := C_arfcnA, TCV_tch_arfcn := TSPX_TCHcarrierA_ho)				
22		[TSPC_DCS]				
23		(TCV_chdescr_arfcn := C_arfcnAd_4, TCV_tch_arfcn := TSPX_TCHcarrierA_hod)				
		ltree_arfcnB				
24		[TSPC_PGSM OR TSPC_EGSM]				
25		(TCV_tch_arfcn := TSPX_TCHcarrierB_ho)				
26		[TSPC_DCS]				
27		(TCV_tch_arfcn := TSPX_TCHcarrierB_hod)				

Detailed Comments: HO from TCH/H_nonFH sv2 of cell B to TCH/F_nonFH sv1 in cell A

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_1_15

Group: GSM_MS_EFR_4220/EFR/

Purpose: To test that the MS shall correctly apply the handover procedure in the non-synchronized case when:
- a call is in progress and,
- handover is performed from a TCH/H sv1, with no frequency hopping on cell A towards a TCH/F, sv2, with no frequency hopping on cell B

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		(TCV_asscmd_ts := TSPX_TmSltA, TCV_T := 20, TCV_Pwrlvl_ho:= INT_TO_BIT(TSPX_PwrlvlA, 5))			
3		+EFR_BasicServiceMT(TSPX_MTBscSvcI, TSPX_MTChRateI)			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_201_Ad, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC, TSPX_k, 100)			frame number in cell B is 100 in advance of frame number in cell A.
6		+HalfRateCh_A_1(TSPX_TCHHSubA, C_Asynho, TSPX_TmSltA, TCV_tsc, ChMod_speech, FreqTCH_omit, FreqTCH_omit, TSPX_TCHcarrierA_ho, TSPX_TCHcarrierA_hod, TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/H, sv1, NonFH on cellA
7		+Est_MT_CallNonFH(TimingAdv(TCV_T), C_Half, TSPX_TCHHSubDef, TSPX_RANDDef)			To establish a MT call with non hopping channel.
8	body	(TCV_ch := TCV_chTch, TCV_chTch1:=TCV_ch)			
9		(TCV_n :=750, TCV_Cntref := TSPX_NoOfHoAccC, TCV_T := 20, TCV_ts := TSPX_TmSltB, TCV_Horf:= TSPX_HoRefE, TCV_tsc := TSPX_TscF, TCV_chtype := '00001'B)			Parameters initialization for Handover procedure.
10		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, C_cch_1Comb, C_BPM_3)			Set TCH/F, sv2, nonFH on cellB
11		+ltree_arfcnB			
12		(TCV_HoCMD := HandOverCmd_nfh(TCV_chtype, TCV_ts, TCV_tsc, C_NCC, C_BCC, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf,			

			TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
13			(TCV_ch := TCV_chTch)			
14			(TCV_Null := OM_SendNextOnHO(TCV_ch, TCV_Cntref))			
15			L!DL_DatRqHoCmd	HndOvSnd(TCV_chTch1, TCV_HoCMD)		HO command
16			L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17			+RR_hocomp1(TCV_n, TCV_chTch1)			
18			+ChanRel_end(TCV_ch)			
		ltree_arfcnB				
19		[TSPC_PGSM OR TSPC_EGSM]				
20		(TCV_chdescr_arfcn := C_BCCHcarrierB_ho, TCV_tch_arfcn := TSPX_TCHcarrierB_ho)				
21		[TSPC_DCS]				
22		(TCV_chdescr_arfcn := C_BCCHcarrierB_hod, TCV_tch_arfcn := TSPX_TCHcarrierB_hod)				

Detailed Comments: HO from TCH/H_nonFH sv1 of cell A to TCH/F_nonFH sv2 in cell B

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_2_1

Group: GSM_MS_EFR_4220/EFR/

Purpose: To verify that the MS correctly applies the handover procedure from SDCCH/4 with no frequency hopping to TCH/F using full rate speech version 2 with frequency hopping.

To verify that if during call establishment a layer 3 MM or CC message just sent by the MS is not Layer 2 acknowledge before the channel change caused by the HANDOVER COMMAND message, the MS shall send the Layer 3 message to the new cell, using the same value in the N(SD) field, after the handover procedure.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+EFR_BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		(TCV_ia_ts:=TSPX_TmSltNotZero, TCV_ts:='000'B, TCV_tsc:=TSPX_TscA, TCV_Cntref:=TSPX_NoOfHoAccA, TCV_Horf:=TSPX_HoRefA, TCV_Pwrlvl_ho:=INT_TO_BIT(TSPX_PwrlvlA,5), TCV_cchannel:=C_cch_1Comb, TCV_T := 20)			Parameters initialization for Immediate Assignment and Handover commands.
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCHb_hof3, FreqTCHb_hof3d, TimingAdv(TCV_T), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F FH on cellB
7		+ltree_arfcnA			Set the ARFCN for NonFH channel on cellA
8		+Est_MO_Call_init(C_CHSDCCH4_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(30), C_BCC, C_one, C_one)			Initiate MO call on cellA
9	body	+ltree_BuildHoCMD			
10		(TCV_chTch1 := TCV_ch)			
11		(TCV_Null := OM_SendNextOnHO(TCV_chTch, TCV_Cntref))			
12		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, TCV_HoCMD)		HO command
13		(TCV_ch:=TCV_chTch)			
14		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
15		+RR_hocomp1(500, TCV_chTch1)			Complete the HO procedure
16		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
17		+SendSeqNo_chk			Is N(SD) correct?

18		+ChanRel_end(TCV_ch)			
		ltree_arfcnA			
19		[TSPC_PGSM OR TSPC_EGSM]			
20		(TCV_tch_arfcn:=C_arfcnA)			
21		[TSPC_DCS]			
22		(TCV_tch_arfcn:=C_arfcnAd_4)			
		ltree_BuildHoCMD			
23		[TSPC_PGSM OR TSPC_EGSM]			
24		(TCV_HoCMD:= HandOverCmd_fh('00001'B, TCV_ts, TCV_tsc, C_NCC_3, C_BCC_0, C_BCCHcarrierB_ho, 15, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_06, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
25		[TSPC_DCS]			
26		(TCV_HoCMD:= HandOverCmd_fh('00001'B, TCV_ts, TCV_tsc, C_NCC_3, C_BCC_0, C_BCCHcarrierB_hod, 11, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_20_B5d, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			

Detailed Comments: HO from SDCCH/4, nonFH to TCH/F, sv2, FH

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_2_2

Group: GSM_MS_EFR_4220/EFR/

Purpose: To verify that the MS correctly applies the handover procedure from SDCCH/8 with frequency hopping to TCH/F using full rate speech version 2 without frequency hopping in the nonsynchronized case during call establishment.

To verify that if during call establishment a layer 3 MM or CC message just sent by the MS is not Layer 2 acknowledge before the channel change caused by the HANDOVER COMMAND message, the MS shall send the Layer 3 message to the new cell, using the same value in the N(SD) field, after the handover procedure.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+EFR_BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		(TCV_ia_ts:=TSPX_TmSltNotZero1, TCV_ts:='000'B, TCV_tsc:=TSPX_TscB, TCV_Cntref:=TSPX_NoOfHoAccB, TCV_Horf:=TSPX_HoRefB, TCV_Pwrlvl_ho:=INT_TO_BIT(TSPX_PwrlvlB,5), TCV_cchannel:=C_cch_1nonComb, TCV_T := 20)			Parameters initialization for Immediate Assignment and Handover commands.
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TCV_ia_ts, TCV_tsc, ChMod_sign, FreqSDCCHa_hof1, FreqSDCCHa_hof1d, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3)			Set SDCCH/8 FH on cell A
7		(TCV_ch :=TCV_chTch)			
8		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F nonFH on cellB
9		+Est_MO_Call_init(C_CHSDCCH8_FH, MoblAllc_281, MoblAllc_281d, TimingAdv(30), TCV_tsc, 3, 3)			Initiate MO call on cellA
10		+ltree_arfcnB			
11	body	(TCV_HoCMD:= HandOverCmd_nfh('00001'B, TCV_ts, TCV_tsc, C_NCC_3, C_BCC_0, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			

12		(TCV_chTch1 := TCV_ch)			
13		(TCV_Null := OM_SendNextOnHO(TCV_chTch, TCV_Cntref))			
14		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, TCV_HoCMD)		HO command
15		(TCV_ch:=TCV_chTch)			
16		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17		+RR_hocomp1(500, TCV_chTch1)			Complete the HO procedure
18		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
19		+SendSeqNo_chk			Is N(SD) correct?
20		+ChanRel_end(TCV_ch)			
	ltree_arfcnB				
21	[TSPC_PGSM OR TSPC_EGSM]				
22	(TCV_tch_arfcn:=TSPX_TCHcarrierB_ho, TCV_chdescr_arfcn:=C_BCCHcarrierB_ho)				
23	[TSPC_DCS]				
24	(TCV_tch_arfcn:=TSPX_TCHcarrierB_hod, TCV_chdescr_arfcn:=C_BCCHcarrierB_hod)				

Detailed Comments: HandOver from SDCCH/8, FH to TCH/F, sv2, nonFH

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_2_3

Group: GSM_MS_EFR_4220/EFR/

Purpose: To verify that the MS correctly applies the handover procedure from SDCCH/8 without frequency hopping to TCH/F using full rate speech version 2 with frequency hopping in the nonsynchronized case during call establishment.

To verify that if during call establishment a layer 3 MM or CC message just sent by the MS is not Layer 2 acknowledge before the channel change caused by the HANDOVER COMMAND message, the MS shall send the Layer 3 message to the new cell, using the same value in the N(SD) field, after the handover procedure.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+EFR_BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		+ltree_data_init			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+SDCCH8_A_1_nociph(TSPX_SDCCH8SubDef, C_Immass, TCV_ia_ts, TCV_tsc, ChMod_sign, FreqTCH(TSPX_TCHcarrierA_ho), FreqTCH(TSPX_TCHcarrierA_hod), TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3)			Set SDCCH/8 nonFH on cell A
7		(TCV_ch :=TCV_chTch)			
8		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCHb_hof6, FreqTCHb_hof6d, TimingAdv(TCV_T), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F FH on cellB
9		+ltree_arfcnA			Set the ARFCN for NonFH channel on cellA
10		+Est_MO_Call_init(C_CHSDCCH8_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(30), TCV_tsc, C_one, C_one)			Initiate MO call on cellA
11	body	+ltree_BuildHoCMD			
12		(TCV_chTch1 := TCV_ch)			
13		(TCV_Null := OM_SendNextOnHO(TCV_chTch, TCV_Cntref))			
14		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, TCV_HoCMD)		HO command
15		(TCV_ch:=TCV_chTch)			
16		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.

17			+RR_hocomp1(500, TCV_chTch1)			Complete the HO procedure
18			L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
19			+SendSeqNo_chk			Is N(SD) correct?
20			+ChanRel_end(TCV_ch)			
		ltree_data_init				
21		[TSPC_PGSM OR TSPC_EGSM]				
22		(TCV_ia_ts:='000'B, TCV_ts:='000'B, TCV_tsc:=TSPX_TscC, TCV_Cntref:=TSPX_NoOfHoAccC, TCV_Horf:=TSPX_HoRefC, TCV_Pwrlvl_ho:=INT_TO_BIT(TSPX_PwrlvlC,5), TCV_cchannel:=C_cch_1nonComb, TCV_T := 20)				
23		[TSPC_DCS]				
24		(TCV_ia_ts:='000'B, TCV_ts:=TSPX_TmSltNotZero, TCV_tsc:=TSPX_TscC, TCV_Cntref:=TSPX_NoOfHoAccC, TCV_Horf:=TSPX_HoRefC, TCV_Pwrlvl_ho:=INT_TO_BIT(TSPX_PwrlvlC,5), TCV_cchannel:=C_cch_1nonComb, TCV_T := 20)				
		ltree_arfcnA				
25		[TSPC_PGSM OR TSPC_EGSM]				
26		(TCV_tch_arfcn:=C_arfcnA)				
27		[TSPC_DCS]				
28		(TCV_tch_arfcn:=C_arfcnAd_4)				
		ltree_BuildHoCMD				
29		[TSPC_PGSM OR TSPC_EGSM]				
30		(TCV_HoCMD:= HandOverCmd_fh('00001'B, TCV_ts, TCV_tsc, C_NCC_3, C_BCC_0, C_BCCHcarrierB_ho, 2, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_omit, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_03, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				
31		[TSPC_DCS]				
32		(TCV_HoCMD:= HandOverCmd_fh('00001'B, TCV_ts, TCV_tsc, C_NCC_3, C_BCC_0, C_BCCHcarrierB_hod, 3, C_HSN_0, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_20_B10d, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				

Detailed Comments: HandOver from SDCCH/8, nonFH to TCH/F, sv2, FH

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_2_4

Group: GSM_MS_EFR_4220/EFR/

Purpose: To verify that the MS correctly applies the handover procedure from TCH/F without frequency hopping to TCH/F using full rate speech version 2 with frequency hopping in the nonsynchronized case during call establishment.

To verify that if during call establishment a layer 3 MM or CC message just sent by the MS is not Layer 2 acknowledge before the channel change caused by the HANDOVER COMMAND message, the MS shall send the Layer 3 message to the new cell, using the same value in the N(SD) field, after the handover procedure.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+EFR_BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		(TCV_ia_ts:=TSPX_TmSltNotZero, TCV_ts:=TSPX_TmSltNotZero1, TCV_tsc:=TSPX_TscA, TCV_Cntref:=TSPX_NoOfHoAccA, TCV_Horf:=TSPX_HoRefA, TCV_PwrlvI_ho:=INT_TO_BIT(TSPX_PwrlvlA,5), TCV_cchannel:=C_cch_1Comb, TCV_T := 20, TCV_ChMod.mode := C_ChMod_r)			Parameters initialization for Immediate Assignment and Handover commands.
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+FullRateCh_A_1_nociph(C_Immass, TCV_ia_ts, TCV_tsc, TCV_ChMod, FreqTCH(TSPX_TCHcarrierA_ho), FreqTCH(TSPX_TCHcarrierA_hod), TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F nonFH on cell A
7		(TCV_ch :=TCV_chTch)			
8		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCHb_hof1, FreqTCHb_hof1d, TimingAdv(TCV_T), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F FH on cellB
9		+ltree_arfcnA			Set the ARFCN for NonFH channel on cellA
10		+Est_MO_Call_init(C_CHTCHF_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(30), TCV_tsc, C_one, C_one)			Initiate MO call on cellA
11	body	+ltree_BuildHoCMD			
12		(TCV_chTch1 := TCV_ch)			
13		(TCV_Null := OM_SendNextOnHO(TCV_chTch, TCV_Cntref))			

14		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, TCV_HoCMD)		HO command
15		(TCV_ch:=TCV_chTch)			
16		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17		+RR_hocomp1(500, TCV_chTch1)			Complete the HO procedure
18		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
19		+SendSeqNo_chk			Is N(SD) correct?
20		+ChanRel_end(TCV_ch)			
	ltree_arfcnA				
21	[TSPC_PGSM OR TSPC_EGSM]				
22	(TCV_tch_arfcn:=66)				
23	[TSPC_DCS]				
24	(TCV_tch_arfcn:=754)				
	ltree_BuildHoCMD				
25	[TSPC_PGSM OR TSPC_EGSM]				
26	(TCV_HoCMD:=HandOverCmd_fh('00001'B, TCV_ts, TCV_tsc, C_NCC_1, C_BCC_5, C_BCCHcarrierB_ho, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_20_B3, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				
27	[TSPC_DCS]				
28	(TCV_HoCMD:=HandOverCmd_fh('00001'B, TCV_ts, TCV_tsc, C_NCC_1, C_BCC_5, C_BCCHcarrierB_hod, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_20_B3d, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				

Detailed Comments: HandOver from TCH/F, nonFH to TCH/F, sv2, FH

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_2_5

Group: GSM_MS_EFR_4220/EFR/

Purpose: To verify that the MS correctly applies the handover procedure from TCH/F without frequency hopping to TCH/F using full rate speech version 2 without frequency hopping in the nonsynchronized case during call establishment.

To verify that if during call establishment a layer 3 MM or CC message just sent by the MS is not Layer 2 acknowledge before the channel change caused by the HANDOVER COMMAND message, the MS shall send the Layer 3 message to the new cell, using the same value in the N(SD) field, after the handover procedure.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+EFR_BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		(TCV_ia_ts:=TSPX_TmSltNotZero, TCV_ts:=TSPX_TmSltNotZero1, TCV_tsc:=TSPX_TscB, TCV_Cntref:=TSPX_NoOfHoAccB, TCV_Horf:=TSPX_HoRefB, TCV_PwrlvI_ho:=INT_TO_BIT(TSPX_PwrlvlB,5), TCV_cchannel:=C_cch_1Comb, TCV_T := 20, TCV_ChMod.mode := C_ChMod_r)			Parameters initialization for Immediate Assignment and Handover commands.
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+FullRateCh_A_1_nociph(C_Immass, TCV_ia_ts, TCV_tsc, TCV_ChMod, FreqTCH(TSPX_TCHcarrierA_ho), FreqTCH(TSPX_TCHcarrierA_hod), TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F nonFH on cell A
7		(TCV_ch :=TCV_chTch)			
8		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F nonFH on cellB
9		+ltree_arfcnA			Set the ARFCN for NonFH channel on cellA
10		+Est_MO_Call_init(C_CHTCHF_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(30), TCV_tsc, C_one, C_one)			Initiate MO call on cellA
11		+ltree_arfcnB			
12	body	+ltree_BuildHoCMD			
13		(TCV_chTch1 := TCV_ch)			
14		(TCV_Null :=			

		OM_SendNextOnHO(TCV_chTch, TCV_Cntref))				
15		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, TCV_HoCMD)			HO command
16		(TCV_ch:=TCV_chTch)				
17		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))			See Note 1.
18		+RR_hocomp1(500, TCV_chTch1)				Complete the HO procedure
19		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)			
20		+SendSeqNo_chk				Is N(SD) correct?
21		+ChanRel_end(TCV_ch)				
	ltree_arfcnA					
22	[TSPC_PGSM OR TSPC_EGSM]					
23	(TCV_tch_arfcn:=66)					
24	[TSPC_DCS]					
25	(TCV_tch_arfcn:=791)					
	ltree_arfcnB					
26	[TSPC_PGSM OR TSPC_EGSM]					
27	(TCV_tch_arfcn:=31, TCV_chdescr_arfcn:=C_BCCHcarrierB_ho)					
28	[TSPC_DCS]					
29	(TCV_tch_arfcn:=749, TCV_chdescr_arfcn:=C_BCCHcarrierB_hod)					
	ltree_BuildHoCMD					
30	[TSPC_PGSM OR TSPC_EGSM]					
31	(TCV_HoCMD:=HandOverCmd_nfh('00001'B, TCV_ts, TCV_tsc, C_NCC_1, C_BCC_5, TCV_chdescr_arfcn, 31, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))					
32	[TSPC_DCS]					
33	(TCV_HoCMD:=HandOverCmd_nfh('00001'B, TCV_ts, TCV_tsc, C_NCC_1, C_BCC_5, TCV_chdescr_arfcn, 749, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))					

Detailed Comments: HandOver from TCH/F, nonFH to TCH/F, sv2, nonFH

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_2_6

Group: GSM_MS_EFR_4220/EFR/

Purpose: To verify that the MS correctly applies the handover procedure from TCH/F with frequency hopping to TCH/F using full rate speech version 2 with frequency hopping in the nonsynchronized case during call establishment.

To verify that if during call establishment a layer 3 MM or CC message just sent by the MS is not Layer 2 acknowledge before the channel change caused by the HANDOVER COMMAND message, the MS shall send the Layer 3 message to the new cell, using the same value in the N(SD) field, after the handover procedure.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+EFR_BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		+ltree_data_init			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Admin, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+FullRateCh_A_1_nociph(C_Immass, TCV_ia_ts, TCV_tsc, TCV_ChMod, FreqTCHa_hof_ind5, FreqTCHa_hof_ind5d, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F FH on cell A
7		(TCV_ch:=TCV_chTch)			Store TCH on cell A for call setup
8		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCHb_hof_ind8, FreqTCHb_hof_ind8d, TimingAdv(TCV_T), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F FH on cellB
9		+Est_MO_Call_init(C_CHTCHF_FH, MoblAllc_ind5A, MoblAllc_ind5Ad, TimingAdv(30), TCV_tsc, 5, 5)			Initiate MO call on cellA
10	body	+ltree_BuildHoCMD			
11		(TCV_chTch1 := TCV_ch)			
12		(TCV_Fn := OM_ComingFn(TCV_ch), TCV_Null:= OM_SendNextOn(TCV_ch, TCV_Fn), TCV_Strt := OC_StartTime(TCV_Fn,(C_StartingTimeHO + 100), 1))			Starting time is set to 1.1 s pluse 100 frames
13		(TCV_Null := OM_SendNextOnHO(TCV_chTch, TCV_Cntref))			
14		L!DL_DatRqHoCmd(DL_DatRqHoCmd.msg.strt:= TCV_Strt)	HndOvSnd(TCV_ch, TCV_HoCMD)		HO command

15		(TCV_ch:=TCV_chTch)			
16		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17		+RR_hocomp1(500, TCV_chTch1)			Complete the HO procedure
18		L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
19		+SendSeqNo_chk			Is N(SD) correct?
20		+ChanRel_end(TCV_ch)			
	ltree_data_init				
21	[TSPC_PGSM OR TSPC_EGSM]				
22	(TCV_ia_ts:=TSPX_TmSltNotZero, TCV_ts:='000'B, TCV_tsc:=TSPX_TscC, TCV_Cntref:=TSPX_NoOfHoAccC, TCV_Horf:=TSPX_HoRefC, TCV_Pwrlvl_ho:=INT_TO_BIT(TSPX_PwrlvlC,5), TCV_cchannel:=C_cch_1Comb, TCV_T := 20, TCV_ChMod.mode := C_ChMod_r)				Parameters initialization for Immediate Assignment and Handover commands.
23	[TSPC_DCS]				
24	(TCV_ia_ts:=TSPX_TmSltNotZero, TCV_ts:=TSPX_TmSltNotZero1, TCV_tsc:=TSPX_TscC, TCV_Cntref:=TSPX_NoOfHoAccC, TCV_Horf:=TSPX_HoRefC, TCV_Pwrlvl_ho:=INT_TO_BIT(TSPX_PwrlvlC,5), TCV_cchannel:=C_cch_1Comb, TCV_T := 20, TCV_ChMod.mode := C_ChMod_r)				
	ltree_BuildHoCMD				
25	[TSPC_PGSM OR TSPC_EGSM]				
26	(TCV_HoCMD:=HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscB, C_NCC_3, C_BCC_0, C_BCCHcarrierB_ho, 8, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_ind8B, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				
27	[TSPC_DCS]				
28	(TCV_HoCMD:=HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscB, C_NCC_3, C_BCC_0, C_BCCHcarrierB_hod, 8, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_ind8Bd, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				

Detailed Comments: HO from TCH/F, FH to TCH/F, sv2, FH

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_2_7

Group: GSM_MS_EFR_4220/EFR/

Purpose: To verify that the MS correctly applies the handover procedure from TCH/F with frequency hopping to TCH/F using full rate speech version 2 without frequency hopping in the nonsynchronized case during call establishment.

To verify that if during call establishment a layer 3 MM or CC message just sent by the MS is not Layer 2 acknowledge before the channel change caused by the HANDOVER COMMAND message, the MS shall send the Layer 3 message to the new cell, using the same value in the N(SD) field, after the handover procedure.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+EFR_BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		(TCV_ia_ts:=TSPX_TmSltNotZero, TCV_ts:='000'B, TCV_tsc:=TSPX_TscA, TCV_Cntref:=TSPX_NoOfHoAccA, TCV_Horf:=TSPX_HoRefA, TCV_Pwrlvl_ho:=INT_TO_BIT(TSPX_PwrlvlA,5), TCV_cchannel:=C_cch_1Comb, TCV_T := 20, TCV_ChMod.mode := C_ChMod_r)			Parameters initialization for Immediate Assignment and Handover commands.
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+FullRateCh_A_1_nociph(C_Immass, TCV_ia_ts, TCV_tsc, TCV_ChMod, FreqTCHa_hof_ind3, FreqTCHa_hof_ind3d, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F FH on cell A
7		(TCV_ch :=TCV_chTch)			
8		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F nonFH on cellB
9		+Est_MO_Call_init(C_CHTCHF_FH, MoblAllc_281, MoblAllc_ad281d, TimingAdv(30), TCV_tsc, 3, 3)			Initiate MO call on cellA
10		+ltree_arfcnB			
11		(TCV_chTch1 := TCV_ch)			
12		(TCV_HoCMD:= HandOverCmd_nfh('00001'B, TCV_ts, TCV_tsc, C_NCC_3, C_BCC_0, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, ChMod_omit, RelTmdDif_omit, TimingAdv_omit,			

			CphMod_omit))			
13			(TCV_Null := OM_SendNextOnHO(TCV_chTch, TCV_Cntref))			
14			L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, TCV_HoCMD)		HO command
15			(TCV_ch:=TCV_chTch)			
16			L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17			+RR_hocomp1(500, TCV_chTch1)			Complete the HO procedure
18			L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
19			+SendSeqNo_chk			Is N(SD) correct?
20			+ChanRel_end(TCV_ch)			
		ltree_arfcnB				
21		[TSPC_PGSM OR TSPC_EGSM]				
22		(TCV_tch_arfcn:=TSPX_TCHcarrierB_ho, TCV_chdescr_arfcn:=C_BCCHcarrierB_ho)				
23		[TSPC_DCS]				
24		(TCV_tch_arfcn:=TSPX_TCHcarrierB_hod, TCV_chdescr_arfcn:=C_BCCHcarrierB_hod)				

Detailed Comments: HandOver from TCH/F, FH to TCH/F, sv2, nonFH

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_2_8

Group: GSM_MS_EFR_4220/EFR/

Purpose: To verify that the MS correctly applies the handover procedure from TCH/H without frequency hopping to TCH/F using full rate speech version 2 with frequency hopping in the nonsynchronized case during call establishment.

To verify that if during call establishment a layer 3 MM or CC message just sent by the MS is not Layer 2 acknowledge before the channel change caused by the HANDOVER COMMAND message, the MS shall send the Layer 3 message to the new cell, using the same value in the N(SD) field, after the handover procedure.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+EFR_BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		(TCV_ia_ts:=TSPX_TmSltNotZero, TCV_ts:=TSPX_TmSltNotZero1, TCV_tsc:=TSPX_TscA, TCV_Cntref:=TSPX_NoOfHoAccA, TCV_Horf:=TSPX_HoRefA, TCV_PwrlvI_ho:=INT_TO_BIT(TSPX_PwrlvlA,5), TCV_cchannel:=C_cch_1Comb, TCV_T := 20)			Parameters initialization for Immediate Assignment and Handover commands.
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Immass, TCV_ia_ts, TCV_tsc, TCV_ChMod, FreqTCH_omit, FreqTCH_omit, TSPX_TCHcarrierA_ho, TSPX_TCHcarrierA_hod, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/H nonFH on cell A
7		(TCV_ch :=TCV_chTch)			
8		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCHb_hof1, FreqTCHb_hof1d, TimingAdv(TCV_T), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F FH on cellB
9		+ltree_arfcnA			Set the ARFCN for NonFH channel on cellA
10		+Est_MO_Call_init(C_CHTCHH_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(30), TCV_tsc, C_one, C_one)			Initiate MO call on cellA
11	body	+ltree_BuildHoCMD			
12		(TCV_chTch1 := TCV_ch)			
13		(TCV_Null := OM_SendNextOnHO(TCV_chTch, TCV_Cntref))			
14		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch,		HO command

				TCV_HoCMD)		
15			(TCV_ch:=TCV_chTch)			
16			L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17			+RR_hocomp1(500, TCV_chTch1)			Complete the HO procedure
18			L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
19			+SendSeqNo_chk			Is N(SD) correct?
20			+ChanRel_end(TCV_ch)			
		ltree_arfcnA				
21		[TSPC_PGSM OR TSPC_EGSM]				
22		(TCV_tch_arfcn:=66)				
23		[TSPC_DCS]				
24		(TCV_tch_arfcn:=754)				
		ltree_BuildHoCMD				
25		[TSPC_PGSM OR TSPC_EGSM]				
26		(TCV_HoCMD:=HandOverCmd_fh('00001'B, TCV_ts, TCV_tsc, C_NCC_1, C_BCC_5, C_BCCHcarrierB_ho, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_20_B3, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				
27		[TSPC_DCS]				
28		(TCV_HoCMD:=HandOverCmd_fh('00001'B, TCV_ts, TCV_tsc, C_NCC_1, C_BCC_5, C_BCCHcarrierB_hod, 12, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, Frql_20_B3d, Frql_omit, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				

Detailed Comments: HandOver from TCH/H, nonFH to TCH/F, sv2, FH

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_2_9

Group: GSM_MS_EFR_4220/EFR/

Purpose: To verify that the MS correctly applies the handover procedure from TCH/H without frequency hopping to TCH/F using full rate speech version 2 without frequency hopping in the nonsynchronized case during call establishment.

To verify that if during call establishment a layer 3 MM or CC message just sent by the MS is not Layer 2 acknowledge before the channel change caused by the HANDOVER COMMAND message, the MS shall send the Layer 3 message to the new cell, using the same value in the N(SD) field, after the handover procedure.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+EFR_BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		(TCV_ia_ts:=TSPX_TmSltNotZero, TCV_ts:=TSPX_TmSltNotZero1, TCV_tsc:=TSPX_TscB, TCV_Cntref:=TSPX_NoOfHoAccB, TCV_Horf:=TSPX_HoRefB, TCV_PwrlvI_ho:=INT_TO_BIT(TSPX_PwrlvlB,5), TCV_cchannel:=C_cch_1Comb, TCV_T := 20)			Parameters initialization for Immediate Assignment and Handover commands.
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Immass, TCV_ia_ts, TCV_tsc, TCV_ChMod, FreqTCH_omit, FreqTCH_omit, TSPX_TCHcarrierA_ho, TSPX_TCHcarrierA_hod, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/H nonFH on cell A
7		(TCV_ch :=TCV_chTch)			
8		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F nonFH on cellB
9		+ltree_arfcnA			Set the ARFCN for NonFH channel on cellA
10		+Est_MO_Call_init(C_CHTCHH_NonFH, MoblAllc_omit, MoblAllc_omit, TimingAdv(30), TCV_tsc, C_one, C_one)			
11		+ltree_arfcnB			
12	body	+ltree_BuildHoCMD			
13		(TCV_chTch1 := TCV_ch)			
14		(TCV_Fn := OM_ComingFn(TCV_ch),			starting time is set to 1.1 s pluse 100 frames

			TCV_Null:= OM_SendNextOn(TCV_ch, TCV_Fn), TCV_Strt := OC_StartTime(TCV_Fn,(C_StartingTimeHO + 100), 1))			
15			(TCV_Null := OM_SendNextOnHO(TCV_chTch, TCV_Cntref))			
16			L!DL_DatRqHoCmd(DL_DatRqHoCmd.msg. strt:= TCV_Strt)	HndOvSnd(TCV_ch, TCV_HoCMD)		
17			(TCV_ch:=TCV_chTch)			
18			L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
19			+RR_hocomp1(500, TCV_chTch1)			
20			L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
21			+SendSeqNo_chk			
22			+ChanRel_end(TCV_ch)			
		ltree_arfcnA				
23		[TSPC_PGSM OR TSPC_EGSM]				
24		(TCV_tch_arfcn:=66)				
25		[TSPC_DCS]				
26		(TCV_tch_arfcn:=791)				
		ltree_arfcnB				
27		[TSPC_PGSM OR TSPC_EGSM]				
28		(TCV_tch_arfcn:=31, TCV_chdescr_arfcn:=C_BCCHcarrierB_ho)				
29		[TSPC_DCS]				
30		(TCV_tch_arfcn:=749, TCV_chdescr_arfcn:=C_BCCHcarrierB_hod)				
		ltree_BuildHoCMD				
31		[TSPC_PGSM OR TSPC_EGSM]				
32		(TCV_HoCMD:=HandOverCmd_nfh('00001'B, TCV_ts, TCV_tsc, C_NCC_1, C_BCC_5, TCV_chdescr_arfcn, 31, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				
33		[TSPC_DCS]				
34		(TCV_HoCMD:=HandOverCmd_nfh('00001'B, TCV_ts, TCV_tsc, C_NCC_1, C_BCC_5, TCV_chdescr_arfcn, 749, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, ChMod_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))				

Detailed Comments: HandOver from TCH/H, nonFH to TCH/F, sv2, nonFH

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_2_10

Group: GSM_MS_EFR_4220/EFR/

Purpose: To verify that the MS correctly applies the handover procedure from TCH/H with frequency hopping to TCH/F using full rate speech version 2 with frequency hopping in the nonsynchronized case during call establishment.

To verify that if during call establishment a layer 3 MM or CC message just sent by the MS is not Layer 2 acknowledge before the channel change caused by the HANDOVER COMMAND message, the MS shall send the Layer 3 message to the new cell, using the same value in the N(SD) field, after the handover procedure.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+EFR_BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		+ltree_data_init			
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+HalfRateCh_A_1_nociph(TSPX_TCHHSubDef, C_Immass, TCV_ia_ts, TCV_tsc, TCV_ChMod, FreqTCHa_hof_ind5, FreqTCHa_hof_ind5d, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/H FH on cell A
7		(TCV_ch :=TCV_chTch)			
8		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCHb_hof_ind8, FreqTCHb_hof_ind8d, TimingAdv(TCV_T), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F FH on cellB
9		+Est_MO_Call_init(C_CHTCHH_FH, MoblAllc_ind5A, MoblAllc_ind5Ad, TimingAdv(30), TCV_tsc, 5, 5)			
10	body	+ltree_BuildHoCMD			
11		(TCV_chTch1 := TCV_ch)			
12		(TCV_Null := OM_SendNextOnHO(TCV_chTch, TCV_Cntref))			
13		L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, TCV_HoCMD)		HO command
14		(TCV_ch:=TCV_chTch)			
15		L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
16		+RR_hocomp1(500, TCV_chTch1)			
17		L?DL_DatInSetup (TCV_Mt1 :=	SetupRcv(SetupInd_01)		

		DL_DatInSetup.msg.mt)			
18		+SendSeqNo_chk			
19		+ChanRel_end(TCV_ch)			
		ltree_data_init			
20		[TSPC_PGSM OR TSPC_EGSM]			
21		(TCV_ia_ts:=TSPX_TmSltNotZero, TCV_ts:='000'B, TCV_tsc:=TSPX_TscC, TCV_Cntref:=TSPX_NoOfHoAccC, TCV_Horf:=TSPX_HoRefC, TCV_Pwrlvl_ho:=INT_TO_BIT(TSPX_PwrlvlC,5), TCV_cchannel:=C_cch_1Comb, TCV_T := 20)			Parameters initialization for Immediate Assignment and Handover commands.
22		[TSPC_DCS]			
23		(TCV_ia_ts:=TSPX_TmSltNotZero, TCV_ts:=TSPX_TmSltNotZero1, TCV_tsc:=TSPX_TscC, TCV_Cntref:=TSPX_NoOfHoAccC, TCV_Horf:=TSPX_HoRefC, TCV_Pwrlvl_ho:=INT_TO_BIT(TSPX_PwrlvlC,5), TCV_cchannel:=C_cch_1Comb, TCV_T := 20)			
		ltree_BuildHoCMD			
24		[TSPC_PGSM OR TSPC_EGSM]			
25		(TCV_HoCMD:=HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscB, C_NCC_3, C_BCC_0, C_BCCHcarrierB_ho, 8, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_ind8B, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			
26		[TSPC_DCS]			
27		(TCV_HoCMD:=HandOverCmd_fh('00001'B, TCV_ts, TSPX_TscB, C_NCC_3, C_BCC_0, C_BCCHcarrierB_hod, 8, TSPX_HSN, TCV_Horf, TCV_Pwrlvl_ho, Synchi(C_not_report_otd, C_non_synchronised), Frql_omit, Frql_ind8Bd, CellChDes_Omit, ChMod_omit, Freqchseq_omit, MoblAllc_omit, RelTmdDif_omit, TimingAdv_omit, CphMod_omit))			

Detailed Comments: HandOver from TCH/H, FH to TCH/F, sv2, FH

Notes:

1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_2_2_11

Group: GSM_MS_EFR_4220/EFR/

Purpose: To verify that the MS correctly applies the handover procedure from TCH/H with frequency hopping to TCH/F using full rate speech version 2 with no frequency hopping in the nonsynchronized case during call establishment.

To verify that if during call establishment a layer 3 MM or CC message just sent by the MS is not Layer 2 acknowledge before the channel change caused by the HANDOVER COMMAND message, the MS shall send the Layer 3 message to the new cell, using the same value in the N(SD) field, after the handover procedure.

Default: OtherEventsFail_01

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		+EFR_BasicServiceMO(TSPX_MOBscSvcA, TSPX_MOChRateA)			
3		(TCV_ia_ts:=TSPX_TmSltNotZero, TCV_ts:='000'B, TCV_tsc:=TSPX_TscA, TCV_Cntref:=TSPX_NoOfHoAccA, TCV_Horf:=TSPX_HoRefA, TCV_Pwrlvl_ho:=INT_TO_BIT(TSPX_PwrlvlA,5), TCV_cchannel:=C_cch_1Comb, TCV_T := 20)			Parameters initialization for Immediate Assignment and Handover commands.
4		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_20_Aman, CellChDes_202_Adman, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_Omit, BcchFreqLst_48, BcchFreqLst_Omit, C_noRestablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd_4, TSPX_IMSI, C_Test_fh, C_NCCP_10)			
5		+StartCellB_ho(C_E_neighbourdefault, C_BCCHcarrierB_ho, C_BCCHcarrierB_hod, C_Immass, TCV_slot, TCV_tsc, TimingAdv(TCV_T), C_TxInt_5, C_Max_7, C_NECI_0, C_ATT_0, C_BABR_0, TCV_cchannel, C_BPM_3, C_T3212_0, C_MCC_1, C_PLMN_1, C_LAC_2, CellChDes_20_Bman, CellChDes_201_Bd, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC_0, C_NCC_3, TSPX_k, 0)			The timeslots of Cells A and B are not coincident at the antenna connector.
6		+HalfRateCh_A_1_nociph(TSPX_TCHHSubDef, C_Immass, TCV_ia_ts, TCV_tsc, TCV_ChMod, FreqTCHa_hof_ind3, FreqTCHa_hof_ind3d, TimingAdv(30), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/H FH on cell A
7		(TCV_ch :=TCV_chTch)			
8		+FullRateCh_B_1(C_Asynho, TCV_ts, TCV_tsc, ChMod_EFR2, FreqTCH(TSPX_TCHcarrierB_ho), FreqTCH(TSPX_TCHcarrierB_hod), TimingAdv(TCV_T), C_BABR_0, TCV_cchannel, C_BPM_3)			Set TCH/F nonFH on cellB
9		+Est_MO_Call_init(C_CHTCHH_FH, MoblAllc_281, MoblAllc_ad281d, TimingAdv(30), TCV_tsc, 3, 3)			Initiate MO call on cellA
10	body	+ltree_arfcnB			
11		(TCV_chTch1 := TCV_ch)			
12		(TCV_HoCMD:= HandOverCmd_nfh('00001'B, TCV_ts, TCV_tsc, C_NCC_3, C_BCC_0, TCV_chdescr_arfcn, TCV_tch_arfcn, TCV_Horf, TCV_Pwrlvl_ho, Synchi_omit, ChMod_omit, RelTmdDif_omit, TimingAdv_omit,			

			CphMod_omit))			
13			(TCV_Null := OM_SendNextOnHO(TCV_chTch, TCV_Cntref))			
14			L!DL_DatRqHoCmd	HndOvSnd(TCV_ch, TCV_HoCMD)		HO command
15			(TCV_ch:=TCV_chTch)			
16			L!DL_DatRqPhyinfo	PhyInfo_01(TCV_ch, TimingAdv(TCV_T))		See Note 1.
17			+RR_hocomp1(500,TCV_chTch1)			Complete the HO procedure
18			L?DL_DatInSetup (TCV_Mt1 := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
19			+SendSeqNo_chk			Is N(SD) correct?
20			+ChanRel_end(TCV_ch)			
		ltree_arfcnB				
21		[TSPC_PGSM OR TSPC_EGSM]				
22		(TCV_tch_arfcn:=TSPX_TCHcarrierB_ho, TCV_chdescr_arfcn:=C_BCCHcarrierB_ho)				
23		[TSPC_DCS]				
24		(TCV_tch_arfcn:=TSPX_TCHcarrierB_hod, TCV_chdescr_arfcn:=C_BCCHcarrierB_hod)				

Detailed Comments: HO from TCH/H, FH to TCH/F, sv2, nonFH

Notes:
1. PhyInfo is dequeued from the sending buffer, but held through OM_SendNextOnHO at the lower layers of the tester for sending till having received TCV_Cntref access bursts.

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_3

Group: GSM_MS_EFR_4220/EFR/

Purpose: 1) To verify that the MS indicates and includes in the mobile originating SETUP for speech call all the speech versions that it supports.

2) To verify that subsequently after receipt of an IMMEDIATE ASSIGNMENT message allocating an SDCCH, after completion of establishment of the main signalling link, after having sent a CM SERVICE REQUEST message, after having successfully performed authentication and cipher mode setting

procedures, after having sent a SETUP message, after having received a CALL PROCEEDING message followed by an ALERTING message and an ASSIGNMENT COMMAND message allocating either speech full rate version 1 TCH or speech full rate version 2 TCH or speech half rate version 1 TCH (for an MS also supporting half rate), the MS sends an ASSIGNMENT COMPLETE message.

3) To verify that subsequently, after the suite of actions specified in test purposes 1 and 2, the MS after receiving a CONNECT message returns a CONNECT ACKNOWLEDGE message.

4) To verify that after the suite of actions specified in test purposes 1 and 2, the MS after receiving a CONNECT message attaches the user connection to the radio path. (This is checked by verifying that there is a point in time after transmission of the first L2 frame containing the (complete) CONNECT message, where the MS is sending appropriate speech or data frames whenever it doesn't have to transmit or acknowledge an I frame on layer 2 of the FACCH.)

These test purposes are tested for all rates supported by the MS (full rate versions 1 and 2, plus optional half rate version 1 support).

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2		[TSPC_Serv_TS11]			Telephony supported
3		+EFR_BasicServiceMO(C_Telephony, C_Full_v2)			Init Full Rate v2.
4		+local_testfullrate			M = 1.
5		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
6		+EFR_BasicServiceMO(C_Telephony, C_Full)			Init Full Rate v1.
7		+local_testfullrate			M = 2.
8		[TSPC_DualRate]			Half rate supported.
9		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
10		+EFR_BasicServiceMO(C_Telephony, C_Half)			Init Half Rate v1.
11		+local_testhalfrate			M = 3.
12		[NOT TSPC_DualRate]			
13	E_0023	[C_Yes]		I	IXIT error
		local_testfullrate			
14		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNA, TSPX_RANDA, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
15		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, TCV_ChMod, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
16		+local_initcall			

		local_testhalfrate			
17		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TCV_ChMod, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			7.
18		+local_initcall			
		local_initcall			
19		(TCV_Null := OO_DialCalledNum())			4.
20		[TSPC_CalledNumDisp]			
21		(TCV_Res := OO_CalledNumCHK())			5.
22	E_0024	[NOT TCV_Res]		(F)	
23	E_0025	[TCV_Res]		(P)	
24		+InitCall(TCV_Service)			6.
25		+localtree			
26		[NOT TSPC_CalledNumDisp]			
27		+InitCall(TCV_Service)			
28		+localtree			
		localtree			
29	E_0026	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_04)	(P)	
30		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
31		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
32		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_04)		
33		ACTIVATE(OtherEventsFail)			Restore Normal default
34		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNA, TSPX_RANDA))		
35		L?DL_DatInAuthRes (TCV_Sres := DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
36		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, TSPX_RANDA))			
37	E_0027	[NOT TCV_Res]		(F)	
38		+PostMainLinkRel(TCV_ch)			
39		[TCV_Res]			
40		+localtree1			
		localtree1			
41		+Ciphering_on(TCV_ch)			
42		+EFR_RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			
43	E_0028	L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)	(P)	
44		L!DL_DatRqAlert	AlertSnd(TCV_ch, Alerting_01(TCV_TI))		
45		[TSPC_AlertInd]			
46		(TCV_Res := OO_AltIndCHK())			
47	E_0029	[NOT TCV_Res]		(F)	
48		+PostMainLinkRel(TCV_ch)			
49		[TCV_Res]			
50		+localtree2			
51		[NOT TSPC_AlertInd]			
52		+localtree2			

		localtree2			
53		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
54		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
55		+localtree3			
		localtree3			
56		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
57	E_0030	L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))	(P)	
58		(TCV_Res := OO_TCHThroConnCHK())			
59	E_0031	[NOT TCV_Res]		(F)	
60		+PostMainLinkRel(TCV_chTch)			
61	E_0032	[TCV_Res]		(P)	
62		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
63	E_0033	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	
64		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
65		+PostMainLinkRel(TCV_chTch)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_4

Group: GSM_MS_EFR_4220/EFR/

Purpose: 1) In acceptance to a SETUP message indicating speech, the MS shall indicate and include in the CALL CONFIRMED message all the speech versions that it supports.

2) To verify that upon receipt of the ASSIGNMENT COMMAND message specifying either speech full rate version 1 or speech full rate version 2 or speech half rate verson 1 (for an MS also supporting half rate), the mobile station continues a mobile terminating call establishment with early assignment of traffic channel

a) by replying to the ASSIGNMENT COMMAND with an ASSIGNMENT COMPLETE message and,

b) by continuing the call establishment by through connecting TCH in both directions if it supports immediate connect, or by sending an ALERTING message otherwise.

3) To verfiy that for speech calls, themobile station shall attach the user connection at latest when sending the connect message, except if there is no compatible radio resource available at this time. In this case the attachment shall be delayed until such a resource becomes available.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(300)			
2	body	[TSPC_Serv_TS11]			
3		+EFR_BasicServiceMT(C_Telephony, C_Full_v2)			
4		+local_testfullrate			M = 1.
5		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
6		+EFR_BasicServiceMT(C_Telephony, C_Full)			
7		+local_testfullrate			M = 2.
8		[TSPC_DualRate]			
9		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
10		+EFR_BasicServiceMT(C_Telephony, C_Half)			
11		+local_testhalfrate			M = 3.
12		[NOT TSPC_DualRate]			
13	E_0034	[C_Yes]		I	
		local_testfullrate			
14		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNA, TSPX_RANDA, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			
15		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, TCV_ChMod, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
16		+localtree(C_Full)			
		local_testhalfrate			
17		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TCV_ChMod, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			4.
18		+localtree(C_Half)			

		localtree(rate:IA5String)			
19		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
20		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
21		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
22		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
23		L?DL_EstInPgRes	PagingRes(PagingRes_03(TSPX_CKSNDef))		
24		ACTIVATE(OtherEventsFail)			Restore Normal default
25		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNA, TSPX_RANDA))		
26		L?DL_DatInAuthRes(TCV_Sres := DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
27		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, TSPX_RANDA))			
28	E_0035	[NOT TCV_Res]		(F)	
29		+PostMainLinkRel(TCV_ch)			
30	E_0036	[TCV_Res]		(P)	5.
31		+localtree1(rate)			
		localtree1(rate:IA5String)			
32		+Ciphering_on(TCV_ch)			
33		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		6.
34		+ltree_RcvCallConfirmed			
35		+EFR_ChkRcvSpchPrefs			check all spchprefs are present in bcap
36		[TSPX_Immconn]			
37		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
38		+asstrafficch(rate)			7.
39		+localtree3			
40		[NOT TSPX_Immconn]			
41		+asstrafficch(rate)			7.
42		L?DL_DatInAlert	AlertRcv(AlertingInd_01(TI_01))		
43		[TSPC_AlertInd]			
44		(TCV_Res := OO_AltIndCHK())			
45	E_0037	[NOT TCV_Res]		(F)	8.
46		+PostMainLinkRel(TCV_chTch)			
47	E_0038	[TCV_Res]		(P)	
48		(TCV_Null := OO_HookOff())			9.
49		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
50		+localtree3			
51		[NOT TSPC_AlertInd]			
52		(TCV_Null := OO_HookOff())			9.
53		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
54		+localtree3			
		localtree3			

55		[TCV_Setup_mt.bcap1.itc = '000'B]			Speech Call
56		(TCV_Res := OO_TCHThroConnCHK())			
57	E_0039	[NOT TCV_Res]		(F)	
58		+PostMainLinkRel(TCV_chTch)			
59	E_0040	[TCV_Res]		(P)	
60		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
61		+localtree5			
62		[TCV_Setup_mt.bcap1.itc <> '000'B]			Data Call
63		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
64		(TCV_Res := OO_TCHThroConnCHK())			
65	E_0041	[NOT TCV_Res]		(F)	
66		+PostMainLinkRel(TCV_chTch)			
67	E_0042	[TCV_Res]		(P)	
68		+localtree5			
		localtree5			
69		(TCV_Null := OO_TermCall())			
70		L?DL_DatInDisc	DiscRcv(TCV_chTch, DisconnR(TI_01, Cause_Def))		
71		L!DL_DatRqRel	ReleaseSnd(TCV_chTch, Release_08(TI_02))		
72		L?DL_DatInRelCmp	RelComRcv(ReleaseCmp_03(TI_01))		
73		+PostMainLinkRel(TCV_chTch)			
		ltree_RcvCallConfirmed			
74		[TSPX_EFR_SpchPrefsNo = 2]			
75		L?DL_DatInCallCo(TCV_CallCfm := DL_DatInCallCo.msg, TCV_EFR_svi1 := DL_DatInCallCo.msg.bcap1.oct_3a_etc_1.svi, TCV_EFR_svi2 := DL_DatInCallCo.msg.bcap1.oct_3a_etc_2.svi)	CallCfm(CallConfirm_04(TI_01, TCV_BcapMO1))		
76		[TSPX_EFR_SpchPrefsNo = 3]			
77		L?DL_DatInCallCo(TCV_CallCfm := DL_DatInCallCo.msg, TCV_EFR_svi1 := DL_DatInCallCo.msg.bcap1.oct_3a_etc_1.svi, TCV_EFR_svi2 := DL_DatInCallCo.msg.bcap1.oct_3a_etc_2.svi, TCV_EFR_svi3 := DL_DatInCallCo.msg.bcap1.oct_3a_etc_3.svi)	CallCfm(CallConfirm_04(TI_01, TCV_BcapMO1))		
78	E_0043	[C_Yes]		I	IXIT ERROR
		asstrafficch(rate:IA5String)			
79		(TCV_Res := FALSE)			
80		+AssCmdGen(TCV_cellid, rate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
81		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			

Detailed Comments:

Test Case Dynamic Behaviour

Test Case Name: TC_26_12_5

Group: GSM_MS_EFR_4220/EFR/

Purpose: 1) To verify that an MS supporting speech in the MM state "idle, updated", when made to call the number 112, sends a CHANNEL REQUEST message with establishment cause "emergency call".

2) To verify that after assignment of a dedicated channel the first layer message sent by the MS on the assigned dedicated channel is a CM SERVICE REQUEST message specifying the correct CKSN and TMSI, with CM Service Type "emergency call establishment" .

3) To verify that authentication and cipher mode setting are performed successfully.

4) To verify that after cipher mode setting acceptance by the SS, the MS sends an EMERGENCY SETUP message.

5) The EFR mobile station shall accept channel assignment to a TCH full rate speech version 1 or 2 and if it supports half rate, in addition it shall accept channel assignment to a TCH half rate speech verson 1.

6) After receipt of a CONNECT ACKNOWLEDGE message during correct establishment of the emergency call, the TCH shall be through connected in both directions if an appropriate TCH is available.

7) To verify that the call is cleared correctly.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_Serv_TS11]			Telephony supported
2		+EFR_BasicServiceMO(C_EmgCall, C_Full)			Init Full Rate v1.
3		+local_make_ecall			M = 1.
4		[TSPC_EFR_EmgCallBcap]			BCAP present
5		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
6		+EFR_BasicServiceMO(C_EmgCall, C_Full_v2)			Init Full Rate v2.
7		+local_make_ecall			M = 2.
8		[TSPC_DualRate]			Half rate
9		(TCV_Null := OM_CphMd(TCV_ch, CphMod_02, TCV_CphKey))			
10		+EFR_BasicServiceMO(C_EmgCall, C_Half)			Init Half Rate v1.
11		+local_make_ecall			M = 3.
12		[NOT TSPC_DualRate]			
13		[NOT TSPC_EFR_EmgCallBcap]			
14	E_0044	[C_Yes]		I	IXIT ERROR
		local_make_ecall			
15		START T_guard(300)			
16		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1, CellOpt_01, CellChDes_Omit, CellChDes_Omit, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_45, BcchFreqLst_02, BcchFreqLst_03, BcchFreqLst_04, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNA, TSPX_RANDA, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_nfh, C_NCCP_2)			1.
17	body	+InitCall(TCV_Service)			
18	E_0045	L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_16)	(P)	2.
19		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
20		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn,		

			TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
21	E_0046	L?DL_EstInCmsRq	CMSerReq(CMServiceReq_06(TSPX_CKSNA))	(P)	3.
22		ACTIVATE(OtherEventsFail)			Restore Normal default
23		L!DL_DatRqAuthRq	AuthReq(TCV_ch, AuthRequest(TSPX_CKSNA, TSPX_RANDA))		
24		L?DL_DatInAuthRes(TCV_Sres := DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
25		+local_continue			
		local_continue			
26		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, TSPX_RANDA))			
27	E_0047	[NOT TCV_Res]		(F)	4.
28		+PostMainLinkRel(TCV_ch)			
29	E_0048	[TCV_Res]		(P)	5.
30		+Ciphering_on(TCV_ch)			
31		+EFR_RcvSetupOrEsetup(TCV_Setup_mo, TCV_Esetup, TCV_Ecall)			6.
32		+local_preferred_rate			
33		L!DL_DatRqCallProc	CallProc(TCV_ch, TCV_CallProc)		
34		L!DL_DatRqAlert	AlertSnd(TCV_ch, Alerting_01(TCV_TI))		
35		+AssCmdGen(TCV_cellid, TCV_ChRate, TSPX_TmSltDef, TSPX_TscDef, TSPX_TCHHSubDef)			
36		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
37		+local_continue1			
		local_continue1			
38		L!DL_DatRqConn	ConnSnd(TCV_chTch, Connect_02(TCV_TI))		
39	E_0049	L?DL_DatInConnAck	ConnAckRcv(ConnectAck_02(TCV_TI0))	(P)	
40		(TCV_Res := OO_TCHThroConnCHK())			
41	E_0050	[NOT TCV_Res]		(F)	7.
42		+PostMainLinkRel(TCV_chTch)			
43	E_0051	[TCV_Res]		(P)	8.
44		L!DL_DatRqDisc	DiscSnd(TCV_chTch, DisconnS(TCV_TI, Cause_01, ProgInd_omit, UuInfo_omit))		
45	E_0052	L?DL_DatInRel	ReleaseRcv(Release_10(TCV_TI0))	(P)	
46		L!DL_DatRqRelCmp	RelComSnd(TCV_chTch, ReleaseCmp_08(TCV_TI))		
47		+PostMainLinkRel(TCV_chTch)			
		local_preferred_rate			
48		(TCV_Rchr := TCV_Setup_mo1.bcap1.rchr)			

49		[TCV_ChRate = C_Full]			9.
50		+FullRateCh_A_1(C_Ass, TSPX_TmSltDef, TSPX_TscDef, TCV_ChMod, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd,TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
51		[(TCV_ChRate = C_Half) AND ((TCV_Rchr = '10'B) OR (TCV_Rchr = '11'B))]			10.
52		+HalfRateCh_A_1(TSPX_TCHHSubDef, C_Ass, TSPX_TmSltDef, TSPX_TscDef, TCV_ChMod, FreqTCH_omit, FreqTCH_omit, C_arfcn_tchA, C_arfcn_tchAd, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			

Detailed Comments:

1. To setup a physical channels as BCCH CCCH and SDCCH4 combined channel.
2. CHANNEL REQUEST with "emergency call establishment" cause received.
3. CM SERVICE REQUEST with "emergency call establishment" service type correct TMSI and CKSN received.
4. Authentication failed.
5. Authentication passed.
6. EMERGENCY SETUP with BC or without BC received.
7. The TCH channel is not through connected, fail.
8. The TCH channel is through connected.
9. dual rate / full rate preferred" case. setup a physical channel as full rate traffic channel and the preferred full rate channel is to be assigned.
10. dual rate / half rate preferred" case. setup a physical channel as half rate traffic channel and the preferred half rate channel is to be assigned.

Test Step Library

Test Step Group management

Test Step Group ChConfig

Test Step Dynamic Behaviour

Test Step Name: Config_CBMS_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH, SDCCH4 and CBCH onto the physical channel which represents cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, C_FCCH_A, C_SCH_A, C_BCCH_A_1, C_PCH_A_1, C_AGCH_A_1, C_RACH_A_1, C_SDCCH4_A, C_SACCHC4_A, C_CBCH_A))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_A, C_SDCCH40_A, C_SDCCH41_A, C_SDCCH42_A, C_SDCCH43_A, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_A, C_SACCHC40_A, C_SACCHC41_A, C_SACCHC42_A, C_SACCHC43_A, "dummy", "dummy", "dummy", "dummy"))			

Detailed Comments:

Test Step Dynamic Behaviour					
Test Step Name:	Config_FACCHF_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set one physical channel used as TCHF_ACCH's of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_FACCHF_A_1, C_SACCHF_A_1, "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_FACCHF_A_2(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set one physical channel used as TCHF_ACCH's for instance 2 of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_FACCHF_A_2, C_SACCHF_A_2, "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_FACCHF_B_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set one physical channel used as TCHF_ACCH's of cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_FACCHF_B_1, C_SACCHF_B_1, "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_FACCHH_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set one physical channel used as TCHH_ACCH's of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_FACCHH_A_1, C_SACCHH_A_1, "dummy"))			
2		(TCV_Null := OM_Assoc(C_FACCHH_A_1, C_FACCHH0_A_1, C_FACCHH1_A_1, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_SACCHH_A_1, C_SACCHH0_A_1, C_SACCHH1_A_1, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_FACCHH_A_2(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set one physical channel used as TCHH_ACCH's of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_FACCHH_A_2, C_SACCHH_A_2, "dummy"))			
2		(TCV_Null := OM_Assoc(C_FACCHH_A_2, C_FACCHH0_A_2, C_FACCHH1_A_2, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_SACCHH_A_2, C_SACCHH0_A_2, C_SACCHH1_A_2, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_FACCHH_B_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set one physical channel used as TCHH_ACCH's of cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_FACCHH_B_1, C_SACCHH_B_1, "dummy"))			
2		(TCV_Null := OM_Assoc(C_FACCHH_B_1, C_FACCHH0_B_1, C_FACCHH1_B_1, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_SACCHH_B_1, C_SACCHH0_B_1, C_SACCHH1_B_1, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, C_FCCH_A, C_SCH_A, C_BCCH_A_1, C_PCH_A_1, C_AGCH_A_1, C_RACH_A_1, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_A_ho_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3; td, fn:INTEGER)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell A with controlable td and fn.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, td, fn, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, C_FCCH_A, C_SCH_A, C_BCCH_A_1, C_PCH_A_1, C_AGCH_A_1, C_RACH_A_1, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_A_2(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set a physical channel and map the second BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", C_BCCH_A_2, C_PCH_A_2, C_AGCH_A_2, C_RACH_A_2, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_A_3(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set a physical channel and map the third BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", C_BCCH_A_3, C_PCH_A_3, C_AGCH_A_3, C_RACH_A_3, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_A_4(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set a physical channel and map the fourth BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", C_BCCH_A_4, C_PCH_A_4, C_AGCH_A_4, C_RACH_A_4, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_B_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, C_FCCH_B, C_SCH_B, C_BCCH_B_1, C_PCH_B_1, C_AGCH_B_1, C_RACH_B_1, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: Config_BCCH_CCCH_B_ho_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3; td, fn:INTEGER)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell B with controlable td and fn.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, td, fn, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, C_FCCH_B, C_SCH_B, C_BCCH_B_1, C_PCH_B_1, C_AGCH_B_1, C_RACH_B_1, "dummy", "dummy", "dummy"))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Config_SDCCH4_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, C_FCCH_A, C_SCH_A, C_BCCH_A_1, C_PCH_A_1, C_AGCH_A_1, C_RACH_A_1, C_SDCCH4_A, C_SACCHC4_A, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_A, C_SDCCH40_A, C_SDCCH41_A, C_SDCCH42_A, C_SDCCH43_A, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_A, C_SACCHC40_A, C_SACCHC41_A, C_SACCHC42_A, C_SACCHC43_A, "dummy", "dummy", "dummy", "dummy"))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Config_SDCCH4_A_ho_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3; td, fn:INTEGER)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A with controlable ta and fn.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, td, fn, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, C_FCCH_A, C_SCH_A, C_BCCH_A_1, C_PCH_A_1, C_AGCH_A_1, C_RACH_A_1, C_SDCCH4_A, C_SACCHC4_A, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_A, C_SDCCH40_A, C_SDCCH41_A, C_SDCCH42_A, C_SDCCH43_A, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_A, C_SACCHC40_A, C_SACCHC41_A, C_SACCHC42_A, C_SACCHC43_A, "dummy", "dummy", "dummy", "dummy"))			

Detailed Comments:

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_B_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, C_FCCH_B, C_SCH_B, C_BCCH_B_1, C_PCH_B_1, C_AGCH_B_1, C_RACH_B_1, C_SDCCH4_B, C_SACCHC4_B, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_B, C_SDCCH40_B, C_SDCCH41_B, C_SDCCH42_B, C_SDCCH43_B, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_B, C_SACCHC40_B, C_SACCHC41_B, C_SACCHC42_B, C_SACCHC43_B, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_B_ho_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3; td, fn:INTEGER)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B with controlable ta and fn.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, td, fn, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, C_FCCH_B, C_SCH_B, C_BCCH_B_1, C_PCH_B_1, C_AGCH_B_1, C_RACH_B_1, C_SDCCH4_B, C_SACCHC4_B, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_B, C_SDCCH40_B, C_SDCCH41_B, C_SDCCH42_B, C_SDCCH43_B, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_B, C_SACCHC40_B, C_SACCHC41_B, C_SACCHC42_B, C_SACCHC43_B, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: Config_SDCCH4_C_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell C.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellC, C_FCCH_C, C_SCH_C, C_BCCH_C_1, C_PCH_C_1, C_AGCH_C_1, C_RACH_C_1, C_SDCCH4_C, C_SACCHC4_C, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_C, C_SDCCH40_C, C_SDCCH41_C, C_SDCCH42_C, C_SDCCH43_C, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_C, C_SACCHC40_C, C_SACCHC41_C, C_SACCHC42_C, C_SACCHC43_C, "dummy", "dummy", "dummy", "dummy"))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Config_SDCCH8_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set one physical channel used as SDCCH8 channel for instance 1 of cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_SDCCH8_A_1, C_SACCHC8_A_1, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH8_A_1, C_SDCCH80_A_1, C_SDCCH81_A_1, C_SDCCH82_A_1, C_SDCCH83_A_1, C_SDCCH84_A_1, C_SDCCH85_A_1, C_SDCCH86_A_1, C_SDCCH87_A_1, C_SACCHC8_A_1, C_SACCHC80_A_1, C_SACCHC81_A_1, C_SACCHC82_A_1, C_SACCHC83_A_1, C_SACCHC84_A_1, C_SACCHC85_A_1, C_SACCHC86_A_1, C_SACCHC87_A_1))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: CombinedBCCH_A_CBMS(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.

Default: OtherEvents

Comments: time slot = 0, ARFCN = 20 (GSM900) or ARFCN = 590 (DCS1800) cell A for RR testing.

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_CBMS_A_1(par_bspwr,mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_CBMS_A_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			

Detailed Comments: 1. For P-GSM900.
2. For DCS1800.

Test Step Dynamic Behaviour

Test Step Name: CombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_A_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_A_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			

Detailed Comments: 1. For P-GSM900.
2. For DCS1800.

Test Step Dynamic Behaviour

Test Step Name: CombinedBCCH_A_ho(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3; td, fn:INTEGER)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_A_ho_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm, td, fn)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_A_ho_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm, td, fn)			

Detailed Comments: 1. For P-GSM900.
2. For DCS1800.

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_B(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_B_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_B_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_B_ho(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3; td, fn:INTEGER)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_B_ho_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm, td, fn)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_B_ho_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm, td, fn)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_C(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell C.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_C_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_C_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour

Test Step Name: NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set one physical channel used as FCHH_SCH_BCCH_CCCH for cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_BCCH_CCCH_A_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_A_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm)			

Detailed Comments:
1. For P-GSM900.
2. For DCS1800.

Test Step Dynamic Behaviour

Test Step Name: NonCombinedBCCH_A_2(acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set a physical channel and map the second BCCH, CCCH onto the physical channel which represents cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_BCCH_CCCH_A_2(63, 19, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnA), C_BCCH_CCCH_2, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_A_2(63, 15, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnAd), C_BCCH_CCCH_2, ta, babr, cch_con, bpm)			

Detailed Comments:
1. For P-GSM900.
2. For DCS1800.

Test Step Dynamic Behaviour

Test Step Name: NonCombinedBCCH_A_3(acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set a physical channel and map the third BCCH, CCCH onto the physical channel which represents cell A.

Default: OtherEvents

Comments: time slot = 4, ARFCN = TSPX_BCCHcarrierA

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_BCCH_CCCH_A_3(63, 19, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnA), C_BCCH_CCCH_3, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_A_3(63, 15, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnAd), C_BCCH_CCCH_3, ta, babr, cch_con, bpm)			

Detailed Comments:
1. For P-GSM900.
2. For DCS1800.

Test Step Dynamic Behaviour					
Test Step Name:	NonCombinedBCCH_A_4(acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set a physical channel and map the fourth BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Comments:	time slot = 6, ARFCN = TSPX_BCCHcarrierA				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_BCCH_CCCH_A_4(63, 19, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnA), C_BCCH_CCCH_4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_A_4(63, 15, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnAd), C_BCCH_CCCH_4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	NonCombinedBCCH_A_ho(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3; td, fn:INTEGER)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH onto the physical channel which represents cell A for RR testing.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM]			1.
2		+Config_BCCH_CCCH_A_ho_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm, td, fn)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_A_ho_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm, td, fn)			
Detailed Comments:	1. For P-GSM900. 1.1 Frequence and cell_id for cell A in HO cases 2. For DCS1800. 2.1 Frequence and cell_id for cell A in HO cases				

Test Step Dynamic Behaviour

Test Step Name: NonCombinedBCCH_B(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH onto the physical channel which represents cell B for RR testing.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM]			1.
2		+Config_BCCH_CCCH_B_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_B_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm)			

Detailed Comments:
1. For P-GSM900.
1.1 Frequenz and cell_id for cell B in HO cases
2. For DCS1800.
2.1 Frequenz and cell_id for cell B in HO cases

Test Step Dynamic Behaviour

Test Step Name: NonCombinedBCCH_B_ho(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3; td, fn:INTEGER)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH onto the physical channel which represents cell B for RR testing.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM]			1.
2		+Config_BCCH_CCCH_B_ho_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm, td, fn)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_B_ho_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm, td, fn)			

Detailed Comments:
1. For P-GSM900.
1.1 Frequenz and cell_id for cell B in HO cases
2. For DCS1800.
2.1 Frequenz and cell_id for cell B in HO cases

Test Step Dynamic Behaviour

Test Step Name: FullRateCh_A_1(acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg, Freqd:FRQPARA; arfcng, arfcnd:INTEGER; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set one physical channel used as TCHF_ACCH's for instance 1 of cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[arfcng <> C_Noarfcn]			Non hopping
2		+FullRateCh_A_1_nociph(acttypeT, slotT, tscT, chmod, FreqTCH(arfcng), FreqTCH(arfcnd), ta, babr, cch_con, bpm)			
3		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			
4		[TSPC_PGSM OR TSPC_EGSM]			
5		(TCV_tch_arfcn:= arfcng)			
6		[TSPC_DCS]			
7		(TCV_tch_arfcn:= arfcnd)			
8		[arfcng = C_Noarfcn]			Hopping
9		+FullRateCh_A_1_nociph(acttypeT, slotT, tscT, chmod, Freqg, Freqd, ta, babr, cch_con, bpm)			
10		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: FullRateCh_A_1_nociph(acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg, Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set one physical channel used as TCHF_ACCH's for instance 1 of cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_chTch := C_FACCHF_A_1, TCV_sacchTch := C_SACCHF_A_1, TCV_chtype := '00001'B)			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_FACCHF_A_1(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_TCHF_ACCHF_1, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
5		[TSPC_DCS]			
6		+Config_FACCHF_A_1(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_TCHF_ACCHF_1, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: FullRateCh_A_2(acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set one physical channel used as TCHF_ACCH's for instance 2 of cell A for TC_26_6_13_1.

Default: OtherEvents

Comments: used as after time channel.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+FullRateCh_A_2_nociph(acttypeT, slotT, tscT, chmod, Freqg,Freqd, ta, babr, cch_con, bpm)			
2		(TCV_Null := OM_CphMd(TCV_chTch1, CphMod_01, TCV_CphKey))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: FullRateCh_A_2_nociph(acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)
Group: GSM_MS_EFR_4220/management/ChConfig/
Objective: To set one physical channel used as TCHF_ACCH's for instance 2 of cell A for TC_26_6_13_1.
Default: OtherEvents
Comments: used as after time channel.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_chTch1 := C_FACCHF_A_2, TCV_sacchTch1 := C_SACCHF_A_2)			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_FACCHF_A_2(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_TCHF_ACCHF_2, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)			
5		[TSPC_DCS]			
6		+Config_FACCHF_A_2(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_TCHF_ACCHF_2, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: FullRateCh_B_1(acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)
Group: GSM_MS_EFR_4220/management/ChConfig/
Objective: To set one physical channel used as TCHF_ACCH's for instance 1 of cell B.
Default: OtherEvents
Comments: FullRateCh_B_1.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+FullRateCh_B_1_nociph(acttypeT, slotT, tscT, chmod, Freqg, Freqd, ta, babr, cch_con, bpm)			
2		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: FullRateCh_B_1_nociph(acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)
Group: GSM_MS_EFR_4220/management/ChConfig/
Objective: To set one physical channel used as TCHF_ACCH's for instance 1 of cell B.
Default: OtherEvents
Comments: Used in TC_26_6_13_5 for after time channel

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_chTch := C_FACCHF_B_1, TCV_sacchTch := C_SACCHF_B_1)			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_FACCHF_B_1(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_TCHF_ACCHF_1, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5_B, TCV_sysinfo6_B)			
5		[TSPC_DCS]			
6		+Config_FACCHF_B_1(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_TCHF_ACCHF_1, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5_B, TCV_sysinfo6_B)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: HalfRateCh_A_1(sub:B_1; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg, Freqd:FRQPARA; arfcng, arfcnd:INTEGER; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set one physical channel used as TCHH_ACCH's for instance 1 of cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[arfcng <> C_Noarfcn]			not hopping
2		+HalfRateCh_A_1_nociph(sub, acttypeT, slotT, tscT, chmod, FreqTCH(arfcng), FreqTCH(arfcnd), ta, babr, cch_con, bpm)			
3		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			
4		[TSPC_PGSM OR TSPC_EGSM]			
5		(TCV_tch_arfcn:= arfcng)			
6		[TSPC_DCS]			
7		(TCV_tch_arfcn:= arfcnd)			
8		[arfcng = C_Noarfcn]			hopping
9		+HalfRateCh_A_1_nociph(sub, acttypeT, slotT, tscT, chmod, Freqg, Freqd, ta, babr, cch_con, bpm)			
10		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: HalfRateCh_A_1_nociph(sub:B_1; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg, Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_EFR_4220/management/ChConfig/

Objective: To set one physical channel used as TCHH_ACCH's for instance 1 of cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_chTch := OC_SubchOfFacchh(sub, C_CellA, 1), TCV_sacchTch := OC_SubchOfSacchh(sub, C_CellA, 1))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_FACCHH_A_1(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_TCHH_ACCHH_1, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
5		+localtree			
6		[TSPC_DCS]			
7		+Config_FACCHH_A_1(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_TCHH_ACCHH_1, ta, babr, cch_con, bpm)			
8		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
9		+localtree			
		localtree			
10		[sub = '0'B]			
11		(TCV_chtype := '00010'B)			
12		[sub = '1'B]			
13		(TCV_chtype := '00011'B)			

Detailed Comments:

Test Step Dynamic Behaviour					
Test Step Name:	HalfRateCh_A_2(bstring:B_1; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set one physical channel used as TCHH_ACCH's for instance 2 of cell A for TC_26_6_13_1.				
Default:	OtherEvents				
Comments:	used for after time channel.				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+HalfRateCh_A_2_nociph(bstring, acttypeT, slotT, tscT, chmod, Freqg, Freqd, ta, babr, cch_con, bpm)			
2		(TCV_Null := OM_CphMd(TCV_chTch1, CphMod_01, TCV_CphKey))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	HalfRateCh_A_2_nociph(bstring:B_1; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set one physical channel used as TCHH_ACCH's for instance 2 of cell A for TC_26_6_13_1.				
Default:	OtherEvents				
Comments:	used for after time channel.				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_chTch1 := OC_SubchOfFacchh(bstring, C_CellA, 2), TCV_sacchTch1 := OC_SubchOfSacchh(bstring, C_CellA, 2))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_FACCHH_A_2(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_TCHH_ACCHH_2, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)			
5		[TSPC_DCS]			
6		+Config_FACCHH_A_2(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_TCHH_ACCHH_2, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch1, TCV_sysinfo5, TCV_sysinfo6)			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	HalfRateCh_B_1(bstring:B_1; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set one physical channel used as TCHH_ACCH's for instance 1 of cell B				
Default:	OtherEvents				
Comments:	used as after time channel assigned by handover command				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+HalfRateCh_B_1_nociph(bstring, acttypeT, slotT, tscT, chmod, Freqg, Freqd, ta, babr, cch_con, bpm)			
2		(TCV_Null := OM_CphMd(TCV_chTch, CphMod_01, TCV_CphKey))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	HalfRateCh_B_1_nociph(bstring:B_1; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set one physical channel used as TCHH_ACCH's for instance 1 of cell B				
Default:	OtherEvents				
Comments:	used as after time channel assigned by handover command				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_chTch := OC_SubchOfFacchh(bstring, C_CellB, 1), TCV_sacchTch := OC_SubchOfSacchh(bstring, C_CellB, 1))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_FACCHH_B_1(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_TCHH_ACCHH_1, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5_B, TCV_sysinfo6_B)			
5		[TSPC_DCS]			
6		+Config_FACCHH_B_1(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_TCHH_ACCHH_1, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5_B, TCV_sysinfo6_B)			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	SDCCH8_A_1_nociph(bstring:B_3; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_EFR_4220/management/ChConfig/				
Objective:	To set one physical channel used as SDCCH8 channel for instance 1 of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_ch := OC_SubchOfSdcch8(bstring, C_CellA, 1), TCV_sacch8 := OC_SubchOfSacch8(bstring, C_CellA, 1))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_SDCCH8_A_1(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_SDCCH8_SACCHC8_1, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacch8, TCV_sysinfo5, TCV_sysinfo6)			
5		[TSPC_DCS]			
6		+Config_SDCCH8_A_1(63, 19, chmod, acttypeT, slotT, tscT, Freqd, C_SDCCH8_SACCHC8_1, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacch8, TCV_sysinfo5, TCV_sysinfo6)			
Detailed Comments:					

Test Step Group Miscellaneous

Test Step Dynamic Behaviour					
Test Step Name:	Adjust_gsmanddcs_powerlvl(powerlevel1,powerlevel2:INTEGER; pdu_ass: ASS_CMD_PDU)				
Group:	GSM_MS_EFR_4220/Miscellaneous/				
Objective:					
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			
2		[powerlevel1 <> 0]			
3		(pdu_ass.pcmd := Pcmd_19(INT_TO_BIT(powerlevel1,5)))			
4		[powerlevel1 = 0]			
5		[TSPC_DCS]			
6		(pdu_ass.pcmd := Pcmd_19(INT_TO_BIT(powerlevel2,5)))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	AssCh_complete(oldch,newch:LOGICCH; pdu_ass: ASS_CMD_PDU)				
Group:	GSM_MS_EFR_4220/Miscellaneous/				
Objective:	To provide a generic test step to Assign a traffic channel. Assign complete expected and verdict is PASS.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqAssCmd START T_dlyAss	AssCmd(oldch,pdu_ass)		
2	E_0053	?TIMEOUT T_dlyAss		(F)	
3		+PostLinkRelEnd(oldch)			
4		L?DL_EstIn (TCV_FnAss := DL_EstIn.fn) CANCEL T_dlyAss	DLEstInd(newch)		
5	E_0054	L?DL_DatInAssCom	AssCmp(newch, AsgnCmp_02)	(P)	
6		L!MDL_RelRq	MDLRelReq(oldch)		
7		L?DL_EstIn CANCEL T_dlyAss	DLEstInd(oldch)		
8	E_0055	L?DL_DatInAssfl	AssFl_any_cau(oldch)	(F)	
9		+PostLinkRelEnd(oldch)			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: AssCmdGen(cell:CellID; rate:IA5String; slot:SN; tsc:TSC; sch:B_1)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective: To generate the ASSIGNMENT COMMAND message.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[rate <> C_Half]			Full rate
2		(TCV_Tchtype := '00001'B)			
3		+ltree_Asgn			
4		[rate = C_Half]			Half rate
5		[sch = '0'B]			
6		(TCV_Tchtype := '00010'B)			
7		+ltree_Asgn			
8		[sch = '1'B]			
9		(TCV_Tchtype := '00011'B)			
10		+ltree_Asgn			
		ltree_Asgn			
11		[cell = C_CellA]			
12		+ltree_AsgnTchA			
13		[cell = C_CellB]			
14		+ltree_AsgnTchB			
		ltree_AsgnTchA			
15		[TSPC_PGSM OR TSPC_EGSM]			
16		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, slot, tsc, 7, C_arfcn_tchA, CellChDes_Omit, TCV_ChMod, StartingTm_omit, CphMod_omit))			
17		[TSPC_DCS]			
18		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, slot, tsc, 3, C_arfcn_tchAd, CellChDes_Omit, TCV_ChMod, StartingTm_omit, CphMod_omit))			
		ltree_AsgnTchB			
19		[TSPC_PGSM OR TSPC_EGSM]			
20		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, slot, tsc, 7, C_arfcn_tchB, CellChDes_Omit, TCV_ChMod, StartingTm_omit, CphMod_omit))			
21		[TSPC_DCS]			
22		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, slot, tsc, 3, C_arfcn_tchBd, CellChDes_Omit, TCV_ChMod, StartingTm_omit, CphMod_omit))			

Detailed Comments: The test step makes the assumption that the channel rate is known when the test step is called.

Test Step Dynamic Behaviour

Test Step Name: AssCmdGen_fh(rate:IA5String; slot:SN; tsc:TSC; sch:B_1; par_pwlvlg, par_pwlvld:INTEGER; maio:MAIO; hsn:HSN; frql_g, frql_d:FRQL; cchd_g, cchd_d:CCHD; chmod:CHMOD; ma_g, ma_d:MA; Cphms:CPHMS)

Group: GSM_MS_EFR_4220/Miscellaneous/

Objective: To generate a ASSIGNMENT COMMAND message in case of hopping without starting time.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[rate <> C_Half]			Full rate
2		(TCV_Tchtype := '00001'B)			
3		+ltree_Asgn			
4		[rate = C_Half]			Half rate
5		[sch = '0'B]			
6		(TCV_Tchtype := '00010'B)			
7		+ltree_Asgn			
8		[sch = '1'B]			
9		(TCV_Tchtype := '00011'B)			
10		+ltree_Asgn			
		ltree_Asgn			
11		[TSPC_PGSM OR TSPC_EGSM]			
12		(TCV_AssCmd := AsgnCmd_fh(slot, tsc, TCV_Tchtype, par_pwlvlg, maio, hsn, frql_g, cchd_g, chmod, ma_g, Cphms))			
13		[TSPC_DCS]			
14		(TCV_AssCmd := AsgnCmd_fh(slot, tsc, TCV_Tchtype, par_pwlvld, maio, hsn, frql_d, cchd_d, chmod, ma_d, Cphms))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Authentication(ch: LOGICCH; cksn: BITSTRING; rand:RAND)

Group: GSM_MS_EFR_4220/Miscellaneous/

Objective:

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_DatRqAuthRq	AuthReq(ch, AuthRequest(cksn, rand))		
2		L?DL_DatInAuthRes (TCV_Sres := DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
3		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, rand))			
4	E_0056	[NOT TCV_Res]		(F)	1)
5	E_0057	[TCV_Res]		(P)	

Detailed Comments: Authentication fails.

Test Step Dynamic Behaviour

Test Step Name: CallProcGen(setup:SETUP_MO_PDU; srv:SERVICES)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective: To generate a CALL PROCEEDING message which is compatible with input value 'setup'..
Default: OtherEvents
Comments: The generated CALL PROCEEDING is in TCV_CallProc. This test step can handle the following dual mode services: the first BCAP is speech or the second BCAP is speech

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_CallProc := CallProced_03, TCV_CallProc.ti := TCV_TI)			
2		[(srv = C_AltSpchData) OR (srv = C_SpchData) OR (srv = C_AltSpchFax)]			dual service
3		[setup.bcap1.itc = C_Speech]			1st BCAP is speech
4		(TCV_ce3 := setup.bcap2.ce)			
5		[(TCV_ce3 <> C_BothT) OR(TCV_ce3 <> C_BothNT)]			1.
6		[(TCV_ce3 = C_BothT) OR (TCV_ce3 = C_BothNT)]			2.
7		(TCV_CallProc.bcap1 := setup.bcap1, TCV_CallProc.bcap1.rchr := '00'B, TCV_CallProc.bcri := setup.bcri, TCV_CallProc.bcap2 := setup.bcap2, TCV_CallProc.bcap2.ce := INT_TO_BIT((BIT_TO_INT(setup.bcap2.ce) -2), 2), TCV_CallProc.bcap2.nirr := C_nirr_nomeaning)			
8		[NOT (setup.bcap1.itc = C_Speech)]			
9		(TCV_ce1 := setup.bcap1.ce)			
10		[(TCV_ce1 <> C_BothT) AND(TCV_ce1 <> C_BothNT)]			3.
11		[(TCV_ce1 = C_BothT) OR (TCV_ce1 = C_BothNT)]			4.
12		(TCV_CallProc.bcap1 := setup.bcap1, TCV_CallProc.bcap1.ce := INT_TO_BIT((BIT_TO_INT(setup.bcap1.ce) -2), 2), TCV_CallProc.bcap1.nirr := C_nirr_nomeaning, TCV_CallProc.bcri := setup.bcri, TCV_CallProc.bcap2 := setup.bcap2, TCV_CallProc.bcap2.rchr := '11'B)			
13		[(srv <> C_AltSpchData) AND(srv <> C_SpchData) AND(srv <> C_AltSpchFax)]			
14		[srv = C_EmgCall]			8.
15		[srv <> C_EmgCall]			
16		[setup.bcap1.itc = C_Speech]			5.
17		[setup.bcap1.itc <> C_Speech]			
18		(TCV_ce1 := setup.bcap1.ce)			
19		[(TCV_ce1 = C_BothT) OR (TCV_ce1 = C_BothNT)]			6.
20		(TCV_CallProc.bcap1 := setup.bcap1, TCV_CallProc.bcap1.ce := INT_TO_BIT((BIT_TO_INT(setup.bcap1.ce) -2), 2), TCV_CallProc.bcap1.nirr := C_nirr_nomeaning)			
21		[(TCV_ce1 <> C_BothT) OR(TCV_ce1 <> C_BothNT)]			7.

Detailed Comments:

1. BCAP1 is speech and BCAP2 does not reqire negotiation, the CallProc will contain no BCAP's
2. BCAP1 is speech and BCAP2 requires negotiation
3. BCAP2 is speech and BCAP1 does not reqire negotiation, the CallProc will contain no BCAP's
4. BCAP2 is speech and BCAP1 requires negotiation
5. BCAP2 does not exist and BCAP1 is speech, the CallProc will contain no BCAP's
6. BCAP2 does not exist and BCAP1 requires negotiation
7. BCAP2 does not exist and BCAP1 does not require negotiation
8. Emergency call, the CallProc will contain no BCAP

Test Step Dynamic Behaviour

Test Step Name: CCCH_group_Paging_group(ccd: CCD; imsi: HEXSTRING)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective: To calculate the CCCH group and the Paging group from the IMSI and the Control Channel Descriptor according to GSM 05.02, subclause 6.5.2.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[ccd.ccch_con = '000'B]			
2		(TCV_bs_cc_chans := 1)			
3		+localtree1			
4		[ccd.ccch_con = '001'B]			
5		(TCV_bs_cc_chans := 1)			
6		+localtree1			
7		[ccd.ccch_con = '010'B]			
8		(TCV_bs_cc_chans := 2)			
9		+localtree1			
10		[ccd.ccch_con = '100'B]			
11		(TCV_bs_cc_chans := 3)			
12		+localtree1			
13		[ccd.ccch_con = '110'B]			
14		(TCV_bs_cc_chans := 4)			
15		+localtree1			
		localtree1			
16		(TCV_bs_ag_blks_res := BIT_TO_INT(ccd.babr), TCV_bs_pa_mfrms := (2 + BIT_TO_INT(ccd.bpm)))			
17		[ccd.ccch_con = '001'B]			i.e. combined
18	E_0058	[(TCV_bs_ag_blks_res < 0) OR(TCV_bs_ag_blks_res > 2)]		I	Stop! A tester error.
19		[(TCV_bs_ag_blks_res >= 0) AND (TCV_bs_ag_blks_res <= 2)]			
20		(TCV_tmp := 3 - TCV_bs_ag_blks_res)			
21		+localtree2			
22		[ccd.ccch_con <> '001'B]			i.e. not combined
23	E_0059	[(TCV_bs_ag_blks_res < 0) OR(TCV_bs_ag_blks_res > 7)]		I	Stop! A tester error.
24		[(TCV_bs_ag_blks_res >= 0) AND (TCV_bs_ag_blks_res <= 7)]			
25		(TCV_tmp := 9 - TCV_bs_ag_blks_res)			
26		+localtree2			ini. TCV Pgg
		localtree2			
27		(TCV_tmp := TCV_tmp * TCV_bs_pa_mfrms)			
28		(TCV_Pgg := INT_TO_BIT((((OC_BCDtoInt(imsi, 3)) MOD(TCV_bs_cc_chans * TCV_tmp)) MOD TCV_tmp), 8), TCV_Ccchg := (((OC_BCDtoInt(imsi, 3)) MOD(TCV_bs_cc_chans * TCV_tmp)) / TCV_tmp))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: CC_Est_MT_Call
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective: To perform the CC message exchange to establish a mobile station terminating call (late assignment).
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Setup_mt.sig := Signal_01)			
2		L!DL_DatRqSetup	SetupSnd(TCV_ch, TCV_Setup_mt)		
3		L?DL_DatInCallCo	CallCfm(CallConfirm_01(TI_01))		1)
4		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		
5		L?DL_DatInAlert	AlertRcv(AlertingInd_01(TI_01))		
6		(TCV_Null := OO_HookOff())			
7		L?DL_DatInConn	ConnRcv(Connect_01(TI_01))		

Detailed Comments:
1. If the MS supports the bearer capabilities, which are give in Setup message, it has to accept them. However, it may still send bearer capabilites in the Call Confirm message for either EFR or Half Rate

Test Step Dynamic Behaviour

Test Step Name: CC_EstMsTermCall(rate:IA5String; sub:B_1)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective: To perform the CC message exchange to establish a mobile station terminating call with non hopping channel (speech or data call). (Similar to EstMsTermFullRateCallNonFH but without initial RRmtcallprepare)
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+CC_Est_MT_Call			
2		+ltree_Asgn			
3		+Adjust_gsmanddcs_powerlvl(0, 15, TCV_AssCmd)			
4		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
5		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
		ltree_Asgn			
6		[rate = C_Full]			Full rate
7		(TCV_Tchtype := '00001'B)			
8		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, TCV_asscmd_ts, TSPX_TscDef, 9, TCV_tch_arfcn, CellChDes_Omit, ChMod_speech_iei, StartingTm_omit, CphMod_omit))			
9		[rate = C_Full_v2]			Full rate v2
10		(TCV_Tchtype := '00001'B)			
11		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, TCV_asscmd_ts, TSPX_TscDef, 9, TCV_tch_arfcn, CellChDes_Omit, ChMod_EFR2_iei, StartingTm_omit, CphMod_omit))			
12		[rate = C_Half]			Half rate
13		(TCV_Tchtype := INT_TO_BIT((2 + BIT_TO_INT(sub)), 5))			
14		(TCV_AssCmd := AsgnCmd_nfh(TCV_Tchtype, TCV_asscmd_ts, TSPX_TscDef, 9, TCV_tch_arfcn, CellChDes_Omit, ChMod_omit, StartingTm_omit, CphMod_omit))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Ciphering_on(ch: LOGICCH)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective: Send cipher command and wait for successful completion
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Null := OM_CphMdChg(ch, CphMod_01, TCV_CphKey))			
2		L!DL_DatRqCphmCmd	CphCmd(ch, CphModeCmd_01)		
3	E_0060	L?DL_DatInCphmCom	CphCom(CphModeCmp_01)	(P)	

Detailed Comments:

Test Step Dynamic Behaviour					
Test Step Name:	EFR_ChkRcvSpchPrefs				
Group:	GSM_MS_EFR_4220/Miscellaneous/				
Objective:	OtherEvents				
Default:					
Comments:	Check speech versions received in EFR BCAP against IXIT file values. All supported speech versons must be present (in any order) in the received EFR BCAP.				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+EFR_SpchPrefInBCAP(TSPX_EFR_SpchPref1)			
2		+EFR_SpchPrefInBCAP(TSPX_EFR_SpchPref2)			
3		[TSPX_EFR_SpchPrefsNo > 2]			
4		+EFR_SpchPrefInBCAP(TSPX_EFR_SpchPref 3)			
5	E_00 62	[TSPX_EFR_SpchPrefsNo = 2]		(P)	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	EFR_SpchPrefInBCAP(svi:B_4)				
Group:	GSM_MS_EFR_4220/Miscellaneous/				
Objective:	OtherEvents				
Default:					
Comments:	Check for an individual speech preferences being present in received EFR BCAP.				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1	E_00 63	[TCV_EFR_svi1 = svi]		(P)	
2	E_00 64	[TCV_EFR_svi2 = svi]		(P)	
3		[TSPX_EFR_SpchPrefsNo = 3]			
4	E_00 65	[TCV_EFR_svi3 = svi]		(P)	
5	E_00 66	[C_Yes]		I	IXIT ERROR
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: EFR_RcvSetupOrEsetup(Setup: SETUP_MO_PDU; Esetup:ESETUP_PDU; Ecall:BOOLEAN)

Group: GSM_MS_EFR_4220/Miscellaneous/

Objective: To check that Mobile Originated SETUP or ESETUP message contains a bearer capability IE (optional for ESETUP) with all the speech versions supported by the EFR mobile. If matching is successful then a CALL PROCEEDING message response will be sent to the EFR mobile.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[NOT Ecall]			
2		[TSPX_EFR_SpchPrefsNo = 2]			
3		L?DL_DatInSetup (TCV_Setup_mo1 := DL_DatInSetup.msg, TCV_TI := TCV_Setup_mo1.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_EFR_svi1 := TCV_Setup_mo1.bcap1.oct_3a_etc_1.svi, TCV_EFR_svi2 := TCV_Setup_mo1.bcap1.oct_3a_etc_2.svi)	SetupRcv(Setup)		
4		+ltree_chkspeech			
5		[TSPX_EFR_SpchPrefsNo = 3]			
6		L?DL_DatInSetup (TCV_Setup_mo1 := DL_DatInSetup.msg, TCV_TI := TCV_Setup_mo1.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_EFR_svi1 := TCV_Setup_mo1.bcap1.oct_3a_etc_1.svi, TCV_EFR_svi2 := TCV_Setup_mo1.bcap1.oct_3a_etc_2.svi, TCV_EFR_svi3 := TCV_Setup_mo1.bcap1.oct_3a_etc_3.svi)	SetupRcv(Setup)		
7		+ltree_chkspeech			
8	E_0067	[C_Yes]		I	IXIT ERROR
9		[Ecall]			
10		[TSPC_EFR_EmgCallBcap]			
11		[TSPX_EFR_SpchPrefsNo = 2]			
12		L?DL_DatInESetup (TCV_Esetup1 := DL_DatInESetup.msg, TCV_Setup_mo1.ti := TCV_Esetup1.ti, TCV_Setup_mo1.bcap1 := TCV_Esetup1.bcap, TCV_Setup_mo1.ccpd := TCV_Esetup1.ccpd, TCV_Setup_mo1.mt := TCV_Esetup1.mt, TCV_TI := TCV_Setup_mo1.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_EFR_svi1 := TCV_Setup_mo1.bcap1.oct_3a_etc_1.svi, TCV_EFR_svi2 := TCV_Setup_mo1.bcap1.oct_3a_etc_2.svi)	ESetupRcv(Esetup)		
13		+ltree_chkspeech			
14		[TSPX_EFR_SpchPrefsNo = 3]			
15		L?DL_DatInESetup (TCV_Esetup1 := DL_DatInESetup.msg, TCV_TI := TCV_Setup_mo1.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B, TCV_EFR_svi1 := TCV_Setup_mo1.bcap1.oct_3a_etc_1.svi, TCV_EFR_svi2 := TCV_Setup_mo1.bcap1.oct_3a_etc_2.svi, TCV_EFR_svi3 := TCV_Setup_mo1.bcap1.oct_3a_etc_3.svi)	ESetupRcv(Esetup)		
16		+ltree_chkspeech			
17	E_0068	[C_Yes]		I	IXIT ERROR
18		[NOT TSPC_EFR_EmgCallBcap]			
19		L?DL_DatInESetup (TCV_Esetup1 := DL_DatInESetup.msg, TCV_TI := TCV_Setup_mo1.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B)	ESetupRcv(Esetup)		
20		+CallProcGen(TCV_Setup_mo1, TCV_Service)			
		ltree_chkspeech			
21		+EFR_ChkRcvSpchPrefs			check all spchprefs are present in bcap
22		+CallProcGen(TCV_Setup_mo1, TCV_Service)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: FnArith(fn, fn1:FN)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective:
Default: OtherEventsFail
Comments: To calculate the difference of two frame numbers 'fn1' and 'fn'. The test step uses TCV_Cnt1, TCV_Cnt2, TCV_K and TCV_Time

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Cnt1 := (51 *((BIT_TO_INT(fn.t3) - BIT_TO_INT(fn.t2) + 26) MOD 26)) + BIT_TO_INT(fn.t3) +(1326 * BIT_TO_INT(fn.t1_)), TCV_Cnt2 := (51 *((BIT_TO_INT(fn1.t3) - BIT_TO_INT(fn1.t2) + 26) MOD 26)) + BIT_TO_INT(fn1.t3) +(1326 * BIT_TO_INT(fn1.t1_)), TCV_K := ((TCV_Cnt2-TCV_Cnt1+42432) MOD 42432), TCV_Time := TCV_K * 120 / 26)			GSM 04.08 section 10.5.2.30. TCV_K stores the frame number difference. TCV_Time stores the time difference in ms.

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: ImsiDetach(par_mi:MI; ta:TA; sim_rmvd:BOOLEAN)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv) AND(TSPC_DetachOnSIMRmv)]			SIM needs to be removed.
2		(TCV_Null :=OO_SIMRmv())			
3		+ltree_imsidetach			
4		[TSPC_SwitchOnOff]			
5		(TCV_Null := OO_SwitchOff())			
6		+ltree_imsidetach			
7		[TSPC_DetachOnPwrDn]			
8		(TCV_Null := OO_PowerDown())			Remove power source
9		+ltree_imsidetach			
10		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv) OR(NOT TSPC_DetachOnSIMRmv)) AND(NOT TSPC_SwitchOnOff) AND(NOT TSPC_DetachOnPwrDn))]			
11		(TCV_Null := OO_PowerDown())			Remove power source
		ltree_imsidetach			
12		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
13		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
14		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
15	E_0069	L?DL_EstInImsidIn	ImsiDet(ImsiDetach_30(par_mi))	(P)	
16		ACTIVATE(OtherEventsFail)			Restore normal default
17		+ChanRel(TCV_ch)			

Detailed Comments: If IMSI detach procedure has been executed, the channel will be relesed. If ATT=0 no detach has been executed, the channel is still connected.

Test Step Dynamic Behaviour

Test Step Name: MM_PwrOrSimOff(sim_rmvd:BOOLEAN)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv)]			SIM needs to be removed.
2		(TCV_Null :=OO_SIMRmv())			
3		[TSPC_SwitchOnOff]			
4		(TCV_Null := OO_SwitchOff())			
5		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv)) AND(NOT TSPC_SwitchOnOff))]			
6		(TCV_Null := OO_PowerDown())			Remove power source

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_PwrOrSimOn(sim_rmvd:BOOLEAN)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv)]			SIM has been removed
2		(TCV_Null := OO_SIMIns())			
3		[TSPC_SwitchOnOff]			
4		(TCV_Null := OO_SwitchOn())			
5		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv)) AND(NOT TSPC_SwitchOnOff))]			
6		(TCV_Null := OO_PowerUp())			Restore power source

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: RcvSetupOrEsetup(Setup: SETUP_MO_PDU; Esetup:ESETUP_PDU; Ecall:BOOLEAN)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective: To manage Setup Mobile Originated.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[NOT Ecall]			
2		L?DL_DatInSetup (TCV_Setup_mo1 := DL_DatInSetup.msg, TCV_TI := TCV_Setup_mo1.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B)	SetupRcv(Setup)		
3		+CallProcGen(TCV_Setup_mo1, TCV_Service)			
4		[Ecall]			
5		L?DL_DatInESetup (TCV_Esetup1 := DL_DatInESetup.msg, TCV_Setup_mo1.ti := TCV_Esetup1.ti, TCV_Setup_mo1.bcap1 := TCV_Esetup1.bcap, TCV_Setup_mo1.ccpd := TCV_Esetup1.ccpd, TCV_Setup_mo1.mt := TCV_Esetup1.mt, TCV_TI := TCV_Setup_mo1.ti, TCV_TI0 := TCV_TI, TCV_TI.ti_f := '1'B)	ESetupRcv(Esetup)		
6		+CallProcGen(TCV_Setup_mo1, TCV_Service)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: RR_hocomp1(time_fn_fn1:INTEGER; oldch:LOGICCH)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective: To finish the HO-procedure. Timing advance = 20 bits period
Default: RcvHdOvAcc, OtherEvents
Comments: used var's: TCV_ch, TCV_Horf, TCV_Cntref

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RacInHoacc (TCV_FnBurst := DL_RacInHoacc.fn)	HndOvAccRcv(TCV_ch, HandOverAcc_02(TCV_Horf))		1.
2		L?DL_EstIn	DLEstInd(TCV_ch)		
3		(TCV_HoaccPara := OM_GetHoaccPara(TCV_ch))			
4	E_0148	[(OC_OctToInt(TCV_HoaccPara.num_bursts) >= TCV_Cntref) AND(OC_OctToInt(TCV_HoaccPara.num_bursts) <= (TCV_Cntref+20))]		(P)	2.
5		+localtree_continue			
6	E_0149	[C_Yes]		(F)	
7		+localtree_continue			
		localtree_continue			
8		(TCV_Fn1:= OM_ReturnFn(TCV_ch))			
9		L?DL_DatInHoCom(TCV_Fn := DL_DatInHoCom.fn)	HndOvCmpRcv(TCV_ch, HandOverCmp_20)		
10		L!MDL_RelRq	MDLRelReq(oldch)		Release the original channel.
11		+FnArith(TCV_Fn1, TCV_Fn)			3.
12	E_0070	[TCV_Time > time_fn_fn1]		(F)	
13	E_0071	[TCV_Time <= time_fn_fn1]		(P)	

Detailed Comments:
1. Match the 1st access burst. The access bursts except the 1st one are matched by the default test step RcvHdOvAcc.
2. Check the number of access bursts received with tolerance +20 (10 for FACCH interleave & 10 for MS processing).
3. Check HO-time.

Test Step Dynamic Behaviour

Test Step Name: RRmtcallprepare(ta:TA; rand:RAND)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective: To prepare a mobile terminating call establishment.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		+PreEstRRConn(TCV_slot, TCV_tsc, ta)			
2		+Authentication(TCV_ch, TCV_cksn, rand)			
3		+Ciphering_on(TCV_ch)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SelectPagingCh(cell:CellID)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective: To assign a paging channel and an access grant channel to the variable TCV_PgCh and TCV_agch respectively, depending on the parameter 'cell' (cell ID)
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[cell = C_CellA]			
2		[TCV_Ccchg=0]			
3		(TCV_PgCh := C_PCH_A_1, TCV_agch := C_AGCH_A_1, TCV_RaCh := C_RACH_A_1)			
4		[TCV_Ccchg=1]			
5		(TCV_PgCh := C_PCH_A_2, TCV_agch := C_AGCH_A_2, TCV_RaCh := C_RACH_A_2)			
6		[TCV_Ccchg=2]			
7		(TCV_PgCh := C_PCH_A_3, TCV_agch := C_AGCH_A_3, TCV_RaCh := C_RACH_A_3)			
8		[TCV_Ccchg=3]			
9		(TCV_PgCh := C_PCH_A_4, TCV_agch := C_AGCH_A_4, TCV_RaCh := C_RACH_A_4)			
10		[cell = C_CellB]			
11		[TCV_Ccchg=0]			
12		(TCV_PgCh := C_PCH_B_1, TCV_agch := C_AGCH_B_1, TCV_RaCh := C_RACH_B_1)			
13		[TCV_Ccchg=1]			
14		(TCV_PgCh := C_PCH_B_2, TCV_agch := C_AGCH_B_2, TCV_RaCh := C_RACH_B_2)			
15		[TCV_Ccchg=2]			
16		(TCV_PgCh := C_PCH_B_3, TCV_agch := C_AGCH_B_3, TCV_RaCh := C_RACH_B_3)			
17		[TCV_Ccchg=3]			
18		(TCV_PgCh := C_PCH_B_4, TCV_agch := C_AGCH_B_4, TCV_RaCh := C_RACH_B_4)			
19		[cell = C_CellC]			
20		[TCV_Ccchg=0]			
21		(TCV_PgCh := C_PCH_C_1, TCV_agch := C_AGCH_C_1, TCV_RaCh := C_RACH_C_1)			
22		[TCV_Ccchg=1]			
23		(TCV_PgCh := C_PCH_C_2, TCV_agch := C_AGCH_C_2, TCV_RaCh := C_RACH_C_2)			
24		[TCV_Ccchg=2]			
25		(TCV_PgCh := C_PCH_C_3, TCV_agch := C_AGCH_C_3, TCV_RaCh := C_RACH_C_3)			
26		[TCV_Ccchg=3]			
27		(TCV_PgCh := C_PCH_C_4, TCV_agch := C_AGCH_C_4, TCV_RaCh := C_RACH_C_4)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SendSeqNo_chk
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective: To check whether the sending sequence number is the same
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1	E_0072	[TCV_Mt1 = TCV_Mt]		(P)	same N(SD)
2	E_0073	[TCV_Mt1 <> TCV_Mt]		(F)	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SvcSupportedChk(svc:IA5String)
Group: GSM_MS_EFR_4220/Miscellaneous/
Objective: To check wether the Basic Service svc is supported against PICS answers.
Default: OtherEvents
Comments: In PICS, if the svc is declared supported, the test case variable TCV_supported is set to TRUE, otherwisw FALSE.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_supported := C_No)			
2		[(svc = C_Telephony) AND TSPC_Serv_TS11]			TS11
3		(TCV_supported := C_Yes)			
4		[(svc = C_EmgCall) AND TSPC_Serv_TS12]			TS12
5		(TCV_supported := C_Yes)			
6		[(svc = C_AltSpchFax) AND TSPC_Serv_TS61]			TS61
7		(TCV_supported := C_Yes)			
8		[(svc = C_AutoFax) AND TSPC_Serv_TS62]			TS62
9		(TCV_supported := C_Yes)			
10		[(svc = C_Async300) AND TSPC_Serv_BS21]			BS21
11		(TCV_supported := C_Yes)			
12		[(svc = C_Async1200) AND TSPC_Serv_BS22]			BS22
13		(TCV_supported := C_Yes)			
14		[(svc = C_Async120075) AND TSPC_Serv_BS23]			BS23
15		(TCV_supported := C_Yes)			
16		[(svc = C_Async2400) AND TSPC_Serv_BS24]			BS24
17		(TCV_supported := C_Yes)			
18		[(svc = C_Async4800) AND TSPC_Serv_BS25]			BS25
19		(TCV_supported := C_Yes)			
20		[(svc = C_Async9600) AND TSPC_Serv_BS26]			BS26
21		(TCV_supported := C_Yes)			
22		[(svc = C_Sync1200) AND TSPC_Serv_BS31]			BS31
23		(TCV_supported := C_Yes)			
24		[(svc = C_Sync2400) AND TSPC_Serv_BS32]			BS32
25		(TCV_supported := C_Yes)			
26		[(svc = C_Sync4800) AND TSPC_Serv_BS33]			BS33
27		(TCV_supported := C_Yes)			
28		[(svc = C_Sync9600) AND TSPC_Serv_BS34]			BS34
29		(TCV_supported := C_Yes)			
30		[(svc = C_PAD300) AND TSPC_Serv_BS41]			BS41
31		(TCV_supported := C_Yes)			
32		[(svc = C_PAD1200) AND TSPC_Serv_BS42]			BS42
33		(TCV_supported := C_Yes)			
34		[(svc = C_PAD120075) AND TSPC_Serv_BS43]			BS43
35		(TCV_supported := C_Yes)			
36		[(svc = C_PAD2400) AND TSPC_Serv_BS44]			BS44
37		(TCV_supported := C_Yes)			
38		[(svc = C_PAD4800) AND TSPC_Serv_BS45]			BS45
39		(TCV_supported := C_Yes)			
40		[(svc = C_PAD9600) AND TSPC_Serv_BS46]			BS46
41		(TCV_supported := C_Yes)			
42		[(svc = C_Packet2400) AND TSPC_Serv_BS51]			BS51
43		(TCV_supported := C_Yes)			
44		[(svc = C_Packet4800) AND TSPC_Serv_BS52]			BS52
45		(TCV_supported := C_Yes)			
46		[(svc = C_Packet9600) AND TSPC_Serv_BS53]			BS53
47		(TCV_supported := C_Yes)			
48		[(svc = C_AltSpchData) AND TSPC_Serv_BS61]			BS61
49		(TCV_supported := C_Yes)			
50		[(svc = C_SpchData) AND TSPC_Serv_BS81]			BS81
51		(TCV_supported := C_Yes)			
52		[C_Yes]			

Detailed Comments:

Test Step Dynamic Behaviour					
Test Step Name:	Varinit_fix(activ_cell:CellID; lac:OCTETSTRING; subch:B_2; cksn:B_3; rand:RAND; arfcngsm, arfcndcs:INTEGER; ccd: CCD; imsi:HEXSTRING)				
Group:	GSM_MS_EFR_4220/Miscellaneous/				
Objective:					
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_cellid := activ_cell, TCV_ch := OC_SubchOfSdcch4(subch, activ_cell), TCV_sacch := OC_SubchOfSacch4(subch, activ_cell), TCV_ia_ts := '000'B, TCV_lac := lac, TCV_cksn :=cksn, TCV_CphKey := OC_CphKeyGen(TSPX_Ki, rand))			
2		+CCCH_group_Paging_group(ccd, imsi)			
3		+SelectPagingCh(activ_cell)			
4		[TSPC_PGSM OR TSPC_EGSM]			
5		(TCV_chdescr_arfcn:= arfcngsm)			
6		[TSPC_DCS]			
7		(TCV_chdescr_arfcn:= arfcndcs)			
Detailed Comments:					

Test Step Group SysInfo

Test Step Dynamic Behaviour					
Test Step Name:	ChgLAC_A(t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; mcc, mnc, lac:OCTETSTRING)				
Group:	GSM_MS_EFR_4220/Miscellaneous/SysInfo/				
Objective:	To change the LAC of cell A and of System Information Messages according to used testcase.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+SysInfoSending_fh(C_SCH_A, C_BCCH_A_1, t, retr, neci, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lac, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC_1, C_NCCP_2)			
2		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
3		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	ChgLAC_B(t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; mcc, mnc, lac:OCTETSTRING)				
Group:	GSM_MS_EFR_4220/Miscellaneous/SysInfo/				
Objective:	To change the LAC of cell B and of System Information Messages according to used testcase.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+SysInfoSending_fh(C_SCH_B, C_BCCH_B_1, t, retr, neci, att, babr, cch_con, bpm, t3212, C_ci_cellB, mcc, mnc, lac, CellOpt_01, CellChDes_04, CellChDes_03, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_01, BcchFreqLst_48, C_noRestablishment, C_BCC, C_NCC_1, C_NCCP_2)			
2		(TCV_Ccd0B := TCV_Ccd0H, TCV_sysinfo5_B := TCV_sysinf5, TCV_sysinfo6_B := TCV_sysinf6)			
3		+SysInfo_SacchSending(TCV_sacch_B, TCV_sysinfo5_B, TCV_sysinfo6_B)			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: ChgLAI_C(att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, par_mnc, lac:OCTETSTRING)
Group: GSM_MS_EFR_4220/Miscellaneous/SysInfo/
Objective: To change the LAI of cell C to HPLMN.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := '0010'B, TCV_Max := '00'B, TCV_Neci := '0'B, TCV_Ccd0C := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		L!DL_UdatRqSysinfo4	SysInfo4(C_BCCH_C_1, mcc, par_mnc, lac, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_noRestablishment)		
4		L!DL_UdatRqSysinfo6	SysInfo6(TCV_sacch_C, SysInf6(C_ci_cellC, mcc, par_mnc, lac, CellOpt_01, C_NCCP_2))		
5		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_C_1, C_ci_cellC, mcc, par_mnc, lac, TCV_Ccd0C, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_noRestablishment)		
6		[TSPC_DCS]			
7		L!DL_UdatRqSysinfo4	SysInfo4(C_BCCH_C_1, mcc, par_mnc, lac, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_noRestablishment)		
8		L!DL_UdatRqSysinfo6	SysInfo6(TCV_sacch_C, SysInf6(C_ci_cellC, mcc, par_mnc, lac, CellOpt_01, C_NCCP_2))		
9		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_C_1, C_ci_cellC, mcc, par_mnc, lac, TCV_Ccd0C, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_noRestablishment)		

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_cbms(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcchfldcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP; Sub:B_2)

Group: GSM_MS_EFR_4220/Miscellaneous/SysInfo/

Objective: To send system information messages for the CBMS tests.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212), TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(Sub)), 5))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+ltree_sysinfo			
		ltree_sysinfo			
5		[TSPC_PGSM OR TSPC_EGSM]			
6		L!DL_UdatRqSysinfo4	SysInfo4_CBMS(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re, ChDescrp_nfh(TCV_Tchtype, C_S0, C_BCC, C_arfcnA))		
7		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
8		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
9		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
10		(TCV_sysinf5 := SysInf5(bcchflgsm))			
11		[TSPC_DCS]			
12		L!DL_UdatRqSysinfo4	SysInfo4_CBMS(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re, ChDescrp_nfh(TCV_Tchtype, C_S0, C_BCC, C_arfcnAd))		
13		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		
14		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
15		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
16		(TCV_sysinf5 := SysInf5(bcchfldcs))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_fh(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm, cchddcs:CCHD; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcchfldcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP)

Group: GSM_MS_EFR_4220/Miscellaneous/SysInfo/

Objective: To send system information messages for the L3 tests. The following parameters specified by input parameters:
- bcch: BCCH
- sch: SCH
- bcchfl: BCCH frequency list
- ci: Cell identity
- lai: Location area identification
- ccd: Control channel description
- csp: Cell selection parameters
- cchd: Cell channel description
- rachcpar: RACH controll paramenters
- nccp: NCC permitted

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+ltree_sysinfo			
		ltree_sysinfo			
5		[TSPC_PGSM OR TSPC_EGSM]			
6		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchdgsm, TCV_Max, TCV_Tx, Re)		
7		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
8		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
9		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
10		(TCV_sysinf5 := SysInf5(bcchflgsm))			
11		[TSPC_DCS]			
12		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchddcs, TCV_Max, TCV_Tx, Re)		
13		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		
14		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
15		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
16		(TCV_sysinf5 := SysInf5(bcchfldcs), TCV_sysinf6.co.pwrc := '1'B)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_nfh(sch, bcch:LOGICCH; t, retr, neci, att: INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP)

Group: GSM_MS_EFR_4220/Miscellaneous/SysInfo/

Objective: To send system information messages with default parameters defined for the L3 tests for which no special parameters indicated.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+gsmOrDcs			
		gsmOrDcs			
5		[TSPC_PGSM OR TSPC_EGSM]			1.
6		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
7		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
8		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
9		[NOT TSPX_AltNb]			3.
10		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
11		(TCV_sysinf5 := SysInf5(bcchflgsm))			
12		[TSPX_AltNb]			4.
13		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcflaltgsm, TCV_Max, TCV_Tx, Re, nccp)		
14		(TCV_sysinf5 := SysInf5(bcflaltgsm))			
15		[TSPC_DCS]			2.
16		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
17		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
18		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
19		[NOT TSPX_AltNb]			3.
20		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		

21		(TCV_sysinf5 := SysInf5(bcchfldcs))			
22		[TSPX_AltNb]			4.
23		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcflaltdcs, TCV_Max, TCV_Tx, Re, nccp)		
24		(TCV_sysinf5 := SysInf5(bcflaltdcs))			

Detailed Comments:

1. For GSM900 mobile station testing.
2. For DCS1800 mobile station testing.
3. To use alternative neighbour cells description.
4. To use default neighbour cells description.

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_nfhCb(sch, bcch:LOGICCH; t, retr, neci, att: INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP)

Group: GSM_MS_EFR_4220/Miscellaneous/SysInfo/

Objective: To send system information messages with default parameters defined for the L3 tests for which no special parameters indicated.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+gsmOrDcs			
		gsmOrDcs			
5		[TSPC_PGSM OR TSPC_EGSM]			1.
6		L!DL_UdatRqSysinfo4 (DL_UdatRqSysinfo4.msg.rachcp.cba := '1'B)	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
7		L!DL_UdatRqSysinfo1_nh (DL_UdatRqSysinfo1_nh.msg.rachcp.cba := '1'B)	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
8		L!DL_UdatRqSysinfo3 (DL_UdatRqSysinfo3.msg.rachcp.cba := '1'B)	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
9		[NOT TSPX_AltNb]			3.
10		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.rachcp.cba := '1'B)	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
11		(TCV_sysinf5 := SysInf5(bcchflgsm))			
12		[TSPX_AltNb]			4.
13		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.rachcp.cba := '1'B)	SysInfo2(bcch, bcflaltgsm, TCV_Max, TCV_Tx, Re, nccp)		
14		(TCV_sysinf5 := SysInf5(bcflaltgsm))			
15		[TSPC_DCS]			2.
16		L!DL_UdatRqSysinfo4 (DL_UdatRqSysinfo4.msg.rachcp.cba := '1'B)	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
17		L!DL_UdatRqSysinfo1_nh (DL_UdatRqSysinfo1_nh.msg.rachcp.cba := '1'B)	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
18		L!DL_UdatRqSysinfo3 (DL_UdatRqSysinfo3.msg.rachcp.cba := '1'B)	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
19		[NOT TSPX_AltNb]			3.
20		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.rachcp.cba := '1'B)	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		

21		(TCV_sysinf5 := SysInf5(bcchfldcs))			
22		[TSPX_AltNb]			4.
23		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.rachcp.cba := '1'B)	SysInfo2(bcch, bcflaltdcs, TCV_Max, TCV_Tx, Re, nccp)		
24		(TCV_sysinf5 := SysInf5(bcflaltdcs))			

Detailed Comments:
1. For GSM900 mobile station testing.
2. For DCS1800 mobile station testing.
3. To use alternative neighbour cells description.
4. To use default neighbour cells description.

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_e(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm:CCHD; crh, mtmcgsm:INTEGER; bcchflgsm, bcchflegsm:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP)

Group: GSM_MS_EFR_4220/Miscellaneous/SysInfo/

Objective: To send SYNCHRONIZATION CHANNEL INFORMATION and SYSTEM INFORMATION messages with default parameters except CCCH_CONF, Tx-integer, Max-Retrans and ATT which are specified by input parameters.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		1.
3		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
4		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchdgsm, TCV_Max, TCV_Tx, Re)		2.
5		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.bcchfl.extind := '1'B)	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
6		L!DL_UdatRqSysinfo2bis	SysInfo2bis(bcch, bcchflegsm)		
7		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
8		(TCV_sysinf5 := SysInf5(bcchflgsm), TCV_sysinfo5.bcchfl.extind := '1'B)			
9		(TCV_sysinfo5bis := SysInf5bis(bcchflegsm))			
10		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		

Detailed Comments:
1. To send SYNCHRONIZATION INFORMATION message with default parameter for cell A.
2. System Information Type 1 with Cell Channel Description for HO-testin in cell A (GSM900).

Test Step Dynamic Behaviour					
Test Step Name:	SysInfo_SacchSending(ch:LOGICCH; sysinfo5_pdu:SYSINFO5_PDU; sysinfo6_pdu:SYSINFO6_PDU)				
Group:	GSM_MS_EFR_4220/Miscellaneous/SysInfo/				
Objective:	To send SYSTEM INFORMATION 5 and 6 messages defined by parameters 'sysinfo5_pdu' and 'sysinfo6_pdu' in the parametrised 'ch' channel.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[ch <> "dummy"]			
2		L!DL_UdatRqSysinfo6	SysInfo6(ch, sysinfo6_pdu)		
3		L!DL_UdatRqSysinfo5	SysInfo5(ch, sysinfo5_pdu)		
4	E_0074	[ch = "dummy"]		I	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	SysInfo_5bisSending(ch:LOGICCH; sysinfo5bis_pdu:SYSINFO5bis_PDU)				
Group:	GSM_MS_EFR_4220/Miscellaneous/SysInfo/				
Objective:	To send SYSTEM INFORMATION 5bis message defined by parameters 'sysinfo5bis_pdu' in the parametrised 'ch' channel.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[ch <> "dummy"]			
2		L!DL_UdatRqSysinfo5bis	SysInfo5bis(ch, sysinfo5bis_pdu)		
3	E_0075	[ch = "dummy"]		I	
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: SetNECI(att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING)
Group: GSM_MS_EFR_4220/Miscellaneous/SysInfo/
Objective: To set the NECI =1.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := '0010'B, TCV_Max := '11'B, TCV_Neci := '1'B, TCV_Ccd0A := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		L!DL_UdatRqSysinfo4	SysInfo4(C_BCCH_A_1, mcc, mnc, lac, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
4		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
5		[TSPC_DCS]			
6		L!DL_UdatRqSysinfo4	SysInfo4(C_BCCH_A_1, mcc, mnc, lac, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
7		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SetATT(t, retr, att, neci:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING)
Group: GSM_MS_EFR_4220/Miscellaneous/SysInfo/
Objective: To set the ATT flag to "MS's in the cell should apply IMSI attach and detach procedure"
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0A := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		[TSPC_PGSM OR TSPC_EGSM]			1.
3		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
4		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
5		START T_dly(5000)			
6		?TIMEOUT T_dly			3.
7		[TSPC_DCS]			2.
8		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
9		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
10		START T_dly(5000)			3.
11		?TIMEOUT T_dly			

Detailed Comments:
1. For P-GSM900.
2. For DCS1800.
3. Wait for 5 seconds to allow the MS read BCCH information.

Test Step Group OperatorOP

Test Step Dynamic Behaviour					
Test Step Name:	InitCall(srv: SERVICES)				
Group:	GSM_MS_EFR_4220/OperatorOP/				
Objective:	To initiate a call of the basic service `srv`.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OO_InitCall(srv))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	InitNonCallSupp				
Group:	GSM_MS_EFR_4220/OperatorOP/				
Objective:	To attempt a non call related supplementary service at the MS under test.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OO_InitNonCallSS())			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	AtmpShortMsg				
Group:	GSM_MS_EFR_4220/OperatorOP/				
Objective:	To attempt a short message service transaction at the MS under test.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OO_SendMOShortMessage())			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	CheckUssdStringDisplayed(strg: IA5String)				
Group:	GSM_MS_EFR_4220/OperatorOP/				
Objective:	To check whether the correct USSD String 'strg' is displayed on the MS				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Res := OO_CheckUssdStringDisplayed(strg))			
2	E_0076	[TCV_Res]		(P)	
3	E_0077	[NOT TCV_Res]		(F)	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	PLMNsCHK				
Group:	GSM_MS_EFR_4220/OperatorOP/				
Objective:	To check whether the MS presents a list of available PLMNs.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Res := OO_PLMNsCHK())			
2	E_0078	[TCV_Res]		P	
3	E_0079	[NOT TCV_Res]		F	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	RFtransCHK				
Group:	GSM_MS_EFR_4220/OperatorOP/				
Objective:	To check whether the MS transmits any radio signal.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Res := OO_RFoutputCHK())			
2	E_0080	[TCV_Res]		F	
3	E_0081	[NOT TCV_Res]		(P)	
Detailed Comments:					

Test Step Group Postambles

Test Step Dynamic Behaviour					
Test Step Name:	PostLinkRelEnd(ch: LOGICCH)				
Group:	GSM_MS_EFR_4220/Postambles/				
Objective:	To release the RR connection and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(ch, ChRelease_01)		
2		L?DL_RelIn	DLRelInd_01		
3	E_0082	CANCEL		R	1.
4		?TIMEOUT T_release			
5	E_0083	CANCEL		R	1.
Detailed Comments:		1. Cancel of all running timers and final verdict.			

Test Step Dynamic Behaviour					
Test Step Name:	PostMainLinkRel(chnl:LOGICCH)				
Group:	GSM_MS_EFR_4220/Postambles/				
Objective:	To release the main signalling link `ch`, and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(chnl, ChRelease_01)		
2		L?DL_RelIn CANCEL T_release	DLRelInd_01		
3		?TIMEOUT T_release			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	ChanRel(ch: LOGICCH)				
Group:	GSM_MS_EFR_4220/Postambles/				
Objective:	To release the RR connection on the channel TCV_ch and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(ch, ChRelease_01)		
2		L?DL_RelIn CANCEL T_release	DLRelInd_01		
3	E_0084	?TIMEOUT T_release		(F)	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	ChanRel_P(ch: LOGICCH)				
Group:	GSM_MS_EFR_4220/Postambles/				
Objective:	To release the RR connection on the channel TCV_chmaindcch and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(ch, ChRelease_01)		
2	E_0085	L?DL_RelIn	DLRelInd_01	(P)	
3		?TIMEOUT T_release			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	ChanRel_end(ch: LOGICCH)				
Group:	GSM_MS_EFR_4220/Postambles/				
Objective:	To release the RR connection and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(ch, ChRelease_01)		
2		L?DL_RelIn	DLRelInd_01		
3	E_0086	CANCEL		(P)	1.
4		?TIMEOUT T_release			
5	E_0087	CANCEL		(F)	1.
Detailed Comments:		1. Cancel of all running timers.			

Test Step Group Preambles

Test Step Dynamic Behaviour					
Test Step Name:	Chmod(rate:RATE; ce:B_2; ur:B_4; svc:SERVICES)				
Group:	GSM_MS_EFR_4220/Preambles/				
Objective:	To assign correct values to the test case variables TCV_ChMod and TCV_ChModb.				
Default:	OtherEvents				
Comments:	The rules for assignment are in Table 10-1 of GSM 11.10 section 10.3. If the value for ce is "BothT", the rule for transparent is used. If the value for ce is "BothNT", the rule for non transparent is used. That means the preferred value will be used in test configuration.				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+local_tree			
2		[(svc = C_AltSpchData) OR (svc = C_SpchData) OR (svc = C_AltSpchFax)]			dual mode call
3		(TCV_ChMod.mode := C_ChMod_r)			
4		[(svc<> C_AltSpchData) AND(svc <> C_SpchData) AND(svc <> C_AltSpchFax)]			
		local_tree			
5		[((ce = C_notransparent) OR (ce = C_BothNT)) AND (rate = C_Full)]			
6		(TCV_ChMod.mode := C_ChMod_12k, TCV_ChModb.mode := C_ChMod_12k)			
7		[((ce = C_notransparent) OR (ce = C_BothNT)) AND (rate = C_Half)]			
8		[ur <> C_9600bs]			
9		(TCV_ChMod.mode := C_ChMod_6k, TCV_ChModb.mode := C_ChMod_6k)			
10	E_0088	[ur = C_9600bs]		I	not applicable
11		[(ce = C_transparent) OR (ce = C_BothT)]			
12		[(ur = C_9600bs) AND (rate = C_Full)]			
13		(TCV_ChMod.mode := C_ChMod_12k, TCV_ChModb.mode := C_ChMod_12k)			
14		[ur = C_4800bs]			
15		(TCV_ChMod.mode := C_ChMod_6k, TCV_ChModb.mode := C_ChMod_6k)			
16		[(ur <> C_4800bs) AND(ur <> C_9600bs)]			
17		(TCV_ChMod.mode := C_ChMod_3k, TCV_ChModb.mode := C_ChMod_3k)			
18	E_0089	[(ur = C_9600bs) AND (rate = C_Half)]		I	not applicable
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: EFR_BasicServiceMO(svc:SERVICES; rate:RATE)

Group: GSM_MS_EFR_4220/Preambles/

Objective: To generate a EFR Bcap and a MO SETUP/ESETUP message (to match later against received MO SETUP/ESETUP message), and an appropriate channel mode based on the service and rate, as well as on several IXIT parameters, itc, ce, ur, synchronous bit for an MO call.

Bcap shall match the correct NUMBER of speech versions that the EFR mobile supports.

Default: OtherEvents

Comments: Used for CC tests

Nr	Label	Behaviour Description	CRef	V	Comments
1		+local_init			Default values
2		[TCV_Service = C_EmgCall]			
3		(TCV_Ecall := C_Yes)			Emergency Call
4		[(TCV_ChRate = C_Full) OR (TCV_ChRate = C_Half)]			
5		(TCV_ChMod.mode := C_ChMod_r)			Speech v1
6		[NOT TSPC_EFR_EmgCallBcap]			
7		(TCV_Esetup := ESetupNoBcap)			Speech v1 - no bcap
8		[TSPX_EFR_SpchPrefsNo = 2]			
9		(TCV_Esetup := ESetupBcap(EFR_Bcap_Speech_MO2(TCV_Rchr)))			Speech v1 - 2 speech prefs
10		[TSPX_EFR_SpchPrefsNo = 3]			
11		(TCV_Esetup := ESetupBcap(EFR_Bcap_Speech_MO3(TCV_Rchr)))			Speech v1 - 3 speech prefs
12	E_0090	[C_Yes]		I	IXIT error
13		[TCV_ChRate = C_Full_v2]			
14		(TCV_ChMod.mode := C_ChMod_r2,TCV_ChRate := C_Full)			Speech v2
15		[NOT TSPC_EFR_EmgCallBcap]			
16		(TCV_Esetup := ESetupNoBcap)			Speech v2 - no bcap
17		[TSPX_EFR_SpchPrefsNo = 2]			
18		(TCV_Esetup := ESetupBcap(EFR_Bcap_Speech_MO2(TCV_Rchr)))			Speech v2 - 2 speech prefs
19		[TSPX_EFR_SpchPrefsNo = 3]			
20		(TCV_Esetup := ESetupBcap(EFR_Bcap_Speech_MO3(TCV_Rchr)))			Speech v2 - 3 speech prefs
21	E_0091	[C_Yes]		I	IXIT error
22		[TCV_Service = C_Telephony]			
23		[(TCV_ChRate = C_Full) OR (TCV_ChRate = C_Half)]			Telephony
24		(TCV_ChMod.mode := C_ChMod_r)			Speech v1
25		[TSPX_EFR_SpchPrefsNo = 2]			
26		(TCV_Setup_mo := Setup_02(EFR_Bcap_Speech_MO2(TCV_Rchr)))			Speech v1 - 2 speech prefs
27		[TSPX_EFR_SpchPrefsNo = 3]			
28		(TCV_Setup_mo := Setup_02(EFR_Bcap_Speech_MO3(TCV_Rchr)))			Speech v1 - 3 speech prefs
29	E_0092	[C_Yes]		I	IXIT error
30		[TCV_ChRate = C_Full_v2]			
31		(TCV_ChMod.mode := C_ChMod_r2, TCV_ChRate := C_Full)			Speech v2
32		[TSPX_EFR_SpchPrefsNo = 2]			
33		(TCV_Setup_mo := Setup_02(EFR_Bcap_Speech_MO2(TCV_Rchr)))			Speech v2 - 2 speech prefs
34		[TSPX_EFR_SpchPrefsNo = 3]			
35		(TCV_Setup_mo := Setup_02(EFR_Bcap_Speech_MO3(TCV_			Speech v2 - 3 speech prefs

		Rchr)))			
36	E_0093	[C_Yes]		I	IXIT error
37	E_0094	[C_Yes]		I	IXIT error
		local_init			
38		(TCV_Service := svc, TCV_Rchr := C_Rchr_Full, TCV_Ecall := C_No, TCV_ChRate :=rate, TCV_ChMod.iei := '01100011'B, TCV_ChMod.mode := C_ChMod_12k, TCV_ChModb.iei := '01100011'B, TCV_ChModb.mode := C_ChMod_12k, TCV_MODEM := V22bisOrV26ter)			
39		[TSPC_DualRate]			
40		(TCV_Rchr := C_Rchr_DualForDualH)			
41		[NOT TSPC_DualRate]			

Detailed Comments: This test step is called in the preamble of test cases where an MO call is established in the test body.

Test Step Dynamic Behaviour

Test Step Name: EFR_BasicServiceMT(svc:SERVICES; rate:RATE)

Group: GSM_MS_EFR_4220/Preambles/

Objective: To generate a EFR Bcap (to match against received CALL CONNECTED Bcap), an MT SETUP or ESETUP message (to be sent to EFR mobile) and an appropriate channel mode based on the service and rate, as well as on several IXIT parameters, itc, ce, ur, synchronous bit for an MT call.

EFR Bcap shall match the correct NUMBER of speech versions that the EFR mobile supports.

Default: OtherEvents

Comments: Used for CC tests

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Service := svc, TCV_ChRate :=rate, TCV_ChMod.iei := '01100011'B, TCV_ChMod.mode := C_ChMod_12k, TCV_ChModb.iei := '01100011'B, TCV_ChModb.mode := C_ChMod_12k, TCV_Rchr := C_Rchr_Full, TCV_2ndtest := C_No)			Default values
2		[TSPC_DualRate]			
3		(TCV_Rchr := C_Rchr_DualForDualH)			
4		+SvcSupportedChk(svc)			
5		+local_main			
6		[NOT TSPC_DualRate]			
7		+SvcSupportedChk(svc)			
8		+local_main			
		local_main			
9		[TCV_Service = C_Telephony]			
10		[TCV_ChRate = C_Full]			Speech v1
11		(TCV_ChMod.mode := C_ChMod_r, TCV_Setup_mt := Setup_01(TI_02))			
12		[TSPX_EFR_SpchPrefsNo = 2]			
13		(TCV_BcapMO1 := EFR_Bcap_Speech_MO2(TCV_Rchr))			EFR BCAP - 2 speech prefs
14		+local_IFsetup_Telephony			Config speech
15		[TSPX_EFR_SpchPrefsNo = 3]			
16		(TCV_BcapMO1 := EFR_Bcap_Speech_MO3(TCV_Rchr))			EFR BCAP - 3 speech prefs
17		+local_IFsetup_Telephony			Config speech
18	E_0095	[C_Yes]		I	IXIT error
19		[TCV_ChRate = C_Full_v2]			Speech v2
20		(TCV_ChRate := C_Full,TCV_ChMod.mode := C_ChMod_r2,TCV_Setup_mt := Setup_01(TI_02))			
21		[TSPX_EFR_SpchPrefsNo = 2]			
22		(TCV_BcapMO1 := EFR_Bcap_Speech_MO2(TCV_Rchr))			EFR BCAP - 2 speech prefs
23		+local_IFsetup_Telephony			Config speech
24		[TSPX_EFR_SpchPrefsNo = 3]			
25		(TCV_BcapMO1 := EFR_Bcap_Speech_MO3(TCV_Rchr))			EFR BCAP - 3 speech prefs
26		+local_IFsetup_Telephony			Config speech
27	E_0096	[C_Yes]		I	IXIT error
		local_IFsetup_Telephony			
28		[TCV_supported]			
29		(TCV_Null := OO_IFsetup_Telephony())			
30		[NOT TCV_supported]			
31		[TCV_SpecialCase]			
32		START T_dly(C_T_Wait)			
33		?TIMEOUT T_dly			
34	E_0097	[NOT TCV_SpecialCase]		I	parameter error

Detailed Comments: This test step is called in the preamble of test cases where an MT call is established in the test body.

Test Step Dynamic Behaviour

Test Step Name:	Est_MO_Call_init(par1: OCTETSTRING; mag, mad:MA; ta:TA; tsc1:B_3; nfrqg, nfrqd:INTEGER)
Group:	GSM_MS_EFR_4220/Preambles/
Objective:	To initiate a mobile originating call for the supported bearer capability. The channel in use is frequency hopping channel.
Default:	OtherEvents
Comments:	The test case variable TCV_Mt holds the message type of the received SETUP msg.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_cellid:=C_CellA)			
2		+InitCall(TCV_Service)			
3		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_15)		
4		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
5		+localtree_ia(par1)			
6		L?DL_EstInCmsRq	CMSerReq(CMServiceReq_01)		
7		ACTIVATE(OtherEvents)			Restore Normal default
8		+ltree_continue			
		ltree_continue			
9		(TCV_Null := OM_NoL2Ack(2,TCV_ch))			Don't acknowledge setup message
10		+Ciphering_on(TCV_ch)			
11		L?DL_DatInSetup (TCV_Mt := DL_DatInSetup.msg.mt)	SetupRcv(SetupInd_01)		
		localtree_ia(par_chn: OCTETSTRING)			
12		[par_chn = C_CHTCHF_NonFH]			
13		(TCV_Tchtype := '00001'B)			
14		+ltree_immass_nfh(TCV_tch_arfcn, tsc1)			
15		[par_chn = C_CHSDCCH4_NonFH]			
16		(TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubDef)), 5))			
17		+ltree_immass_nfh(TCV_chdescr_arfcn, tsc1)			
18		[par_chn = C_CHSDCCH8_NonFH]			
19		(TCV_Tchtype := INT_TO_BIT((8 + BIT_TO_INT(TSPX_SDCCH8SubDef)), 5))			
20		+ltree_immass_nfh(TCV_tch_arfcn, tsc1)			
21		[par_chn = C_CHTCHF_FH]			
22		(TCV_Tchtype := '00001'B)			
23		+ltree_immass_fh(nfrqg, nfrqd)			
24		[par_chn = C_CHTCHH_FH]			
25		(TCV_Tchtype := INT_TO_BIT((2 + BIT_TO_INT(TSPX_TCHHSubDef)), 5))			
26		+ltree_immass_fh(nfrqg, nfrqd)			
27		[par_chn = C_CHTCHH_NonFH]			
28		(TCV_Tchtype := INT_TO_BIT((2 + BIT_TO_INT(TSPX_TCHHSubDef)), 5))			
29		+ltree_immass_nfh(TCV_tch_arfcn, tsc1)			
30		[par_chn = C_CHSDCCH8_FH]			
31		(TCV_Tchtype := INT_TO_BIT((8 + BIT_TO_INT(TSPX_SDCCH8SubDef)), 5))			
32		+ltree_immass_fh(nfrqg, nfrqd)			
		ltree_immass_fh(frqnog, frqnod:INTEGER)			
33		[TSPC_PGSM OR TSPC_EGSM]			
34		(TCV_maio := INT_TO_BIT((TSPX_MAIO MOD frqnog), 6), TCV_iel := mag.iel)			1.
35		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_221(TCV_Rr, TCV_Fn, TCV_ia_ts, tsc1, TCV_Tchtype, TCV_maio, INT_TO_BIT(TSPX_HSN, 6), ta, mag, TCV_iel))		

36		[TSPC_DCS]			
37		(TCV_maio := INT_TO_BIT((TSPX_MAIO MOD frqnod), 6), TCV_iel := mad.iel)			1.
38		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_221(TCV_Rr, TCV_Fn, TCV_ia_ts, tsc1, TCV_Tchtype, TCV_maio, INT_TO_BIT(TSPX_HSN, 6), ta, mad, TCV_iel))		
		ltree_immass_nfh(arfcn:INTEGER; tsc:B_3)			
39		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_nfh(TCV_Rr, TCV_Fn, TCV_Tchtype, TCV_ia_ts, tsc, ta, arfcn, C_normal_paging))		

Detailed Comments: 1. To make the MAIO is within 0 ... N-1, where N is the number of frequencies of MA. "frqnog" and "frqnod" are number of of frequencies of MA for GSM and DCS respectively.

Test Step Dynamic Behaviour

Test Step Name: Est_MT_Call_FH(ta:TA; rate:IA5String; maio:MAIO; hsn:INTEGER; rand:RAND; slot:SN; tsc:TSC; tch:B_1)

Group: GSM_MS_EFR_4220/Preambles/

Objective: To establish a mobile station terminating call with hopping channel (speech or data call).

Default: OtherEvents

Comments: used var's:
TCV_cellid, TCV_ch

Nr	Label	Behaviour Description	CRef	V	Comments
1		+RRmtcallprepare(ta, rand)			
2		+CC_Est_MT_Call			
3		+localtree			
4		+AssCh_complete(TCV_ch, TCV_chTch, TCV_AssCmd)			
5		L!DL_DatRqConnAck	ConnAckSnd(TCV_chTch, ConnectAck_01(TI_02))		
		localtree			
6		[rate = C_Full]			Full rate
7		[TCV_cellid = C_CellA]			
8		+AssCmdGen_fh(C_Full, slot, tsc, tch, 9, 15, maio, INT_TO_BIT(hsn, 6), Frql_20_A, Frql_20_Ad, CellChDes_Omit, CellChDes_Omit, ChMod_speech_iei, MoblAllc_omit, MoblAllc_omit, CphMod_omit)			
9		[TCV_cellid = C_CellB]			
10		+AssCmdGen_fh(C_Full, slot, tsc, tch, 9, 15, maio, INT_TO_BIT(hsn, 6), Frql_20_B3, Frql_20_B3d, CellChDes_Omit, CellChDes_Omit, ChMod_speech_iei, MoblAllc_omit, MoblAllc_omit, CphMod_omit)			
11		[rate = C_Full_v2]			Full rate
12		[TCV_cellid = C_CellA]			
13		+AssCmdGen_fh(C_Full_v2, slot, tsc, tch, 9, 15, maio, INT_TO_BIT(hsn, 6), Frql_20_A, Frql_20_Ad, CellChDes_Omit, CellChDes_Omit, ChMod_EFR2_iei, MoblAllc_omit, MoblAllc_omit, CphMod_omit)			
14		[TCV_cellid = C_CellB]			
15		+AssCmdGen_fh(C_Full_v2, slot, tsc, tch, 9, 15, maio, INT_TO_BIT(hsn, 6), Frql_20_B3, Frql_20_B3d, CellChDes_Omit, CellChDes_Omit, ChMod_EFR2_iei, MoblAllc_omit, MoblAllc_omit, CphMod_omit)			
16		[rate = C_Half]			Half rate
17		[TCV_cellid = C_CellA]			
18		+AssCmdGen_fh(C_Half, slot, tsc, tch, 9, 15, maio, INT_TO_BIT(hsn, 6), Frql_20_A0, Frql_20_A0d, CellChDes_Omit, CellChDes_Omit, ChMod_omit, MoblAllc_omit, MoblAllc_omit, CphMod_omit)			
19		[TCV_cellid = C_CellB]			
20		+AssCmdGen_fh(C_Half, slot, tsc, tch, 9, 15, maio, INT_TO_BIT(hsn, 6), Frql_20_B3, Frql_20_B3d, CellChDes_Omit, CellChDes_Omit, ChMod_omit, MoblAllc_omit, MoblAllc_omit, CphMod_omit)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Est_MT_CallNonFH(ta:TA; rate:IA5String; sub:B_1; rand:RAND)
Group: GSM_MS_EFR_4220/Preambles/
Objective: To establish a mobile station terminating call with non hopping channel (speech or data call).
Default: OtherEvents
Comments: used var's:
TCV_cellid, TCV_chdescr_arfcn, TCV_Bcap1, TCV_chTch, TCV_asscmd_ts

Nr	Label	Behaviour Description	CRef	V	Comments
1		+RRmtcallprepare(ta, rand)			
2		+CC_EstMsTermCall(rate, sub)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: IdleUpdated(par_bspwr:INTEGER; activ_cell:CellID; sch, bcch:LOGICCH; acttype:BITSTRING; slot, slot2, slot3, slot4:SN; tsc:TSC; t, retr, neci, att:INTEGER; ta:TA; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm, cchddcs:CCHD; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; StartType:INTEGER; subch:B_2; cksn:B_3; rand:RAND; arfcngsm, arfcndcs:INTEGER; imsi:HEXSTRING; sysinfonb:INTEGER; nccp:NCCP)

Group: GSM_MS_EFR_4220/Preambles/

Objective: To ensure that the SIM is either updated to the initial conditions and the MS with CKSN valid, TMSI valid and idle updated in cell A, B or C; or idle updated without TMSI; or totally not updated and all LAI information deleted.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_slot := slot, TCV_tsc := tsc, TCV_Ccd0H := CntrlChDscrp(att, babr, C_cch_1Comb, bpm, t3212))			
2		+MM_PwrOrSimOff(C_SIMneedRmv)			
3		+Varinit_fix(activ_cell, lac, subch, cksn, rand, arfcngsm, arfcndcs, TCV_Ccd0H, imsi)			
4		+ltree_chconf(C_cch_1Comb)			Config LT
5		+localtree1			
		localtree1			
6		+ltree_sysinfo(1, C_cch_1Comb)			ATT= '1'B
7		+MM_PwrOrSimOn(C_SIMneedRmv)			
8		+localtree_Lup_Auth			
		localtree_Lup_Auth			
9		START T_dly(C_T_Wait1stChReq)			
10		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_02)		Any channel request PDU
11		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
12		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, subch, slot, tsc, TCV_chdescr_arfcn, ta))		
13		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(C_norm_period_attach))		Any location update type
14		ACTIVATE(OtherEvents)			Restore Normal default
15		[StartType = C_Start_Rej]			
16		L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(C_rc_LAnotallowed))		Delete LAI, tmsi, cksn
17		+PostMainLinkRel(TCV_ch)			Release Ch
18		+MM_PwrOrSimOff(C_SIMneedRmv)			
19		+localtree2			
20		[StartType = C_Start_Imsi]			
21		+Authentication(TCV_ch, cksn, rand)			
22		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiImsi_01iei, mcc, mnc, TCV_lac))		Location Accept with IMSI
23		+PostMainLinkRel(TCV_ch)			Release Ch
24		+localtree2			
25		[(StartType = C_Start_Tmsi) OR (StartType = C_Start_TmsiOff)]			
26		+Authentication(TCV_ch, cksn, rand)			
27		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiTmsi_01iei, mcc, mnc, TCV_lac))		Location Accept with TMSI
28		L?DL_DatInTmsireCom	TmsiReallocCmp(TCV_ch)		
29		+PostMainLinkRel(TCV_ch)			Release Ch

30		+localtree2			
31		[StartType = C_Start_PLMNnot]			
32		L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(C_rc_plmn_not))		Delete LAI, tmsi, cksn
33		+PostMainLinkRel(TCV_ch)			Release Ch
34		+localtree2			
35	E_0098	[(StartType <> C_Start_Rej) AND(StartType <> C_Start_Tmsi) AND(StartType <> C_Start_TmsiOff) AND(StartType <> C_Start_Imsi) AND(StartType <> C_Start_PLMNnot)]		I	
36		?TIMEOUT T_dly			
37		+ltree_rmvPLMNfbdnlist			
		localtree2			
38		[(att = 1) AND(StartType <> C_Start_TmsiOff)]			ATT no change
39		[cch_con <> C_cch_1Comb]			If non combined
40		+localtree3			
41		+ltree_PgReorg			
42		[cch_con = C_cch_1Comb]			If combined
43		[(att = 1) AND (StartType = C_Start_TmsiOff)]			
44		+ImsiDetach(MiTmsi_01, ta, C_SIMneedRmv)			Switch off, detach
45		[cch_con <> C_cch_1Comb]			If non combined
46		+localtree3			
47		[cch_con = C_cch_1Comb]			If combined
48		[att = 0]			
49		[cch_con = C_cch_1Comb]			If combined
50		+ltree_sysinfo(0, C_cch_1Comb)			ATT=0
51		+ltree_PgReorg			
52		[cch_con <> C_cch_1Comb]			If non combined
53		+localtree3			
54		+ltree_PgReorg			
		localtree3			
55		(TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
56		+Varinit_fix(activ_cell, lac, subch, cksn, rand, arfcngsm, arfcndcs, TCV_Ccd0H, imsi)			
57		+ltree_chconf(cch_con)			Config LT
58		+ltree_sysinfo(att, cch_con)			
		ltree_PgReorg			
59		(TCV_Null:=OM_PgFill(activ_cell, PgReqTp1Reorg))			
60		START T_dly(5000)			
61		?TIMEOUT T_dly			
62		(TCV_Null:=OM_PgFill(activ_cell, PgReqTp1Norm))			
		ltree_sysinfo(attach:INTEGER; con_ch:B_3)			
63		[activ_cell = C_CellA]			
64		+SysInfoSending(attach, con_ch)			SI1-4, cellA
65		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
66		[con_ch = C_cch_1Comb]			If combined
67		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			SI5, 6 on SACCH/C4, cell A
68		[con_ch <> C_cch_1Comb]			
69		+ltree_sysinfo_other(attach)			More SIs, cellA
70		[activ_cell = C_CellB]			
71		+SysInfoSending(attach, con_ch)			SI1-4, cellB
72		(TCV_Ccd0B := TCV_Ccd0H, TCV_sacch_B := TCV_sacch, TCV_sysinfo5_B := TCV_sysinf5, TCV_sysinfo6_B := TCV_sysinf6)			SI5, 6 on SACCH/C4, cell B
73		[con_ch = C_cch_1Comb]			If combined
74		+SysInfo_SacchSending(TCV_sacch_B, TCV_sysinfo5_B, TCV_sysinfo6_B)			SI5, 6 on SACCH/C4, cell A

75		[con_ch <> C_cch_1Comb]			No more SIs, cell B
76		[activ_cell = C_CellC]			
77		+SysInfoSending(attach, con_ch)			SI1-4, cellC
78		(TCV_Ccd0C := TCV_Ccd0H, TCV_sacch_C := TCV_sacch, TCV_sysinfo5_C := TCV_sysinf5, TCV_sysinfo6_C := TCV_sysinf6)			
79		[con_ch = C_cch_1Comb]			If combined
80		+SysInfo_SacchSending(TCV_sacch_C, TCV_sysinfo5_C, TCV_sysinfo6_C)			SI5, 6 on SACCH/C4, cell C
81		[con_ch <> C_cch_1Comb]			No more SIs, cell C
		SysInfoSending(attachb:INTEGER; con_chs:B_3)			
82		[sysinfonb = C_Test_fh]			Hopping
83		+SysInfoSending_fh(sch, bcch, t, retr, neci, attachb, babr, con_chs, bpm, t3212, ci, mcc, mnc, lac, co, cchdgsm, cchddcs, crh, mtmcgsm, mtmcdcs, bcchflgsm, bcchfldcs, Re, bcc, ncc, nccp)			
84		[sysinfonb = C_Test_nfh]			Non hopping
85		+SysInfoSending_nfh(sch, bcch, t, retr, neci, attachb, babr, con_chs, bpm, t3212, ci, mcc, mnc, lac, co, crh, mtmcgsm, mtmcdcs, bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs, Re, bcc, ncc, nccp)			
86		[sysinfonb = C_Test_egsm]			EGSM
87		+SysInfoSending_e(sch, bcch, t, retr, neci, attachb, babr, con_chs, bpm, t3212, ci, mcc, mnc, lac, co, cchdgsm, crh, mtmcgsm, bcchflgsm, bcchfldcs, Re, bcc, ncc, nccp)			
88		[sysinfonb = C_Test_cbms]			CBMS
89		+SysInfoSending_cbms(sch, bcch, t, retr, neci, attachb, babr, con_chs, bpm, t3212, ci, mcc, mnc, lac, co, crh, mtmcgsm, mtmcdcs, bcchflgsm, bcchfldcs, Re, bcc, ncc, nccp, subch)			
		ltree_chconf(conf:B_3)			
90		[activ_cell = C_CellA]			
91		+ltree_chconf_A(conf)			Config LT
92		[activ_cell = C_CellB]			
93		+ltree_chconf_B(conf)			Config LT
94		[activ_cell = C_CellC]			
95		+ltree_chconf_C(conf)			Config LT
		ltree_chconf_A(conf_a:B_3)			
96		[conf_a = C_cch_1nonComb]			
97		+NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1 non combined CCCH, cell A
98		(TCV_Upd := C_NotCombined)			
99		[conf_a = C_cch_1Comb]			
100		[sysinfonb <> C_Test_cbms]			
101		+CombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1 combined CCCH, cell A
102		(TCV_Upd := C_Combined)			
103		[sysinfonb = C_Test_cbms]			
104		+CombinedBCCH_A_CBMS(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1 combined CCCH + CBMS, cell A
105		(TCV_Upd := C_Combined)			
106		[conf_a = C_cch_2nonComb]			2 CCCHs
107		+NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1st non combined CCCH, cell A
108		+NonCombinedBCCH_A_2(acttype, slot2, tsc, ta, babr, conf_a, bpm)			2nd non combined CCCH, cell A
109		(TCV_Upd := C_NotCombined)			
110		[conf_a = C_cch_3nonComb]			
111		+NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta,			1st non combined CCCH, cell A

		babr, conf_a, bpm)			
112		+NonCombinedBCCH_A_2(acttype, slot2, tsc, ta, babr, conf_a, bpm)			2nd non combined CCCH, cell A
113		+NonCombinedBCCH_A_3(acttype, slot3, tsc, ta, babr, conf_a, bpm)			3rd non combined CCCH, cell A
114		(TCV_Upd := C_NotCombined)			
115		[conf_a = C_cch_4nonComb]			
116		+NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1st non combined CCCH, cell A
117		+NonCombinedBCCH_A_2(acttype,slot2, tsc, ta, babr, conf_a, bpm)			2nd non combined CCCH, cell A
118		+NonCombinedBCCH_A_3(acttype, slot3, tsc, ta, babr, conf_a, bpm)			3rd non combined CCCH, cell A
119		+NonCombinedBCCH_A_4(acttype, slot4, tsc, ta, babr, conf_a, bpm)			4th non combined CCCH, cell A
120		(TCV_Upd := C_NotCombined)			
121	E_0099	[(conf_a <> C_cch_1nonComb) AND(conf_a <> C_cch_2nonComb) AND(conf_a <> C_cch_3nonComb) AND(conf_a <> C_cch_4nonComb) AND(conf_a <> C_cch_1Comb)]		I	A wrong CCCH-CONF value, cell A
		ltree_chconf_B(conf_b:B_3)			
122		[conf_b = C_cch_1nonComb]			
123		+NonCombinedBCCH_B(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype,slot,tsc, ta, babr, conf_b, bpm)			1 non combined CCCH, cell B
124		(TCV_Upd := C_NotCombined)			
125		[conf_b = C_cch_1Comb]			
126		+CombinedBCCH_B(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_b, bpm)			1 combined CCCH, cell B
127		(TCV_Upd := C_Combined)			
128	E_0100	[(conf_b <> C_cch_1nonComb) AND(conf_b <> C_cch_1Comb)]		I	A wrong CCCH-CONF value,cell B
		ltree_chconf_C(conf_c:B_3)			
129		[conf_c = C_cch_1Comb]			
130		+CombinedBCCH_C(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_c, bpm)			1 combined CCCH, cell C
131		(TCV_Upd := C_Combined)			
132	E_0101	[conf_c <> C_cch_1Comb]		I	
		ltree_sysinfo_other(attach:INTEGER)			
133		[(cch_con = C_cch_1nonComb) OR (cch_con = C_cch_1Comb)]			
134		[cch_con = C_cch_2nonComb]			
135		+SysInfoSending_fh(sch, C_BCCH_A_2, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 2nd CCCH
136		[cch_con = C_cch_3nonComb]			
137		+SysInfoSending_fh(sch, C_BCCH_A_2, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 2nd CCCH
138		+SysInfoSending_fh(sch, C_BCCH_A_3, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 3rd CCCH
139		[cch_con = C_cch_4nonComb]			
140		+SysInfoSending_fh(sch, C_BCCH_A_2, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 2nd CCCH

141		+SysInfoSending_fh(sch, C_BCCH_A_3, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 3rd CCCH
142		+SysInfoSending_fh(sch, C_BCCH_A_4, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 4th CCCH
		ltree_rmvPLMNfbdnlist			
143		(TCV_Null := OO_PLMNselModeMan())			
144		(TCV_Null := OO_SelPLMN(mnc))			
145		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_02)		Any channel request PDU
146		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
147		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, subch, slot, tsc, TCV_chdescr_arfcn, ta))		
148		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(C_norm_period_attach))		Any location update type
149		ACTIVATE(OtherEvents)			Restore Normal default
150		[StartType = C_Start_PLMNnot]			The PLMN is already in forbidden list on short SIM
151		L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(C_rc_plmn_not))		Delete LAI, tmsi, cksn, PLMN still in list
152		+PostMainLinkRel(TCV_ch)			Release Ch
153		(TCV_Null := OO_PLMNselModeAuto())			
154		[StartType <> C_Start_PLMNnot]			
155		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiTmsi_r02iei, mcc, mnc, TCV_lac))		Location Accept with no TMSI
156		+PostMainLinkRel(TCV_ch)			Release Ch
157	E_0102	(TCV_Null := OO_PLMNselModeAuto())		I	No ChReq received: Inconclusive

Detailed Comments: Formal Parameter Type Comments & Explanations Possible values

par_bspwr INTEGER Base station power level C_E_default, C_E_neighbourdefault

activ_cell CellID Activated cell identity C_CellA, C_CellB, C_CellC

sch LOGICCH Sync. channel identity
If formal para activ_cell = C_CellA, C_SCH_A
If formal para activ_cell = C_CellB, C_SCH_B
If formal para activ_cell = C_CellC, C_SCH_C

bcch LOGICCH 1st broadcast channel id
If formal para activ_cell = C_CellA, C_BCCH_A_1
If formal para activ_cell = C_CellB, C_BCCH_B_1
If formal para activ_cell = C_CellC, C_BCCH_C_1

acttype BITSTRING Type for activation of a channel C_Immass

slot SN Slot number for the 1st slot C_S0

slot2 SN Slot number for the 2nd slot
In case of 2 or 3 or 4 non combined CCCH C_S2

Otherwise (1 combined or 1 non combined) C_SlotNU

slot3 SN Slot number for the 3rd slot
In case of 3 or 4 non combined CCCH C_S4
Otherwise (1 combined or 1, 2 non combined) C_SlotNU

slot4 SN Slot number for the 4th slot
In case of 4 non combined CCCH channels C_S6
Otherwise (1 combined,1, 2, 3 non combined) C_SlotNU

tsc TSC Training Sequence code C_BCC

t INTEGER Rach Control Parameter field, Tx-Integer
for SysInfo1, SysInfo2, SysInfo3, SysInfo4 C_TxInt_5, C_TxInt_7, C_TxInt_10

retr INTEGER Rach Control Parameter field, Max-Retrans
for SysInfo1, SysInfo2, SysInfo3, SysInfo4 C_Max_1, C_Max_2, C_Max_7

neci INTEGER Requested value f. NECI bit in test case C_NECI_0, C_NECI_1

att INTEGER Requested value f. ATTACH bit in test case C_ATT_0, C_ATT_1

ta TA Requested Timing Advance value

babr B_3 Number of blocks reserved for acces grant
If arbitrary TSPX_AGBLKSx
If fixed C_BABR_0, C_BABR_1, C_BABR_2

cch_con B_3 Type of CCCH channel.
If arbitrary TSPX_CcchConfx
1 conbined CCCH channel C_cch_1Comb
1 non conbined CCCH channel C_cch_1nonComb
2 non conbined CCCH channels C_cch_2nonComb
3 non conbined CCCH channels C_cch_3nonComb
4 non conbined CCCH channels C_cch_4nonComb

bpm B_3 Number of multiframes for transmission
of PAGING REQUEST messages to the same
paging subgroup
If arbitrary TSPX_PAMFRMSx
If fixed (2, 4, 5, 9 multiframes) C_BPM_0, C_BPM_2, C_BPM_3, C_BPM_7

t3212 OCTETSTR. Requested value for T3212 in test case (Periodic Location Update Value)
infinite C_T3212_0
6 mn / 1 deci hours C_T3212_1
12 mn / 2 deci hours C_T3212_2
30 mn / 5 deci hours C_T3212_5

ci CI Cell Identity
If formal para activ_cell = C_CellA C_ci_CellA
If formal para activ_cell = C_CellB C_ci_CellB
If formal para activ_cell = C_CellC C_ci_CellC

mcc OCTETSTR. Mobile Contry Code for the LAI C_MCC, C_MCC_x

mnc OCTETSTR. Mobile Network Code for the LAI C_PLMN_Home, C_PLMN_x

lac OCTETSTR. Local Area Code for the LAI C_LAC_x

co CO Cell Options CellOpt_xx

cchdgsm CCHD Cell Channels Description (GSM)
If formal para sysinfonb=C_Test_nfh CellChDes_omit
If formal para sysinfonb=C_Test_cbms CellChDes_omit
If formal para sysinfonb=C_Test_fh GSM CellChDes_xx

cchddcs CCHD Cell Channels Description (DCS)
If formal para sysinfonb=C_Test_nfh CellChDes_omit
If formal para sysinfonb=C_Test_cbms CellChDes_omit
If formal para sysinfonb=C_Test_fh DCS CellChDes_xx

crh INTEGER Cell Reselection Hysteresis C_CellReselectHys12, C_CellReselectHys4

mtmcgsm INTEGER Requested maximum power level for GSM MS C_MaxPwrLvlG

mtmcdcs INTEGER Requested maximum power level for DCS MS C_MaxPwrLvlD

bcchflgsm NCD Neighbour Cells Description for GSM BcchFreqLst_xx

bcflaltgsm NCD Alternative Neighbour Cells Description for GSM.
If sysinfonb = C_Test_nfh Alternat.GSM BcchFreqLst_xx
Otherwise: BcchFreqLst_Omit

bcchfldcs NCD Neighbour Cells Description for DCS BcchFreqLst_xx

bcflaltdcs NCD Alternative Neighbour Cells Description for DCS
If sysinfonb = C_Test_nfh Alternat.DCS BcchFreqLst_xx
Otherwise: BcchFreqLst_Omit

Re B_1 Rach Control Parameter field,
Re-establishment bit for SysInfo1,
SysInfo2, SysInfo3, SysInfo4 C_Restablishment, C_noRestablishmen
bcc BCC Base station Color Code C_BCC

ncc NCC Network Color Code C_NCC

StartType INTEGER Initial conditions for MS
Starting without TMSI C_Start_Imsi
Starting with Tmsi_01 C_Start_Tmsi
Starting with Tmsi_01, MS deactivated C_Start_TmsiOff
Starting with LAI deleted, MS deactivated C_Start_Rej
Starting with PLMN not allowed C_Start_PLMNnot

subch BITSTRING Selected sub-channel
If Formal para cch_con = C_cch_1Comb, TSPX_SDCCH4SubDef TSPX_SDCCH4SubA, SubB, SubC
cksn BITSTRING Ciphering Key Sequence Number used for authentication TSPX_CKSNDef, TSPX_CKSNA, B, C

rand BITSTRING Random Number for authentication TSPX_RANDDef, TSPX_RANDA, B , C

arfcngsm INTEGER BCCH channel ARFCN (GSM)
If sysinfonb = C_Test_fh C_Noarfcn
Otherwise: If activ_cell = C_CellA C_arfcnA_x If activ_cell = C_CellB C_arfcnB_x If activ_cell = C_CellC C_arfcnC_x

arfcndcs INTEGER BCCH channel ARFCN (DCS)
If sysinfonb = C_Test_fh C_Noarfcn

	Otherwise:	If activ_cell = C_CellA
C_arfcnAd_x	If activ_cell = C_CellB	C_arfcnBd_x If activ_cell = C_CellC C_arfcnCd_x
imsi	HEXSTRING	The expected IMSI In TC_26_7_3_2 TSPX_IMSI C_shortIMSI
sysinfonb	INTEGER	Type of sysinfo sending
		Frequency Hopping C_Test_fh
		Non Frequency hopping C_Test_nfh
		EGSM sequence C_Test_egsm
		CBMS sequence C_Test_cbms
nccp	NCCP	Network Country Code Permitted in SysInfo2 C_NCCP_2, C_NCCP_5

Test Step Dynamic Behaviour

Test Step Name: PreEstRRConn(slot:SN; tsc:TSC; ta:TA)
Group: GSM_MS_EFR_4220/Preambles/
Objective: To establish a RR connection on SDCCH subchannel defined by TSPX_SDCCH4SubDef in cell A
Default: OtherEvents
Comments: The calling tree shall prepare variable for the step: TCV_ch for the SDCCH4 subchannel TSPX_SDCCH4SubDef. This test step is used for non RR testing.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
2		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
3		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
4		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, slot, tsc, TCV_chdescr_arfcn, ta))		
5		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
6		ACTIVATE(OtherEvents)			Restore Normal default

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: StartCellA_ho(par_bspwr, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING; cchdgsm, cchddcs:CCHD; bcchflgsm, bcchfldcs:NCD; Re:BITSTRING; bcc:BCC; ncc:NCC; td, fn:INTEGER)

Group: GSM_MS_EFR_4220/Preambles/

Objective: To setup a physical channel as combined BCCH, CCCH and SDCCH4 then broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with default parameters of cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
2		+ltree_chconf			
3		+SysInfoSending_fh(C_SCH_A, C_BCCH_A_1, t, retr, neci, att, babr, cch_con, bpm, t3212, C_ci_cellA, mcc, mnc, lac, CellOpt_01, cchdgsm, cchddcs, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchflgsm, bcchfldcs, Re, bcc, ncc, C_NCCP_2)			
4		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
5		[cch_con = C_cch_1Comb]			
6		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			
7		[cch_con = C_cch_1nonComb]			
8	E_0103	[(cch_con <> C_cch_1nonComb) AND(cch_con <> C_cch_1Comb)] **ltree_chconf**		I	
9		[cch_con = C_cch_1Comb]			
10		+CombinedBCCH_A_ho(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, arfcn1, arfcn2, acttype, slot, tsc, ta, babr, cch_con, bpm, td, fn)			
11		[cch_con = C_cch_1nonComb]			
12		+NonCombinedBCCH_A_ho(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, arfcn1, arfcn2, acttype, slot, tsc, ta, babr, cch_con, bpm, td, fn)			
13	E_0104	[(cch_con <> C_cch_1nonComb) AND(cch_con <> C_cch_1Comb)]		I	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: StartCellB_ho(par_bspwr, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING; cchdgsm, cchddcs:CCHD; bcchflgsm, bcchfldcs:NCD; Re:BITSTRING; bcc:BCC; ncc:NCC; td, fn:INTEGER)

Group: GSM_MS_EFR_4220/Preambles/

Objective: To setup a physical channel as combined BCCH, CCCH and SDCCH4 then broadcast SYSTEM INFORMATION messages 2, 3, 4, 5 and 6 with default parameters of cell B.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_sacch_B := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
2		+ltree_chconf			
3		+SysInfoSending_fh(C_SCH_B, C_BCCH_B_1, t, retr, neci, att, babr, cch_con, bpm, t3212, C_ci_cellB, mcc, mnc, lac, CellOpt_01, cchdgsm, cchddcs, C_CellReselectHys12, C_MaxPwrLvlG, C_MaxPwrLvlD, bcchflgsm, bcchfldcs, Re, bcc, ncc, C_NCCP_2)			
4		(TCV_Ccd0B := TCV_Ccd0H, TCV_sysinfo5_B := TCV_sysinf5, TCV_sysinfo6_B := TCV_sysinf6)			
5		[cch_con = C_cch_1Comb]			
6		+SysInfo_SacchSending(TCV_sacch_B, TCV_sysinfo5_B, TCV_sysinfo6_B)			
7		[cch_con = C_cch_1nonComb]			
8	E_0105	[(cch_con <> C_cch_1nonComb) AND(cch_con <> C_cch_1Comb)]		I	
		ltree_chconf			
9		[cch_con = C_cch_1Comb]			
10		+CombinedBCCH_B_ho(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, arfcn1, arfcn2, acttype, slot, tsc, ta, babr, cch_con, bpm, td, fn)			
11		[cch_con = C_cch_1nonComb]			
12		+NonCombinedBCCH_B_ho(par_bspwr, C_MaxPwrLvlG, C_MaxPwrLvlD, arfcn1, arfcn2, acttype, slot, tsc, ta, babr, cch_con, bpm, td, fn)			
13	E_0106	[(cch_con <> C_cch_1nonComb) AND(cch_con <> C_cch_1Comb)]		I	

Detailed Comments:

Defaults Library

Default Dynamic Behaviour

Default Name: OtherEvents
Group: GSM_MS_EFR_4220/
Objective: To match unexpected events and sign final verdict for preambles.
Comments: used in preambles.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_02		4.
2		RETURN			
3		?TIMEOUT T_guard			1.
4		+ltree_SDCCH1_release			
5		+ltree_SDCCH2_release			
6		+ltree_TCH1_release			
7		+ltree_TCH2_release			
8		(TCV_Null := OO_TguardTimeOut())			
9	E_0107	CANCEL		I	3.
10		L?OTHERWISE			2.
11		+ltree_SDCCH1_release			
12		+ltree_SDCCH2_release			
13		+ltree_TCH1_release			
14		+ltree_TCH2_release			
15		L?DL_RelIn	DLRelInd_01		
16	E_0108	CANCEL		I	3.
17		?TIMEOUT T_guard			1.
18	E_0109	CANCEL		I	3.
		ltree_SDCCH1_release			
19		[TCV_ch <> "dummy"]			
20		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
21		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
22		[TCV_ch1 <> "dummy"]			
23		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
24		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
25		[TCV_chTch <> "dummy"]			
26		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
27		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
28		[TCV_chTch1 <> "dummy"]			
29		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
30		[TCV_chTch1 = "dummy"]			

Detailed Comments:
1. The guard timer times out, inconclusive.
2. Unexpected events, inconclusive.
3. Cancel of all running timers.
4. Local end termination request allowed.

Default Dynamic Behaviour

Default Name: OtherEventsFail
Group: GSM_MS_EFR_4220/
Objective: To match unexpected events and fail the test case.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_02		
2		RETURN			
3		?TIMEOUT T_guard			1.
4		+ltree_SDCCH1_release			
5		+ltree_SDCCH2_release			
6		+ltree_TCH1_release			
7		+ltree_TCH2_release			
8		(TCV_Null := OO_TguardTimeOut())			
9	E_01 10	CANCEL		I	3.
10		L?OTHERWISE			2.
11		+ltree_SDCCH1_release			
12		+ltree_SDCCH2_release			
13		+ltree_TCH1_release			
14		+ltree_TCH2_release			
15		L?DL_RelIn	DLRelInd_01		
16	E_01 11	CANCEL		F	3.
17		?TIMEOUT T_guard			
18	E_01 12	CANCEL		F	3.
		ltree_SDCCH1_release			
19		[TCV_ch <> "dummy"]			
20		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
21		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
22		[TCV_ch1 <> "dummy"]			
23		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
24		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
25		[TCV_chTch <> "dummy"]			
26		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
27		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
28		[TCV_chTch1 <> "dummy"]			
29		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
30		[TCV_chTch1 = "dummy"]			

Detailed Comments:
1. The guard timer times out, inconclusive.
2. Unexpected events, fail.
3. Cancel of all running timers.

Default Dynamic Behaviour

Default Name: OtherEventsFail_01
Group: GSM_MS_EFR_4220/
Objective: To match irrelevant CHANNEL REQUEST msg and MEASUREMENT REPORT msg and return or match other unexpected events and fail the test case.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_01		6.
2		RETURN			
3		?TIMEOUT T_guard			1.
4		+ltree_SDCCH1_release			
5		+ltree_SDCCH2_release			
6		+ltree_TCH1_release			
7		+ltree_TCH2_release			
8		(TCV_Null := OO_TguardTimeOut())			
9	E_01 13	CANCEL		I	5.
10		L?DL_RacInChRq	ChReq(ChRequest_02)		2.
11		RETURN			
12		L?DL_UdatInMsrRpt	MsrRept(MsrReport_02)		3.
13		RETURN			
14		L?OTHERWISE			4.
15		+ltree_SDCCH1_release			
16		+ltree_SDCCH2_release			
17		+ltree_TCH1_release			
18		+ltree_TCH2_release			
19		L?DL_RelIn	DLRelInd_01		
20	E_01 14	CANCEL		F	5.
21		?TIMEOUT T_guard			
22	E_01 15	CANCEL		F	5.
		ltree_SDCCH1_release			
23		[TCV_ch <> "dummy"]			
24		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
25		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
26		[TCV_ch1 <> "dummy"]			
27		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
28		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
29		[TCV_chTch <> "dummy"]			
30		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
31		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
32		[TCV_chTch1 <> "dummy"]			
33		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
34		[TCV_chTch1 = "dummy"]			

Detailed Comments:
1. The guard timer times out, inconclusive.
2. To throw away any irrelevant channel request.
3. To throw away any measurement report.
4. Other unexpected events, fail.
5. Cancel of all running timers.
6. Cover also DLRelInd_02 for local end termination.

Default Dynamic Behaviour

Default Name: OtherEventsFail_02

Group: GSM_MS_EFR_4220/

Objective: To match unexpected events and fail the test case but ignore channel request messages that are sent before the lower tester has sent (and the MS received) the Immediate Assigment message.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_02		
2		RETURN			
3		L?DL_RacInChRq	ChReq(ChRequest_02)		1.
4		RETURN			
5		?TIMEOUT T_guard			2.
6		+ltree_SDCCH1_release			
7		+ltree_SDCCH2_release			
8		+ltree_TCH1_release			
9		+ltree_TCH2_release			
10		(TCV_Null := OO_TguardTimeOut())			
11	E_0116	CANCEL		I	4.
12		L?OTHERWISE			3.
13		+ltree_SDCCH1_release			
14		+ltree_SDCCH2_release			
15		+ltree_TCH1_release			
16		+ltree_TCH2_release			
17		L?DL_RelIn	DLRelInd_01		
18	E_0117	CANCEL		F	4.
19		?TIMEOUT T_guard			
20	E_0118	CANCEL		F	4.
		ltree_SDCCH1_release			
21		[TCV_ch <> "dummy"]			
22		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
23		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
24		[TCV_ch1 <> "dummy"]			
25		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
26		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
27		[TCV_chTch <> "dummy"]			
28		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
29		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
30		[TCV_chTch1 <> "dummy"]			
31		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
32		[TCV_chTch1 = "dummy"]			

Detailed Comments:

1. To throw away any irrelevant channel request.
2. The guard timer times out, inconclusive.
3. Unexpected events, fail.
4. Cancel of all running timers.

Default Dynamic Behaviour

Default Name: OtherEvents_01
Group: GSM_MS_EFR_4220/
Objective: To match irrelevant messages and return

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?OTHERWISE			1.
2		RETURN			

Detailed Comments: 1. Cover also DLRelInd_02 for local end termination.

Default Dynamic Behaviour

Default Name: OtherEvents_02
Group: GSM_MS_EFR_4220/
Objective: To match unexpected events and sign final verdict for preambles but ignore channel request messages that are sent before the lower tester has sent (and the MS received) the Immediate Assigment message.
Comments: used in preambles.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_02		
2		RETURN			
3		L?DL_RacInChRq	ChReq(ChRequest_02)		1.
4		RETURN			
5		?TIMEOUT T_guard			2.
6		+ltree_SDCCH1_release			
7		+ltree_SDCCH2_release			
8		+ltree_TCH1_release			
9		+ltree_TCH2_release			
10		(TCV_Null := OO_TguardTimeOut())			
11	E_0119	CANCEL		I	4.
12		L?OTHERWISE			3.
13		+ltree_SDCCH1_release			
14		+ltree_SDCCH2_release			
15		+ltree_TCH1_release			
16		+ltree_TCH2_release			
17		L?DL_RelIn	DLRelInd_01		
18	E_0120	CANCEL		I	4.
19		?TIMEOUT T_guard			2.
20	E_0121	CANCEL		I	4.
		ltree_SDCCH1_release			
21		[TCV_ch <> "dummy"]			
22		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
23		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
24		[TCV_ch1 <> "dummy"]			
25		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
26		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
27		[TCV_chTch <> "dummy"]			
28		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
29		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
30		[TCV_chTch1 <> "dummy"]			
31		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
32		[TCV_chTch1 = "dummy"]			

Detailed Comments:
1. To throw away any irrelevant channel request.
2. The guard timer times out, inconclusive.
3. Unexpected events, inconclusive.
4. Cancel of all running timers.

Default Dynamic Behaviour

Default Name: RcvHdOvAcc
Group: GSM_MS_EFR_4220/
Objective: To match any HANDOVER ACCESS then return to calling tree.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RacInHoacc	HndOvAcc_01		
2		RETURN			

Detailed Comments:

Annex E (informative): Cell Selection Abstract Test Suite

The machine processable ATS

The electronic forms of the machine processable files (TTCN MP format) corresponding to the ATS for Layer 3 are contained in the file cs-4m1.mp[3)] (Cell Selection) on the diskette included as a part of this ETS (Part 3).

The graphical form ATS

The TTCN.GR representation of this ATS is contained in the remainder of this annex.

Suite overview

Suite structure

Test Suite Structure	
Suite Name:	GSM_MS_CS_4220
Standards Ref:	ETS 300 557
PICS Ref:	
PIXIT Ref:	
Test Method(s):	Distributed Single Layer Test Method
Comments:	v4.22.0

Structure and Objectives			
Test Group Reference	**Selection Ref**	**Test Group Objective**	**Page Nr**
GSM_MS_CS_4220/CellSelection/	SelExpr_0000	To verify cell selection and reselection	2348
Detailed Comments:			

3) This file is located in an archive file named 6073_E8.ZIP. Other file formats are available on request.

Test case index

Test Case Index				
Test Group Reference	**Test Case Id**	**Selection ref**	**Description**	**Page Nr**
GSM_MS_CS_4220/ CellSelection/	TC_20_1	SelExpr_0000	Cell selection	2348
GSM_MS_CS_4220/ CellSelection/	TC_20_2	SelExpr_0000	Cell selection with varying signal strength values	2352
GSM_MS_CS_4220/ CellSelection/	TC_20_4	SelExpr_0000	Cell reselection using TEMPORARY_OFFSET, CELL_RESELECT_OFFSET, POWER_OFFSET and PENALTY_TIME parameters	2354
GSM_MS_CS_4220/ CellSelection/	TC_20_5	SelExpr_0000	Cell reselection using parameters transmitted in the System Information type 2bis, type 2ter, type 7 and type 8 messages	2357
GSM_MS_CS_4220/ CellSelection/	TC_20_6	SelExpr_0000	Cell reselection timings	2359
GSM_MS_CS_4220/ CellSelection/	TC_20_7	SelExpr_0000	Priority of cells	2363
GSM_MS_CS_4220/ CellSelection/	TC_20_8	SelExpr_0000	Cell reselection when C1 (serving cell) < 0 for 5 seconds	2367
GSM_MS_CS_4220/ CellSelection/	TC_20_9	SelExpr_0000	Running average of the surrounding cell BCCH carrier signal levels	2369
GSM_MS_CS_4220/ CellSelection/	TC_20_10	SelExpr_0000	Running average of the serving cell BCCH carrier signal level	2372
GSM_MS_CS_4220/ CellSelection/	TC_20_12	SelExpr_0000	Decoding the BCCH information of the neighbour carriers on the list of six strongest neighbour carriers	2374
GSM_MS_CS_4220/ CellSelection/	TC_20_13	SelExpr_0000	Decoding the BSIC of the neighbour carriers on the list of six strongest neighbour carriers	2376
GSM_MS_CS_4220/ CellSelection/	TC_20_15	SelExpr_0000	Cell reselection due to MS rejection "LA not allowed"	2378
GSM_MS_CS_4220/ CellSelection/	TC_20_16	SelExpr_0000	Downlink signalling failure	2382
GSM_MS_CS_4220/ CellSelection/	TC_20_17	SelExpr_0000	Cell selection if no suitable cell found in 10 s	2385
Detailed Comments:				

Test step index

Test Step Index			
Test Step Group Reference	**Test Step Id**	**Description**	**Page Nr**
GSM_MS_CS_4220/management/	StopAllBCCH	To stop the RF transmission of all BCCH channels in all active cells.	2386
GSM_MS_CS_4220/management/ChConfig/	Config_CBMS_A_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH, SDCCH4 and CBCH onto the physical channel which represents cell A.	2387
GSM_MS_CS_4220/management/ChConfig/	Config_BCCH_CCCH_A_1	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell A.	2387
GSM_MS_CS_4220/management/ChConfig/	Config_BCCH_CCCH_A_2	To set a physical channel and map the second BCCH, CCCH onto the physical channel which represents cell A.	2387
GSM_MS_CS_4220/management/ChConfig/	Config_BCCH_CCCH_A_3	To set a physical channel and map the third BCCH, CCCH onto the physical channel which represents cell A.	2388
GSM_MS_CS_4220/management/ChConfig/	Config_BCCH_CCCH_A_4	To set a physical channel and map the fourth BCCH, CCCH onto the physical channel which represents cell A.	2388
GSM_MS_CS_4220/management/ChConfig/	Config_BCCH_CCCH_B_1	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell B.	2388
GSM_MS_CS_4220/management/ChConfig/	Config_SDCCH4_A_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.	2389
GSM_MS_CS_4220/management/ChConfig/	Config_SDCCH4_A_1_fn		2389
GSM_MS_CS_4220/management/ChConfig/	Config_SDCCH4_B_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.	2390
GSM_MS_CS_4220/management/ChConfig/	Config_SDCCH4_B_1_fn		2390
GSM_MS_CS_4220/management/ChConfig/	Config_SDCCH4_C_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell C	2391
GSM_MS_CS_4220/management/ChConfig/	Config_SDCCH4_D_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell D	2391
GSM_MS_CS_4220/management/ChConfig/	Config_SDCCH4_E_1	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell E	2392
GSM_MS_CS_4220/management/ChConfig/	Config_SDCCH8_A_1	To set one physical channel used as SDCCH8 channel for instance 1 of cell A.	2392
GSM_MS_CS_4220/management/ChConfig/	Config_SDCCH8_A_2	To set one physical channel used as SDCCH8 channel for instance 2 of cell A.	2393
GSM_MS_CS_4220/management/ChConfig/	Config_SDCCH8_B_1	To set one physical channel used as SDCCH8 channel for instance 1 of cell B.	2393
GSM_MS_CS_4220/management/ChConfig/	CombinedBCCH_A_CBMS	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.	2394
GSM_MS_CS_4220/management/ChConfig/	CombinedBCCH_A	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.	2394
GSM_MS_CS_4220/management/ChConfig/	CombinedBCCH_A_fn		2394
GSM_MS_CS_4220/management/ChConfig/	CombinedBCCH_B	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.	2395
GSM_MS_CS_4220/management/ChConfig/	CombinedBCCH_B_fn		2395
GSM_MS_CS_4220/management/ChConfig/	CombinedBCCH_C	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell C.	2395
GSM_MS_CS_4220/management/ChConfig/	CombinedBCCH_D	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell D.	2396
GSM_MS_CS_4220/management/ChConfig/	CombinedBCCH_E	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell E.	2396

GSM_MS_CS_4220/management/ChConfig/	NonCombinedBCCH_A	To set one physical channel used as FCHH_SCH_BCCH_CCCH for cell A.	2396
GSM_MS_CS_4220/management/ChConfig/	NonCombinedBCCH_A_2	To set a physical channel and map the second BCCH, CCCH onto the physical channel which represents cell A.	2397
GSM_MS_CS_4220/management/ChConfig/	NonCombinedBCCH_A_3	To set a physical channel and map the third BCCH, CCCH onto the physical channel which represents cell A.	2397
GSM_MS_CS_4220/management/ChConfig/	NonCombinedBCCH_A_4	To set a physical channel and map the fourth BCCH, CCCH onto the physical channel which represents cell A.	2397
GSM_MS_CS_4220/management/ChConfig/	NonCombinedBCCH_B	To set a physical channel and map FCCH, SCH, BCCH, CCCH onto the physical channel which represents cell B for RR testing.	2398
GSM_MS_CS_4220/management/ChConfig/	SDCCH8_A_1_nocip h	To set one physical channel used as SDCCH8 channel for instance 1 of cell A.	2398
GSM_MS_CS_4220/management/ChConfig/	SDCCH8_A_2_nocip h	To set one physical channel used as hopping SDCCH8 channel for instance 2 of cell A, for TC_26_6_13_1.	2399
GSM_MS_CS_4220/management/ChConfig/	SDCCH8_B_1_nocip h	To set one physical channel used as hopping SDCCH8 channel for instance 1 of cell B For TC_26_6_13_5.	2399
GSM_MS_CS_4220/Miscellaneous/	Authentication	To execute the authentication procedure.	2400
GSM_MS_CS_4220/Miscellaneous/	CCCH_group_Paging _group		2401
GSM_MS_CS_4220/Miscellaneous/	ImsiAttach		2402
GSM_MS_CS_4220/Miscellaneous/	ImsiAttachIni		2403
GSM_MS_CS_4220/Miscellaneous/	ImsiAttachNoReactio n		2404
GSM_MS_CS_4220/Miscellaneous/	ImsiDetach		2405
GSM_MS_CS_4220/Miscellaneous/	ImsiDetachNoReactio n		2406
GSM_MS_CS_4220/Miscellaneous/	MM_LupInit2		2406
GSM_MS_CS_4220/Miscellaneous/	MM_LupRej		2406
GSM_MS_CS_4220/Miscellaneous/	MM_PwrOrSimOff		2407
GSM_MS_CS_4220/Miscellaneous/	MM_PwrOrSimOn		2407
GSM_MS_CS_4220/Miscellaneous/	NoReaction	To check whether MS doesn't initiate any RR connections.	2407
GSM_MS_CS_4220/Miscellaneous/	PageOnCarriers		2408
GSM_MS_CS_4220/Miscellaneous/	Pause		2409
GSM_MS_CS_4220/Miscellaneous/	SelectPagingCh		2409
GSM_MS_CS_4220/Miscellaneous/	Varinit_fix		2410
GSM_MS_CS_4220/Miscellaneous/	Wait		2410
GSM_MS_CS_4220/Miscellaneous/	WaitForInService	To wait until the MS enters the Idle and updated state.	2410
GSM_MS_CS_4220/Miscellaneous/SysInfo/	SysInfoSending_cbm s	To send system information messages for the CBMS tests.	2411
GSM_MS_CS_4220/Miscellaneous/SysInfo/	SysInfoSending_fh	To send system information messages for the L3 tests. The following parameters specified by input parameters.	2413
GSM_MS_CS_4220/Miscellaneous/SysInfo/	SysInfoSending_fh_bisA		2415
GSM_MS_CS_4220/Miscellaneous/SysInfo/	SysInfoSending_fh_bisB		2417
GSM_MS_CS_4220/Miscellaneous/SysInfo/	SysInfoSending_fh_rcc		2419
GSM_MS_CS_4220/Miscellaneous/SysInfo/	SysInfoSending_fh_rpct		2421
GSM_MS_CS_4220/Miscellaneous/SysInfo/	SysInfoSending_fh_rpctp		2423
GSM_MS_CS_4220/Miscellaneous/SysInfo/	SysInfoSending_nfh	To send system information messages with default parameters defined for L3 tests for which no other special parameters indicated.	2425
GSM_MS_CS_4220/Miscellaneous/SysInfo/	SysInfoSending_nfhCb	To send system information messages with default parameters defined for L3 tests except that the cell is barred.	2427
GSM_MS_CS_4220/Miscellaneous/SysInfo/	SysInfoSending_e	Sending of Systeminformation for cell A in EGSM test case TC_26_10_2_2.	2428
GSM_MS_CS_4220/Miscellaneous/SysInfo/	SysInfo_SacchSendin g	To send SYSTEM INFORMATION 5 and 6 messages defined by parameters 'sysinfo5_pdu' and 'sysinfo6_pdu' in the parametrised 'ch' channel.	2429

GSM_MS_CS_4220/Miscellaneous/SysInfo/	SysInfo_5bisSending	To send SYSTEM INFORMATION 5bis message defined by parameters 'sysinfo5bis_pdu' in the parametrised 'ch' channel.	2429
GSM_MS_CS_4220/Miscellaneous/SysInfo/	SetNECI	To set the NECI =1.	2430
GSM_MS_CS_4220/Miscellaneous/SysInfo/	SetATT	To set the ATT flag to "MS's in the cell should apply IMSI attach and detach procedure"	2431
GSM_MS_CS_4220/OperatorOP/	InitCall		2432
GSM_MS_CS_4220/OperatorOP/	PLMNsCHK	To check whether the MS presents a list of available PLMNs.	2432
GSM_MS_CS_4220/Postambles/	PostLinkRelEnd	To release the RR connection and bring the MS back to Idle state. This teststep decides the final verdict and therefor it shall be used only at the end of testcases.	2432
GSM_MS_CS_4220/Postambles/	PostMainLinkRel	To release the main signalling link `ch`, and bring the MS back to Idle state.	2432
GSM_MS_CS_4220/Postambles/	ChanRel	To release the RR connection on the channel TCV_ch and bring the MS back to Idle state.	2433
GSM_MS_CS_4220/Postambles/	ChanRel_P	To release the RR connection on the channel TCV_chmaindcch and bring the MS back to Idle state. This teststep decides the verdict (P).	2433
GSM_MS_CS_4220/Postambles/	ChanRel_end	To release the RR connection and bring the MS back to Idle state. This teststep decides the final verdict and therefor it shall be used only at the end of testcases.	2433
GSM_MS_CS_4220/Preambles/	BasicServiceMO	1. This test step generates an appropriate MO SETUP message according to the IXIT parameters of Bearer services/Teleservices for an MO call. 2. The detailed algorithms for Bcap derivation for each Bearer service/Teleservice are described in test steps BS2xMO, BS3xMO, BS4xMO, BS5xMO, BS61or81MO, TS61MO and TS62MO.	2434
GSM_MS_CS_4220/Preambles/	BS2xMO	MO SETUP message with right Bcap for BS2x service.	2437
GSM_MS_CS_4220/Preambles/	BS3xMO	MO SETUP message with right Bcap for BS3x service.	2440
GSM_MS_CS_4220/Preambles/	BS4xMO	MO SETUP message with right Bcap for BS4x service.	2443
GSM_MS_CS_4220/Preambles/	BS5xMO	MO SETUP message with right Bcap for BS5x service.	2445
GSM_MS_CS_4220/Preambles/	BS61or81MO	To get a MO SETUP message with right BC IE for BS61 or BS81 service.	2446
GSM_MS_CS_4220/Preambles/	Chmod	To assign correct values to the test case variables TCV_ChMod and TCV_ChModb.	2450
GSM_MS_CS_4220/Preambles/	IdleUpdated	This test step is called at the preamble of each test case.	2451
GSM_MS_CS_4220/Preambles/	PreEstRRConn	To establish a RR connection on TSPX_SDCCH4SubDef	2458
GSM_MS_CS_4220/Preambles/	PreEstRRC_MM	To establish a RR connection for MM testcases.	2459
GSM_MS_CS_4220/Preambles/	TS61MO	MO SETUP message with right Bcap for TS61 service.	2460
GSM_MS_CS_4220/Preambles/	TS62MO	MO SETUP message with right Bcap for TS62 service.	2462

Detailed Comments:

Default index

Default Index			
Default Group Reference	**Default Id**	**Description**	**Page Nr**
GSM_MS_CS_4220/	OtherEvents	To match unexpected events and sign final verdict for preambles.	2464
GSM_MS_CS_4220/	OtherEventsFail	To match unexpected events and fail the test case.	2465
GSM_MS_CS_4220/	OtherEventsFail_01	To match irrelevant CHANNEL REQUEST msg and return or match other unexpected events and fail the test case.	2466
GSM_MS_CS_4220/	OtherEventsFail_02	To match unexpected events and fail the test case but ignore channel request messages that are sent before the lower tester has sent (and the MS received) the Immediate Assigment message.	2467
GSM_MS_CS_4220/	OtherEvents_01	To match irrelevant messages and return	2468
GSM_MS_CS_4220/	OtherEvents_02	To match unexpected events and sign final verdict for preambles but ignore channel request messages that are sent before the lower tester has sent (and the MS received) the Immediate Assigment message.	2469
Detailed Comments:			

Declarations Part

Test suite type definitions

Simple type definitions

Simple Type Definitions		
Type Name	**Type Definition**	**Comments**
B_1	BITSTRING [1]	Generic type for 1 bit value
B_2	BITSTRING [2]	Generic type for 2 bits value
B_3	BITSTRING [3]	Generic type for 3 bits value
B_4	BITSTRING [4]	Generic type for 4 bits value
B_5	BITSTRING [5]	Generic type for 5 bits value
B_6	BITSTRING [6]	Generic type for 6 bits value
B_8	BITSTRING [8]	Generic type for 8 bits value
BCCHFRQ	BITSTRING [5]	position of a bcch carrier in the bcch channel list
BCC	BITSTRING [3]	BS colour code
BCDN	OCTETSTRING [1..10]	BCD numbers, GSM 04.08, 10.5.4.7, octet 4-13
BSIC	BITSTRING [6]	base station identity code
CCCH_CON	BITSTRING[3]	number of physical channels for ccch GSM 04.08, 10.5.2.11
CCSTATE	INTEGER(1,3,4,6,7,8,9,10,11,12,19,26,27)	
CellID	IA5String	Cell identifier
CHANNEL	BITSTRING [2]	needed channel type
CHMOD_VAL	BITSTRING[8]	value for channel mode GSM 04.08, 10.5.2.6, 10.5.2.7
CH_TDMA	BITSTRING[5]	channel type and TDMA offset GSM 04.08, 10.5.2.5
CI	OCTETSTRING [2]	cell identity, GSM 04.08, 10.5.1.1
CKSN	BITSTRING[3]	ciphering key sequence number(only key sequence) GSM 04.08, 10.5.1.2
CLRSUP	BITSTRING('10100001'B)	CLIR suppression GSM 04.08, 10.5.4.11a
CLRINV	BITSTRING('10100010'B)	CLIR invocation GSM 04.08, 10.5.4.11b
CMSVTYPE	BITSTRING [4]	CM service type
CP_CAU	OCTETSTRING [1]	cp cause GSM 04.11, 8.1.4.2
CPHALG	BITSTRING[3]	Ciphering algorithm identifier GSM 04.08, 10.5.2.9
CS	BITSTRING [2]	coding standard
EstMode	OCTETSTRING [1]	establish mode
EXTB	BITSTRING [1]	extension bit
FCS	HEXSTRING [2]	TP failure cause
FLWCNTL	INTEGER(0, 1, 2)	type of flow control. 0--outband flow control; 1--inband flow control; 2-- no flow control
HORF	BITSTRING [8]	handover reference
HSN	BITSTRING[6]	Hopping sequence number e.g.GSM 04.08 10.5.2.5
IDTYPE	B_4	identity type
IEI_4	BITSTRING [4]	information element identifier, type 1
IEI_8	BITSTRING [8]	information element identifier, type 2-4
L2FMTYPE	INTEGER(1..3)	L2 frame type
LENGTH	OCTETSTRING [1]	IE length or L2 pseudo length. The L2 pseudo length is composed of a L2 pseudo length value, which is 6 bits long, and of 2 additional bits. See GSM 04.08 section 10.5.2.19
LOGICCH	IA5String	

LOGCH	IA5String	
MAC	OCTETSTRING[3]	mac, e.g. GSM 04.08 10.5.2.21
MAIO	BITSTRING[6]	MAIO,GSM 04.08 10.5.2.5
MAXTXPOW	INTEGER(0..31)	Max Tx Power Level GSM 04.08, 10.5.2.4
MR	OCTETSTRING [1]	SMS RP or TP message reference
MT	BITSTRING[8]	message type, structure of it see MT0
MTI	BITSTRING [3]	SMS RP message type indicator
NCC	BITSTRING [3]	PLMN colour code
NCCP	OCTETSTRING [1]	ncc permitted
NTI	BITSTRING('10000000'B, '10000001'B, '10000010'B)	notification indicator GSM 04.08, 10.5.4.20
PD	B_4	Protocol discriminator
PGG	BITSTRING [8]	the paging group of an MS to be paged in binary presentation
RATE	IA5String("F", "H")	
RAND	BITSTRING [128]	random number
RCSD	BITSTRING [8]	reverse call setup direction GSM 04.08, 10.5.4.22a
RelMode	OCTETSTRING [1]	release mode
REJCAU	OCTETSTRING [1]	reject cause GSM 04.08, 10.5.3.6
RPI	BITSTRING [8]	repeat indicator
RRCAU	BITSTRING [8]	RR cause GSM 04.08, 10.5.2.31
RXLEV	BITSTRING [6]	received signal strength
SAPID	OCTETSTRING [1]	sap identifier
SENDINGMODE	INTEGER(1..4)	the mode of sending two consecutive messages containing paging mode IE
SERVICES	IA5String("C_Telephony", "C_EmgCall", "C_AltSpchFax", "C_AutoFax", "C_Async300", "C_Async1200", "C_Async120075", "C_Async2400", "C_Async4800", "C_Async9600", "C_Sync1200", "C_Sync2400", "C_Sync4800", "C_Sync9600", "C_PAD300", "C_PAD1200", "C_PAD120075", "C_PAD2400", "C_PAD4800", "C_PAD9600", "C_Packet2400", "C_Packet4800", "C_Packet9600", "C_AltSpchData", "C_SpchData")	
SHOCT	B_4	spare half octet for type 1 information element GSM 04.08, 10.5.1.8
SKI	B_4	Skip indicator
SN	BITSTRING [3]	all possible slot number, GSM 04.08 10.5.2.5
SPB	B_1	spare bit
SP2B	B_2	2 spare bits
SP3B	B_3	3 spare bits
SP5B	B_5	5 spare bits
SP6B	B_6	6 spare bits
SRES	OCTETSTRING [4]	authentication response signature
TPCD	OCTETSTRING [1..157]	TP command data
TPSCTS	HEXSTRING [14]	TP service centre time stamp
TPUD	OCTETSTRING [1..140]	TP user data
TSC	B_3	Training sequence code,GSM 04.08 10.5.2.5
T1	BITSTRING [11]	
T1_	B_5	T1'
T2	B_5	
T3	BITSTRING [6]	
T3_	B_3	T3'
TI_V	B_3	TI value
TMSI_V	OCTETSTRING[4]	TMSI value is octetstring of length 4
TZONES	INTEGER(-79..79)	time zones in 15 minute steps
WI	HEXSTRING [2]	wait indication, unit in second
Detailed Comments:		

Structured Type definitions

Structured Type Definition		
Type Name:	ACST	
Comments:	Auxiliary states (CC information element) GSM 04.08, 10.5.4.4	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb	EXTB	
sp3b	SP3B	
has	BITSTRING [2]	
mpas	BITSTRING [2]	
Detailed Comments:		

Structured Type Definition		
Type Name:	EFR_BCAP_3A_ETC	
Comments:	Speech Versions preferences as supported by the EFR mobile - see GSM 04.08/10.5.4.5/octet 3a etc/"speech version indication"	
Element Name	**Type Definition**	**Comments**
extb	EXTB	
cod	BITSTRING[1]	
spr2	BITSTRING[2]	
svi	BITSTRING [4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	BCAP	
Comments:	Bearer capability (CC information element) GSM 04.08, 10.5.4.5	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
rchr	BITSTRING [2]	
cs	BITSTRING [1]	
tm	BITSTRING [1]	
itc	BITSTRING [3]	
oct_3a_etc_1	EFR_BCAP_3A_ETC	
oct_3a_etc_2	EFR_BCAP_3A_ETC	
oct_3a_etc_3	EFR_BCAP_3A_ETC	
oct_3a_etc_4	EFR_BCAP_3A_ETC	
oct_3a_etc_5	EFR_BCAP_3A_ETC	
oct_3a_etc_6	EFR_BCAP_3A_ETC	
extb4	EXTB	
spb	SPB	
strc	BITSTRING [2]	
dplxm	BITSTRING [1]	
config	BITSTRING [1]	
nirr	BITSTRING [1]	
est	BITSTRING [1]	
extb5	EXTB	
accid	BITSTRING [2]	
ra	BITSTRING [2]	
sacp	BITSTRING [3]	
extb6	EXTB	
l1id	BITSTRING [2]	
uil1	BITSTRING [4]	
sb	BITSTRING [1]	
extb6a	EXTB	
nsb	BITSTRING [1]	
nb	BITSTRING [1]	
ndb	BITSTRING [1]	
ur	BITSTRING [4]	
extb6b	EXTB	
ir	BITSTRING [2]	
nictx	BITSTRING [1]	
nicrx	BITSTRING [1]	
pi	BITSTRING [3]	
extb6c	EXTB	
ce	BITSTRING [2]	
modemt	BITSTRING [5]	
extb7	EXTB	
l2id	BITSTRING [2]	
uil2	BITSTRING [5]	
Detailed Comments:		

Structured Type Definition		
Type Name:	BLOCKTYPE	
Comments:	Block type, GSM 04.12, 3.3.1	
Element Name	**Type Definition**	**Comments**
spare1	SPB	
lpd	BITSTRING[2]	
lb	BITSTRING[1]	
sequence_number	BITSTRING[4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CAU	
Comments:	Cause (CC information element) GSM 04.08, 10.5.4.11	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
cs	CS	
spb	SPB	
location	BITSTRING [4]	
extb3a	EXTB	
rec	BITSTRING [7]	
extb4	EXTB	
cau_class	BITSTRING [3]	
cau_va	BITSTRING [4]	
cau_di	OCTETSTRING [1..28]	
Detailed Comments:	&COMMON_U06	

Structured Type Definition		
Type Name:	CCCAP	
Comments:	Call Control Capabilities GSM 04.08, 10.5.4.5a	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
spr	BITSTRING [7]	
dtmf	BITSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CCD	
Comments:	Control channel description GSM 04.08, 10.5.2.11	
Element Name	**Type Definition**	**Comments**
spr1	BITSTRING [1]	
att	BITSTRING [1]	
babr	BITSTRING [3]	
ccch_con	BITSTRING [3]	
spr2	BITSTRING [5]	
bpm	BITSTRING [3]	
t3212	OCTETSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CCHD	
Comments:	Cell channel description GSM 04.08, 10.5.2.1	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
rfl	OCTETSTRING [16]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CD	
Comments:	Cell description GSM 04.08, 10.5.2.2	
Element Name	**Type Definition**	**Comments**
bcch_arfcn_h	BITSTRING [2]	
ncc	NCC	
bcc	BCC	
bcch_arfcn_l	BITSTRING [8]	
Detailed Comments:	The info element is two octets long.	

Structured Type Definition		
Type Name:	CDPN	
Comments:	Called party BCD number (CC information element) GSM 04.08, 10.5.4.7	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
tonnpi	TON_NPI	
digits	BCDN	
Detailed Comments:		

Structured Type Definition		
Type Name:	CDPS	
Comments:	Called party subaddress (CC information element) GSM 04.08, 10.5.4.8	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
subad	SUBAD	
Detailed Comments:		

Structured Type Definition		
Type Name:	CE	
Comments:	connection element	
Element Name	**Type Definition**	**Comments**
TorBothTorBothNT	B_2	
NTorBothTorBothNT	B_2	
Detailed Comments:		

Structured Type Definition		
Type Name:	CGPN	
Comments:	Calling party BCD number (CC information element) GSM 04.08, 10.5.4.9	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
tonnpi	TON_NPI	
pisi	PI_SI	
digits	BCDN	
Detailed Comments:		

Structured Type Definition		
Type Name:	CGPS	
Comments:	Calling party subaddress (CC information element) GSM 04.08, 10.5.4.10	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
subad	SUBAD	
Detailed Comments:		

Structured Type Definition		
Type Name:	CHD	
Comments:	Channel description GSM 04.08, 10.5.2.5	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
cht_schn	BITSTRING [5]	
tn	BITSTRING [3]	
tsc	BITSTRING [3]	
hch	BITSTRING [1]	
maio	BITSTRING [6]	
hsn	BITSTRING [6]	
spr	BITSTRING [2]	
arfcn	BITSTRING [10]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CHMOD	
Comments:	Channel mode, channel mode2 GSM 04.08, 10.5.2.6, 10.5.2.7	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
mode	B_8	
Detailed Comments:		

Structured Type Definition		
Type Name:	CHNEED	
Comments:	Channels needed GSM 04.08, 10.5.2.8	
Element Name	**Type Definition**	**Comments**
ch2	CHANNEL	
ch1	CHANNEL	
Detailed Comments:		

Structured Type Definition		
Type Name:	CNN	
Comments:	Connected number (CC information element) GSM 04.08, 10.5.4.13	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
tonnpi	TON_NPI	
pisi	PI_SI	
digits	BCDN	
Detailed Comments:		

Structured Type Definition		
Type Name:	CNS	
Comments:	Connected subaddress (CC information element) GSM 04.08, 10.5.4.14	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
subad	SUBAD	
Detailed Comments:		

Structured Type Definition		
Type Name:	CO	
Comments:	Cell options (BCCH / SACCH) GSM 04.08, 10.5.2.3	
Element Name	**Type Definition**	**Comments**
sprb	BITSTRING [1]	
pwrc	BITSTRING [1]	
dtx	BITSTRING [2]	
rlt	BITSTRING [4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	Component_T	
Comments:	Component for sending (downlink) GSM 04.80, 3.6	
Element Name	**Type Definition**	**Comments**
comp_part1	OCTETSTRING	
invokeId	OCTETSTRING [1]	
comp_part2	OCTETSTRING	
comp_part3	OCTETSTRING	
comp_part4	OCTETSTRING	
comp_part5	OCTETSTRING	
comp_part6	OCTETSTRING	
comp_part7	OCTETSTRING	
comp_part8	OCTETSTRING	
comp_part9	OCTETSTRING	
comp_part10	OCTETSTRING	
comp_part11	OCTETSTRING	
comp_part12	OCTETSTRING	
Detailed Comments:	This component is defined for the use in the downlink (from the test system to the IUT) direction.	

Structured Type Definition		
Type Name:	CPDATA	
Comments:	CP-User data element GSM 04.11, 8.1.4.1	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
rpack	RPACK	
rpdata	RPDATA	
rperr	RPERR	
rpsmma	RPSMMA	
Detailed Comments:	One of the 4 types of rpdu is contained in the CPDATA.	

Structured Type Definition		
Type Name:	CPHMS	
Comments:	Cipher mode setting GSM 04.08, 10.5.2.9	
Element Name	**Type Definition**	**Comments**
iei	IEI_4	
algid	BITSTRING [3]	
sc	BITSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CPHKSN	
Comments:	Ciphering key sequence number GSM 04.08, 10.5.1.2	
Element Name	**Type Definition**	**Comments**
sprb	BITSTRING [1]	
ks	BITSTRING [3]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CPH_RES	
Comments:	Cipher response GSM 04.08, 10.5.2.10	
Element Name	**Type Definition**	**Comments**
iei	IEI_4	
sprb	BITSTRING [3]	
cr	BITSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CSP	
Comments:	Cell selection parameters GSM 04.08, 10.5.2.4	
Element Name	**Type Definition**	**Comments**
crh	BITSTRING [3]	
mtmc	BITSTRING [5]	
acs	BITSTRING [1]	
neci	BITSTRING [1]	
ram	BITSTRING [6]	
Detailed Comments:		

Structured Type Definition		
Type Name:	CST	
Comments:	Call status (CC information element) GSM 04.08, 10.5.4.6	
Element Name	**Type Definition**	**Comments**
cs	CS	
csv	BITSTRING [6]	
Detailed Comments:		

Structured Type Definition		
Type Name:	FIE	
Comments:	Facility information element GSM 04.80, 3.6	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
components_1	Components	
components_t	Component_T	
Detailed Comments:	When sending normally only one component is sent, but when receiving any number of components can be received even though normally we are only interested in one component during the testing.	

Structured Type Definition		
Type Name:	FN	
Comments:	The time in reference to the frame numbering corresponding to the absolute frame number modulo 42432.	
Element Name	**Type Definition**	**Comments**
t1_	T1_	
t3	T3	
t2	T2	
Detailed Comments:	In STRT and RQR	

Structured Type Definition		
Type Name:	FRQCHS	
Comments:	Frequency channel sequence GSM 04.08, 10.5.2.12	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
larfcn	OCTETSTRING [1]	
incs	OCTETSTRING [8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	FRQL	
Comments:	Frequency list, frequency short list GSM 04.08, 10.5.2.5, 10.5.2.6	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
fl	OCTETSTRING	
Detailed Comments:	frequency short list has a fixed length of 10 octets and does not contain the length indicator.	

Structured Type Definition		
Type Name:	FRQPARA	
Comments:	Parameters for Description of basic physical channel in frequency domain.	
Element Name	**Type Definition**	**Comments**
hch	BITSTRING[1]	
maio	BITSTRING [6]	
hsn	BITSTRING [6]	
spr	BITSTRING [2]	
arfcn	BITSTRING [10]	
maclength	LENGTH	
mac_8n	BITSTRING [8]	
mac_7n	BITSTRING [8]	
mac_6n	BITSTRING [8]	
mac_5n	BITSTRING [8]	
mac_4n	BITSTRING [8]	
mac_3n	BITSTRING [8]	
mac_2n	BITSTRING [8]	
mac_1n	BITSTRING [8]	
flst	FRQL	
Detailed Comments:		

Structured Type Definition		
Type Name:	HLCMP	
Comments:	High layer compatibility (CC information element) GSM 04.08, 10.5.4.16, ITU Q.931	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
cs	BITSTRING [2]	
in	BITSTRING [3]	
pmpp	BITSTRING [2]	
extb4	EXTB	
hlci	BITSTRING [7]	
extb4a	EXTB	
ehlci	BITSTRING [7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	HOACC_PARA	
Comments:	Handover Access parameters	
Element Name	**Type Definition**	**Comments**
num_bursts	OCTETSTRING[1]	
min_mspwr	OCTETSTRING[1]	
max_mspwr	OCTETSTRING[1]	
min_msta	BITSTRING[8]	
max_msta	BITSTRING[8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	IARESTOCT	
Comments:		
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iaroct1	OCTETSTRING[1..11]	
iaroct2	IARESTOCT2	
Detailed Comments:		

Structured Type Definition		
Type Name:	IARESTOCT2	
Comments:		
Element Name	**Type Definition**	**Comments**
p	BITSTRING[2]	
frqparalen	BITSTRING[6]	
spbt1	BITSTRING[2]	
maio	BITSTRING[6]	
ma	OCTETSTRING[0..8]	
spbt2	OCTETSTRING[0..8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	KPF (keypad facility)	
Comments:	keypad facility GSM 04.08 clause 10.5.4.17	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
extb	EXTB	
kpf_info	IA5String [1]	
Detailed Comments:	Is extb needed?	

Structured Type Definition		
Type Name:	L1HD	
Comments:	Layer 1 Header	
Element Name	**Type Definition**	**Comments**
spr1	BITSTRING[3]	
mspwrlvl	BITSTRING[5]	
spr2	BITSTRING[1]	
ta	BITSTRING[7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	LAI	
Comments:	Location area identification GSM 04.08, 10.5.1.3	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
mcc	OCTETSTRING [2]	
mnc	OCTETSTRING [1]	
lac	OCTETSTRING [2]	
Detailed Comments:		

Structured Type Definition		
Type Name:	LLCMP	
Comments:	Low layer compatibility (CC information element) GSM 04.08, 10.5.4.18 and ETS DE/SPS-5034-1 (update of ETS 300 102-1)	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
cs	BITSTRING [2]	
itc	BITSTRING [5]	
extb3a	EXTB	
negind	BITSTRING[1]	
spb3a	SP6B	
extb4	EXTB	
tm	BITSTRING [2]	
itr	BITSTRING [5]	
extb4_1	EXTB	
rateml	BITSTRING [7]	
extb5	EXTB	
l1id	BITSTRING [2]	
uil1	BITSTRING [5]	
extb5a	EXTB	
sb	BITSTRING [1]	
neg	BITSTRING [1]	
ur	BITSTRING [5]	
extb5b1	EXTB	
ir	BITSTRING [2]	
nictx	BITSTRING [1]	
nicrx	BITSTRING [1]	
fctx	BITSTRING [1]	
fcrx	BITSTRING [1]	
spb5b1	SPB	
extb5b2	EXTB	
hdrb	BITSTRING[1]	
mfs	BITSTRING[1]	
mode	BITSTRING[1]	
llineg	BITSTRING[1]	
ass	BITSTRING[1]	
ibob	BITSTRING[1]	
spb5b2	SPB	
extb5c	EXTB	
nsb	BITSTRING [2]	
ndb	BITSTRING [2]	
pi	BITSTRING [3]	
extb5d	EXTB	
dplxm	BITSTRING [1]	
modemt	BITSTRING [6]	
extb6	EXTB	
l2id	BITSTRING [2]	
uil2	BITSTRING [5]	
extb6a1	EXTB	
modl2	BITSTRING[2]	
spb6a1	SP3B	
q933	BITSTRING [2]	
extb6a2	EXTB	
usl2pi	BITSTRING[7]	
extb6b	EXTB	
ws	BITSTRING[7]	
extb7	EXTB	
l3id	BITSTRING [2]	
uil3	BITSTRING [5]	
extb7a1	EXTB	
modl3	BITSTRING[2]	
spb7a1	SP5B	

extb7a2	EXTB	
usl3pi	BITSTRING[7]	
extb7b	EXTB	
spb7b	SP3B	
dps	BITSTRING [4]	
extb7c	EXTB	
pws	BITSTRING[7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	LUT	
Comments:	Location updating type GSM 04.08, 10.5.3.5	
Element Name	**Type Definition**	**Comments**
foreq	BITSTRING [1]	
sprb	BITSTRING [1]	
lut	BITSTRING [2]	
Detailed Comments:		

Structured Type Definition		
Type Name:	MA	
Comments:	Mobile allocation GSM 04.08, 10.5.2.21	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
mac_8n	BITSTRING [8]	
mac_7n	BITSTRING [8]	
mac_6n	BITSTRING [8]	
mac_5n	BITSTRING [8]	
mac_4n	BITSTRING [8]	
mac_3n	BITSTRING [8]	
mac_2n	BITSTRING [8]	
mac_1n	BITSTRING [8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	MI	
Comments:	Mobile identity GSM 04.08, 10.5.1.4	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
idigit_1	BITSTRING [4]	
oei	BITSTRING [1]	
toi	BITSTRING [3]	
idigits_other	OCTETSTRING [0..7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	MODEMTYPE	
Comments:		
Element Name	**Type Definition**	**Comments**
V22bisOrV26ter	B_5	
Detailed Comments:		

Structured Type Definition		
Type Name:	MSCLM1	
Comments:	mobile station classmark 1 GSM 04.08, 10.5.1.5	
Element Name	**Type Definition**	**Comments**
spr1	BITSTRING [1]	
rl	BITSTRING [2]	
spr2	BITSTRING [1]	
a5_1	BITSTRING [1]	
rfpc	BITSTRING [3]	
Detailed Comments:	2 octets long	

Structured Type Definition		
Type Name:	MSCLM2	
Comments:	mobile station classmark 2 GSM 04.08, 10.5.1.6	
Element Name	**Type Definition**	**Comments**
iel	LENGTH	
spr1	BITSTRING [1]	
rl	BITSTRING [2]	
spr2	BITSTRING [1]	
a5_1	BITSTRING [1]	
rfpc	BITSTRING [3]	
spr3	BITSTRING [1]	
psc	BITSTRING [1]	
sssi	BITSTRING [2]	
smc	BITSTRING [1]	
spr4	BITSTRING [2]	
fc	BITSTRING [1]	
cm3	BITSTRING [1]	
spr5	BITSTRING [5]	
a5_3	BITSTRING [1]	
a5_2	BITSTRING [1]	
Detailed Comments:	4 octets long	

Structured Type Definition		
Type Name:	MSCLM3	
Comments:	Mobile station classmark 3 GSM 04.08, 10.5.1.7	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
spr1	BITSTRING [4]	
a5_7	BITSTRING [1]	
a5_6	BITSTRING [1]	
a5_5	BITSTRING [1]	
a5_4	BITSTRING [1]	
spr2	OCTETSTRING [11]	
Detailed Comments:	The info element has 14 octets	

Structured Type Definition		
Type Name:	MSRR	
Comments:	measurement results GSM 04.08, 10.5.2.20	
Element Name	**Type Definition**	**Comments**
ba_used	BITSTRING [1]	
dtx_used	BITSTRING [1]	
rxlev_fsc	RXLEV	
spr1	BITSTRING [1]	
meas_valid	BITSTRING [1]	
rxlev_ssc	RXLEV	
spr2	BITSTRING [1]	
rxqual_fsc	BITSTRING [3]	
rxqual_ssc	BITSTRING [3]	
no_nc	BITSTRING [3]	
rxlev_nc1	RXLEV	
bcchfrq_nc1	BCCHFRQ	
bsic_nc1	BSIC	
rxlev_nc2	RXLEV	
bcchfrq_nc2	BCCHFRQ	
bsic_nc2	BSIC	
rxlev_nc3	RXLEV	
bcchfrq_nc3	BCCHFRQ	
bsic_nc3	BSIC	
rxlev_nc4	RXLEV	
bcchfrq_nc4	BCCHFRQ	
bsic_nc4	BSIC	
rxlev_nc5	RXLEV	
bcchfrq_nc5	BCCHFRQ	
bsic_nc5	BSIC	
rxlev_nc6	RXLEV	
bcchfrq_nc6	BCCHFRQ	
bsic_nc6	BSIC	
Detailed Comments:	The info element has a fixed length of 16 octets.	

Structured Type Definition		
Type Name:	MTDIF	
Comments:	Mobile time difference GSM 04.08, 10.5.1.21a	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
value	OCTETSTRING [3]	
Detailed Comments:		

Structured Type Definition		
Type Name:	NCD	
Comments:	Neighbour cells description GSM 04.08, 10.5.2.22	
Element Name	**Type Definition**	**Comments**
rfl2	BITSTRING [2]	
extind	BITSTRING [1]	
baind	BITSTRING [1]	
rfl4	BITSTRING [4]	
rfl	OCTETSTRING [15]	
Detailed Comments:		

Structured Type Definition		
Type Name:	NCD2	
Comments:	Neighbour cells description2 GSM 04.08, 10.5.2.22b	
Element Name	**Type Definition**	**Comments**
b128	BITSTRING [1]	
mbrpt	BITSTRING [2]	
baind	BITSTRING [1]	
b122_124	BITSTRING [3]	
rfl1	BITSTRING [1]	
rfl	OCTETSTRING [15]	
Detailed Comments:		

Structured Type Definition		
Type Name:	PCMD	
Comments:	Power command and access type GSM 04.08, 10.5.2.28, 10.5.2.28a	
Element Name	**Type Definition**	**Comments**
sprb	BITSTRING [3]	
pl	BITSTRING [5]	
Detailed Comments:		

Structured Type Definition		
Type Name:	PI	
Comments:	Progress indicator GSM 04.08, 10.5.4.21	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb3	EXTB	
cs	BITSTRING [2]	
spb	SPB	
loc	BITSTRING [4]	
extb4	EXTB	
prd	BITSTRING [7]	
Detailed Comments:	total 4 octets	

Structured Type Definition		
Type Name:	PI_SI	
Comments:	Presentation indicator & screening indicator GSM 04.08, 10.5.4.9, 10.5.4.13	
Element Name	**Type Definition**	**Comments**
extb	EXTB	
pi	BITSTRING [2]	
sp3b	SP3B	
si	BITSTRING [2]	
Detailed Comments:		

Structured Type Definition		
Type Name:	PM	
Comments:	Page mode GSM 04.08, 10.5.2.26	
Element Name	**Type Definition**	**Comments**
sprb	B_2	
pgm	B_2	
Detailed Comments:		

Structured Type Definition		
Type Name:	RACHCP	
Comments:	Rach control parameters GSM 04.08, 10.5.2.29	
Element Name	**Type Definition**	**Comments**
maxrtx	BITSTRING [2]	
txint	BITSTRING [4]	
cba	BITSTRING [1]	
re	BITSTRING [1]	
acc_2	BITSTRING [5]	
ec	BITSTRING [1]	
acc_1	BITSTRING [10]	
Detailed Comments:	The info element has a fixed length of 3 octets.	

Structured Type Definition		
Type Name:	RPACK	
Comments:	SMS RP ACKNOWLEDGEMENT ms <-> n GSM 04.11, 7.3.3	
Element Name	**Type Definition**	**Comments**
sprb	SP5B	
rpmti	MTI	
rpmr	MR	
Detailed Comments:		

Structured Type Definition		
Type Name:	RPCAU	
Comments:	RP user data element GSM 04.11, 8.2.5.4	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
extb2	EXTB	
rpcau_class	BITSTRING [3]	
rpcau_va	BITSTRING [4]	
rpcau_di	OCTETSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	RPDATA	
Comments:	SMS RP DATA ms <-> n GSM 04.11, 7.3.1	
Element Name	**Type Definition**	**Comments**
sprb	SP5B	
rpmti	MTI	
rpmr	MR	
rpOaddr	CDPN	
rpDaddr	CDPN	
rpusrdat	RPUSRDAT	
Detailed Comments:		

Structured Type Definition		
Type Name:	RPERR	
Comments:	SMS RP ERROR ms <-> n GSM 04.11, 7.3.4	
Element Name	**Type Definition**	**Comments**
sprb	SP5B	
rpmti	MTI	
rpmr	MR	
rpcau	RPCAU	
rpusrdat	RPUSRDAT	
Detailed Comments:		

Structured Type Definition		
Type Name:	RPSMMA	
Comments:	SMS RP SMMA ms -> n GSM 04.11, 7.3.2	
Element Name	**Type Definition**	**Comments**
sprb	SP5B	
rpmti	MTI	
rpmr	MR	
Detailed Comments:	SM memory available	

Structured Type Definition		
Type Name:	RPUSRDAT	
Comments:	RP user data element GSM 04.11, 8.2.5.3, GSM 03.40, 9.2.2	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
tpdeliver	SMDLVR	
tpsubmit	SMSBMT	
tpstatus_rpt	SMST_RPT	
tpcommand	SMCMD	
tpdlvr_sbmt_rpt	SMDLVR_RPT	
Detailed Comments:	One of the six tpdu is contained in the RPUSRDATA. Since the structures of the messages SMS-Deliver-Report and SMS-Submit-Report are identical, they have been combined to tpdlvr_sbmt_rpt and therefore only five tp message types exist in this type definition.	

Structured Type Definition		
Type Name:	RQR	
Comments:	Request reference GSM 04.08, 10.5.2.30	
Element Name	**Type Definition**	**Comments**
ra	BITSTRING [8]	
fn	FN	
Detailed Comments:		

Structured Type Definition		
Type Name:	SI3RO	
Comments:	SI3 Rest Octets GSM 04.08, 10.5.2.34	
Element Name	**Type Definition**	**Comments**
pi	BITSTRING [1]	
cbq	BITSTRING [1]	
cro	BITSTRING [6]	
to	BITSTRING [3]	
pt	BITSTRING [5]	
poi	BITSTRING [1]	
po	BITSTRING [2]	
ti	BITSTRING [1]	
ecsc	BITSTRING [1]	
spr1	BITSTRING [3]	
spr2	BITSTRING [8]	
Detailed Comments:	The info element has a fixed length of 4 octets.	

Structured Type Definition		
Type Name:	SI4RO	
Comments:	SI4 Rest Octets GSM 04.08, 10.5.2.35	
Element Name	**Type Definition**	**Comments**
pi	BITSTRING [1]	
cbq	BITSTRING [1]	
cro	BITSTRING [6]	
to	BITSTRING [3]	
pt	BITSTRING [5]	
poi	BITSTRING [1]	
po	BITSTRING [2]	
spr1	BITSTRING [5]	
spr2	BITSTRING [8]	
spr3	BITSTRING [8]	
spr4	BITSTRING [8]	
spr5	BITSTRING [8]	
spr6	BITSTRING [8]	
spr7	BITSTRING [8]	
spr8	BITSTRING [8]	
Detailed Comments:	The info element has a fixed length of 10 octets.	

Structured Type Definition		
Type Name:	SI7RO	
Comments:	SI7 Rest Octets GSM 04.08, 10.5.2.36	
Element Name	**Type Definition**	**Comments**
pi	BITSTRING [1]	
cbq	BITSTRING [1]	
cro	BITSTRING [6]	
to	BITSTRING [3]	
pt	BITSTRING [5]	
poi	BITSTRING [1]	
po	BITSTRING [2]	
spr1	BITSTRING [5]	
spr2	BITSTRING [8]	
spr3	BITSTRING [8]	
spr4	BITSTRING [8]	
spr5	BITSTRING [8]	
spr6	BITSTRING [8]	
spr7	BITSTRING [8]	
spr8	BITSTRING [8]	
spr9	BITSTRING [8]	
spr10	BITSTRING [8]	
spr11	BITSTRING [8]	
spr12	BITSTRING [8]	
spr13	BITSTRING [8]	
spr14	BITSTRING [8]	
spr15	BITSTRING [8]	
spr16	BITSTRING [8]	
spr17	BITSTRING [8]	
spr18	BITSTRING [8]	
Detailed Comments:	The info element has a fixed length of 20 octets.	

Structured Type Definition		
Type Name:	SIGNAL	
Comments:	Signal (CC information element) GSM 04.08, 10.5.4.23	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
sigv	BITSTRING [8]	
Detailed Comments:		

Structured Type Definition		
Type Name:	SERIAL_NUMBER	
Comments:	Serial number for SMSCB, GSM 3.41, 9.3.2	
Element Name	**Type Definition**	**Comments**
gs	BITSTRING[2]	
message_code	BITSTRING[10]	
update_number	BITSTRING[4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMCMD	
Comments:	SMS COMMAND, ms - > n. GSM 03.40, 9.2.2.4	
Element Name	**Type Definition**	**Comments**
sprb1	SP2B	
srr	BITSTRING[1]	
sprb2	SP3B	
mti	BITSTRING [2]	
mr	MR	
pid	TPPID	
ct	OCTETSTRING [1]	
mn	OCTETSTRING [1]	
da	TPA	
cdl	LENGTH	
cd	TPCD	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMDLVR	
Comments:	SMS DELIVER, n - > ms. GSM 03.40, 9.2.2.1	
Element Name	**Type Definition**	**Comments**
rp	BITSTRING [1]	
udhi	BITSTRING [1]	
sri	BITSTRING [1]	
sprb2	SP2B	
mms	BITSTRING [1]	
mti	BITSTRING [2]	
oa	TPA	
pid	TPPID	
dcs	TPDCS	
scts	TPSCTS	
udl	LENGTH	
ud	TPUD	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMDLVR_RPT	
Comments:	SMS DELIVER or SUBMIT REPORT contained in the RP ERROR PDU, n <-> ms. GSM 03.40, 9.2.2.1a, 9.2.2.2a.	
Element Name	**Type Definition**	**Comments**
sprb1	SP6B	
mti	BITSTRING [2]	
fcs	FCS	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMSBMT	
Comments:	SMS SUBMIT, ms -> n. GSM 03.40, 9.2.2.2	
Element Name	**Type Definition**	**Comments**
rp	BITSTRING [1]	
udhi	BITSTRING [1]	
srr	BITSTRING [1]	
vpf	BITSTRING [2]	
rd	BITSTRING [1]	
mti	BITSTRING [2]	
mr	MR	
da	TPA	
pid	TPPID	
dcs	TPDCS	
vp1	OCTETSTRING [1]	
vp7	TPSCTS	
udl	LENGTH	
ud	TPUD	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMSCBpack	
Comments:	SMS cell broadcasting packing data n -> ms. GSM 03.38, 6.1.2.2.1	
Element Name	**Type Definition**	**Comments**
block1	OCTETSTRING [16]	
block2	OCTETSTRING [22]	
block3	OCTETSTRING [22]	
block4	OCTETSTRING [22]	
Detailed Comments:		

Structured Type Definition		
Type Name:	SMST_RPT	
Comments:	SMS STATUS REPORT, n - > ms. GSM 03.40, 9.2.2.3	
Element Name	**Type Definition**	**Comments**
sprb1	SP5B	
mms	BITSTRING [1]	
mti	BITSTRING [2]	
mr	MR	
ra	TPA	
scts	TPSCTS	
dt	TPSCTS	
st	TPST	
Detailed Comments:		

Structured Type Definition		
Type Name:	SSVI	
Comments:	SS version indicator GSM 04.08, 10.5.4.24	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
ssv	OCTETSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	STRT	
Comments:	Starting time GSM 04.08, 10.5.2.38	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
fn	FN	
Detailed Comments:		

Structured Type Definition		
Type Name:	SUBAD	
Comments:	Subaddress GSM 04.08, 10.5.4.8, 10.5.4.10, 10.5.4.14	
Element Name	**Type Definition**	**Comments**
extb	EXTB	
tos	BITSTRING [3]	
oei	BITSTRING [1]	
sp3b	SP3B	
si	OCTETSTRING [2..23]	
Detailed Comments:		

Structured Type Definition		
Type Name:	SYNCHI	
Comments:	Synchronization indication GSM 04.08, 10.5.2.39	
Element Name	**Type Definition**	**Comments**
iei	IEI_4	
nci	B_1	
rot	B_1	
si	B_2	
Detailed Comments:		

Structured Type Definition		
Type Name:	TA	
Comments:	Timing advance GSM 04.08, 10.5.2.40	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
sprb	BITSTRING [2]	
value	BITSTRING [6]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TDIF	
Comments:	Time difference GSM 04.08, 10.5.1.41	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
value	OCTETSTRING [1]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TI	
Comments:	Transaction identifier GSM 04.08, 10.1	
Element Name	**Type Definition**	**Comments**
ti_f	BITSTRING [1]	
ti_v	TI_V	
Detailed Comments:		

Structured Type Definition		
Type Name:	TMSI	
Comments:	Temporary GSM 04.08, 10.5.2.42	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
tmsi_val	OCTETSTRING [4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TON_NPI	
Comments:	Type of number and numbering plan GSM 04.08, 10.5.4.7, 10.5.4.9, 10.5.4.13, GSM 03.40, 9.1.2.5	
Element Name	**Type Definition**	**Comments**
extb	EXTB	
ton	BITSTRING [3]	
npi	BITSTRING [4]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TPA	
Comments:	TP address GSM 03.40, 9.1.2.5	
Element Name	**Type Definition**	**Comments**
iel	LENGTH	
tonnpi	TON_NPI	
digits	BCDN	
Detailed Comments:		

Structured Type Definition		
Type Name:	TPDCS	
Comments:	SMS data coding scheme GSM 03.38, 4, 5	
Element Name	**Type Definition**	**Comments**
cg	BITSTRING [4]	
code	BITSTRING [4]	
Detailed Comments:	Identifying the coding scheme within the TP user data.	

Structured Type Definition		
Type Name:	TPPID	
Comments:	TP protocol identifier GSM 03.40, 9.2.3.9	
Element Name	**Type Definition**	**Comments**
type	BITSTRING [2]	
value	BITSTRING [6]	
Detailed Comments:		

Structured Type Definition		
Type Name:	TPST	
Comments:	TP status GSM 03.40, 9.2.3.15	
Element Name	**Type Definition**	**Comments**
sprb1	SPB	
value	BITSTRING [7]	
Detailed Comments:		

Structured Type Definition		
Type Name:	UNKWN	
Comments:	unknown IE	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
contents	OCTETSTRING	
Detailed Comments:		

Structured Type Definition		
Type Name:	UU	
Comments:	User-user GSM 04.08, 10.5.4.25	
Element Name	**Type Definition**	**Comments**
iei	IEI_8	
iel	LENGTH	
uupd	BITSTRING [8]	
uui	OCTETSTRING [1..128]	
Detailed Comments:	In SETUP, ALERTING, CONNECT, DISCONNECT, RELEASE and RELEASE COMPLETE messages the uui length is of 0 - 32 bytes. In USER INFORMATION messages the uui length is of 1 - 128.	

ASN1 type definitions

ASN.1 Type Definition	
Type Name:	Component
Comments:	ASN1_Encoding: BER
Type Definition	

```
CHOICE {
   registerSSComponents                          RegisterSS_Components,
   eraseSSComponents                              EraseSS_Components,
   activateSSComponents                          ActivateSS_Components,
   deactivateSSComponents                      DeactivateSS_Components,
   interrogateSSComponents                     InterrogateSS_Components,
   notifySSComponents                             NotifySS_Components,
   registerPasswordComponents                RegisterPassword_Components,
   getPasswordComponents                       GetPassword_Components,
   processUnstructuredSSDataComponents     ProcessUnstructuredSSData_Components,
   forwardCheckSSIndicationComponents        ForwardCheckSSIndication_Components,
   processUnstructuredSSRequestComponents ProcessUnstructuredSSRequest_Components,
   unstructuredSSRequestComponents           UnstructuredSSRequest_Components,
   unstructuredSSNotifyComponents             UnstructuredSSNotify_Components,
   forwardCUGInfoComponents                      ForwardCUGInfo_Components,
   splitMPTYComponents                                SplitMPTY_Components,
   retrieveMPTYComponents                        RetrieveMPTY_Components,
   holdMPTYComponents                             HoldMPTY_Components,
   buildMPTYComponents                            BuildMPTY_Components,
   forwardChargeAdviceComponents           ForwardChargeAdvice_Components,
   generalComponents                                  General_Components
}
```

Detailed Comments:	Plural components as each type represents invoke, return result, return error etc.

ASN.1 Type Definition	
Type Name:	Components
Comments:	GSM 04.80, 3.6
Type Definition	

```
SET OF Component
```

Detailed Comments:	ASN.1 transfer encoding rules: BER is not wholy used for the type Components. The contents of Components, without the octets encoding the tag and the length of SET OF, is carried as the Components value.

ASN.1 Type Definition

Type Name: ActivateSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       activateSS_InvokeComp          [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (12),
                                ss_ForBS                    SS_ForBS_Code },

       activateSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                        SEQUENCE {
                                        localValue                          INTEGER (12),
                                        ss_Info              SS_Info        }      } ,

       activateSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| dataMissing

| unexpectedDataValue

| negativePW_Check

| numberOfPW_AttemptsViolation) },
              callBarredErr                        [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_ErrorStatusErr             [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_SubscriptionViolationeErr [3] IMPLICIT SEQUENCE { invokeID       InvokeIDType,

errorCode     ErrorLocalValues (ss_SubscriptionViolation),

parameter      SS_SubscriptionOption },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                 [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       activateSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m (15 - 30 s)

ASN.1 Type Definition

Type Name: BuildMPTY_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       buildMPTY_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (124)  },

       buildMPTY_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                         localValue                          INTEGER (124) }
} ,

       buildMPTY_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSS_Operation

| ss_NotAvailable

| maxNumberOfMPTY_ParticipantsExceeded

| resourcesNotAvailable ) },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       buildMPTY_RejectComp             [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_buildMPTY (5 - 30 s)

ASN.1 Type Definition

Type Name: DeactivateSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       deactivateSS_InvokeComp          [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (13),
                                ss_ForBS                    SS_ForBS_Code },

       deactivateSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                   InvokeIDType,
                                 result                        SEQUENCE {
                                        localValue                          INTEGER (13),
                                        ss_Info              SS_Info        }      } ,

       deactivateSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| dataMissing

| unexpectedDataValue

| negativePW_Check

| numberOfPW_AttemptsViolation) },
              callBarredErr                       [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_ErrorStatusErr             [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_SubscriptionViolationeErr [3] IMPLICIT SEQUENCE { invokeID       InvokeIDType,

errorCode     ErrorLocalValues (ss_SubscriptionViolation),

parameter      SS_SubscriptionOption },
              systemFailureErr                 [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       deactivateSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: EraseSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       eraseSS_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (11),
                                ss_ForBS                    SS_ForBS_Code },

       eraseSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                        localValue                        INTEGER (11),
                                        ss_Info              SS_Info    }             } ,

       eraseSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues (  bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| dataMissing

| unexpectedDataValue ) },
              callBarredErr                       [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              systemFailureErr                 [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },


       eraseSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: ForwardChargeAdvice_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       forwardChargeAdvice_InvokeComp           [1] IMPLICIT SEQUENCE {
                           invokeID                                        InvokeIDType,
                           localValue                                       INTEGER (125),
                           forwardChargeAdviceArg              SEQUENCE {
                                                ss_Code                       [0] SS_Code,
                                                chargingInformation     [1]  SEQUENCE{
                                                      e1       [1] INTEGER
(0..max10TimesUnitsPerTime)                        OPTIONAL,
                                                      e2       [2]  INTEGER
(0..max10TimesTimeInterval)                         OPTIONAL,
                                                      e3       [3]  INTEGER
(0..max100TimesScalingFactor)                      OPTIONAL,
                                                      e4       [4] INTEGER
(0..max10TimesIncrement)                              OPTIONAL,
                                                      e5       [5]  INTEGER
(0..max10TimesIncrementPerDataInterval)    OPTIONAL,
                                                      e6       [6] INTEGER
(0..maxNumberOfSegmentsPerDataInterval) OPTIONAL,
                                                      e7       [7]  INTEGER
(0..max10TimesInitialTime)                           OPTIONAL } } },

       forwardChargeAdvice_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                           invokeID                    InvokeIDType,
                           result                        SEQUENCE {
                           localValue                   INTEGER (125)  }    } ,

       forwardChargeAdvice_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_aoc = 1 - 40 s

ASN.1 Type Definition

Type Name: ForwardCheckSSIndication_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       forwardCheckSSIndication_InvokeComp           [1] IMPLICIT SEQUENCE {
                              invokeID                      InvokeIDType,
                              localValue                     INTEGER (38)    },

      forwardCheckSSIndication_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments:

ASN.1 Type Definition

Type Name: ForwardCUGInfo_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       forwardCUGInfo_InvokeComp           [1] IMPLICIT SEQUENCE {
                              invokeID                      InvokeIDType,
                              localValue                     INTEGER (120),
                              forwardCUGInfo_Arg             SEQUENCE{
                                       cug_Index              [0]  IMPLICIT CUG_Index
OPTIONAL,
                                       suppressPrefCUG   [1]  IMPLICIT NULL           OPTIONAL,
                                       suppressOA             [2]  IMPLICIT NULL
OPTIONAL } },

        forwardCUGInfo_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments:

ASN.1 Type Definition

Type Name: General_Components

Comments: Non specified components must match this type definition.

Type Definition

```
CHOICE {
  general_InvokeComp          [1] IMPLICIT General_InvokeComponent,
  general_ReturnResultComp [2] IMPLICIT General_ReturnResultComponent,
  general_ReturnErrorComp  [3] IMPLICIT General_ReturnErrorComponent,
  general_RejectComp           [4] IMPLICIT RejectComponent }


-- This is the General InvokeComponent --
General_InvokeComponent ::= SEQUENCE {
  invokeID                     InvokeIDType,
  linked_ID       [0] IMPLICIT InvokeIDType OPTIONAL,
  operation_value              Operation,
  argument                     ANY     OPTIONAL }

-- This is the General ReturnResultComponent --
General_ReturnResultComponent ::= SEQUENCE {
  invokeID       InvokeIDType,
  valueAndResult SEQUENCE {
                    operation_value Operation,
                    result          ANY     } OPTIONAL }

-- This is the General ReturnErrorComponent --
General_ReturnErrorComponent ::= SEQUENCE {
  invokeID InvokeIDType,
  error    ANY }
```

Detailed Comments:

ASN.1 Type Definition

Type Name: GetPassword_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       getPassword_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                linkedID                       [0] IMPLICIT InvokeIDType,
                                localValue                     INTEGER (18),
                                guidanceInfo                ENUMERATED {
                                                                    enterPW  (0),
                                                                    enterNewPW  (1),
                                                                    enterNewPW_Again  (2) }
},

       getPassword_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                        localValue                        INTEGER (18),
                                        currentPassword             Password }            } ,


       getPassword_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: HoldMPTY_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       holdMPTY_InvokeComp           [1] IMPLICIT SEQUENCE {
                                 invokeID                      InvokeIDType,
                                 localValue                     INTEGER (123)  },

       holdMPTY_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                  invokeID                    InvokeIDType,
                                  result                         SEQUENCE {
                                          localValue                          INTEGER (123) }
} ,

       holdMPTY_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSS_Operation

| facilityNotSupported ) },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       holdMPTY_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_holdMPTY (5 - 30 s)

ASN.1 Type Definition

Type Name: InterrogateSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       interrogateSS_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (14),
                                ss_ForBS                    SS_ForBS_Code },

       interrogateSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                   InvokeIDType,
                                 result                        SEQUENCE {
                                        localValue                          INTEGER (14),
                                        interrogateSS_Res             CHOICE {
                                             ss_Status                       [0] IMPLICIT
SS_Status,
                                             basicServiceGroupList   [2] IMPLICIT
BasicServiceGroupList,
                                             forwardingFeatureList   [3] IMPLICIT
ForwardingFeatureList,
                                             cli_RestrictionInfo         [4] IMPLICIT SEQUENCE {
                                                                      ss_Status
SS_Status,
                                                                      cliRestrictionOption
CliRestrictionOption    OPTIONAL } } } } ,

       interrogateSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| ss_NotAvailable

| dataMissing

| unexpectedDataValue) },
              callBarredErr                       [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },


       interrogateSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: NotifySS_Components
Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       notifySS_InvokeComp           [1] IMPLICIT SEQUENCE {
                                 invokeID                      InvokeIDType,
                                 localValue                     INTEGER (16),
                                 notifySS_Arg              SEQUENCE{
                                         ss_Code                        [1] IMPLICIT SS_Code
OPTIONAL,
                                         ss_Status                      [4] IMPLICIT SS_Status
OPTIONAL,
                                         ss_Notification              [5] IMPLICIT OCTET STRING
(SIZE (1)) OPTIONAL,
                                         callIsWaiting_Indicator   [14] IMPLICIT NULL
OPTIONAL,
                                         callOnHold_Indicator     [15] IMPLICIT ENUMERATED {

callRetrieved (0),

callOnHold (1) }         OPTIONAL,
                                         mpty_Indicator               [16] IMPLICIT NULL
OPTIONAL,
                                         cug_Index                      [17]  IMPLICIT CUG_Index
OPTIONAL,
                                         clirSuppressionRejected [18]  IMPLICIT NULL
OPTIONAL }    },

      notifySS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments:

ASN.1 Type Definition

Type Name: ProcessUnstructuredSSData_Components
Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       processUnstructuredSSData_InvokeComp           [1] IMPLICIT SEQUENCE {
                                 invokeID                       InvokeIDType,
                                 localValue                      INTEGER (19),
                                 ss_UserData               SS_UserData },

       processUnstructuredSSData_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                  invokeID                     InvokeIDType,
                                  result                          SEQUENCE {
                                         localValue                          INTEGER (19),
                                         ss_UserData                   SS_UserData   }  } ,

       processUnstructuredSSData_ReturnErrorComp   CHOICE {
              errorCodes                         [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues (unexpectedDataValue) },
              systemFailureErr                 [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       processUnstructuredSSData_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: ProcessUnstructuredSSRequest_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       processUnstructuredSSRequest_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (59),
                                ussd_Arg                    USSD_Arg },

       processUnstructuredSSRequest_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                        localValue                        INTEGER (59),
                                        ussd_Res                       USSD_Res}            }
,

       processUnstructuredSSRequest_ReturnErrorComp   CHOICE {
              errorCodes                         [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues (dataMissing

| unexpectedDataValue

| unknownAlphabet) },
              callBarredErr                       [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              systemFailureErr                 [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       processUnstructuredSSRequest_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: RegisterPassword_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       registerPassword_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                     InvokeIDType,
                                localValue                    INTEGER (17),
                                ss_Code                     SS_Code      },

       registerPassword_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                   InvokeIDType,
                                 result                        SEQUENCE {
                                        localValue                        INTEGER (17),
                                        newPassword                Password    }         } ,

       registerPassword_ReturnErrorComp   CHOICE {
              errorCodes                              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( dataMissing

| unexpectedDataValue

| negativePW_Check

| numberOfPW_AttemptsViolation) },
              callBarredErr                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode      ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_SubscriptionViolationeErr  [3] IMPLICIT SEQUENCE { invokeID    InvokeIDType,

errorCode      ErrorLocalValues (ss_SubscriptionViolation),

parameter      SS_SubscriptionOption },
              systemFailureErr                    [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter     NetworkResource},
              pw_RegistrationFailureErr       [3] IMPLICIT SEQUENCE { invokeID    InvokeIDType,

errorCode     ErrorLocalValues ( pw_RegistrationFailure ),

parameter     PW_RegistrationFailureCause }   },

       registerPassword_RejectComp      [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: RegisterSS_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       registerSS_InvokeComp          [1] IMPLICIT SEQUENCE {
                              invokeID                     InvokeIDType,
                              localValue                    INTEGER (10),
                              registerSS_Arg          SEQUENCE{
                                      ss_Code                        SS_Code,
                                      basicService                   BasicServiceCode
OPTIONAL,
                                      forwardedToNumber         [4] IMPLICIT AddressString
OPTIONAL,
                                      forwardedToSubaddress   [6] IMPLICIT
ISDN_SubaddressString       OPTIONAL,
                                      noReplyConditionTime      [5] IMPLICIT
NoReplyConditionTime        OPTIONAL     }   },

       registerSS_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                invokeID                    InvokeIDType,
                                result                         SEQUENCE {
                                        localValue                        INTEGER (10),
                                        ss_Info              SS_Info          }         } ,

       registerSS_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( bearerServiceNotProvisioned

| teleserviceNotProvisioned

| illegalSS_Operation

| dataMissing

| unexpectedDataValue ) },
              callBarredErr                       [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( callBarred ),

parameter      CallBarringCause },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },


       registerSS_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition

Type Name: RejectComponent

Comments: ITU-T Q.767

Type Definition

```
SEQUENCE {
  invokedID CHOICE {
                      derivable            InvokeIDType,
                      notDerivable      NULL },
  problem    CHOICE {
                      generalProblem         [0] IMPLICIT GeneralProblem,
                      invokeProblem          [1] IMPLICIT InvokeProblem,
                      returnResultProblem [2] IMPLICIT ReturnResultProblem,
                      returnErrorProblem  [3] IMPLICIT ReturnErrorProblem } }
```

Detailed Comments: Reject Component is not specific to any particular operation. The invokeID may be used to identify a specific operation.

ASN.1 Type Definition

Type Name: RetrieveMPTY_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       retrieveMPTY_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (122)  },

       retrieveMPTY_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                         localValue                          INTEGER (122) }
} ,

       retrieveMPTY_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSS_Operation

| facilityNotSupported ) },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       retrieveMPTY_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_retrieveMPTY (5 - 30 s)

ASN.1 Type Definition

Type Name: SplitMPTY_Components

Comments: GSM 04.80, 4.2

Type Definition

```
CHOICE {
       splitMPTY_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (121)  },

       splitMPTY_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                         localValue                          INTEGER (121) }
} ,

       splitMPTY_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSS_Operation

| facilityNotSupported ) },
              ss_ErrorStatusErr              [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_ErrorStatus ),

parameter      SS_Status },
              ss_IncompatibilityErr           [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( ss_Incompatibility ),

parameter      SS_IncompatibilityCause },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       splitMPTY_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer T_splitMPTY (5 - 30 s)

ASN.1 Type Definition

Type Name: UnstructuredSSNotify_Components

Comments: GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       unstructuredSSNotify_InvokeComp           [1] IMPLICIT SEQUENCE {
                                 invokeID                     InvokeIDType,
                                 localValue                    INTEGER (61),
                                 ussd_Arg                    USSD_Arg                  },

       unstructuredSSNotify_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                  invokeID                   InvokeIDType,
                                  result                        SEQUENCE {
                                          localValue                          INTEGER (61) }  } ,

       unstructuredSSNotify_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSubscriber

| illegalEquipment

| absentSubscriber

| dataMissing

| unexpectedDataValue

| unknownAlphabet

| ussd_Busy) },
              systemFailureErr                 [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       unstructuredSSNotify_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition	
Type Name:	UnstructuredSSRequest_Components
Comments:	GSM 09.02, 14.5, 14.6.4, 14.6.6, 14.7.4, 14.7.7

Type Definition

```
CHOICE {
       unstructuredSSRequest_InvokeComp           [1] IMPLICIT SEQUENCE {
                                invokeID                      InvokeIDType,
                                localValue                     INTEGER (60),
                                ussd_Arg                     USSD_Arg                  },

       unstructuredSSRequest_ReturnResultComp  [2] IMPLICIT SEQUENCE {
                                 invokeID                    InvokeIDType,
                                 result                         SEQUENCE {
                                        localValue                          INTEGER (60),
                                        ussd_Res                          USSD_Res       }  } ,

       unstructuredSSRequest_ReturnErrorComp   CHOICE {
              errorCodes                          [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode    ErrorLocalValues ( illegalSubscriber

| illegalEquipment

| absentSubscriber

| dataMissing

| unexpectedDataValue

| unknownAlphabet

| ussd_Busy) },
              systemFailureErr                  [3] IMPLICIT SEQUENCE { invokeID
InvokeIDType,

errorCode     ErrorLocalValues ( systemFailure ),

parameter      NetworkResource}   },

       unstructuredSSRequest_RejectComp            [4] IMPLICIT RejectComponent        }
```

Detailed Comments: Timer m

ASN.1 Type Definition	
Type Name:	AddressString
Comments:	GSM 09.02, 14.7.8

Type Definition

```
OCTET STRING (SIZE ( 1 .. maxAddressLength ))
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	Asn1Integer
Comments:	INTEGER of ASN.1 type

Type Definition

```
INTEGER
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	BasicServiceCode
Comments:	GSM 09.02, 14.7.8

Type Definition

```
CHOICE {
       bearerService   [2] IMPLICIT BearerServiceCode,
       teleservice        [3] IMPLICIT TeleserviceCode       }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	BasicServiceGroupList
Comments:	GSM 09.02, 14.7.8
Type Definition	
`SEQUENCE SIZE ( 1 .. maxNumOfBasicServiceGroups ) OF BasicServiceCode`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	BearerServiceCode
Comments:	GSM 09.02, 14.7.10
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CallBarringCause
Comments:	GSM 09.02, 14.7.7
Type Definition	

`ENUMERATED {`
`  barringServiceActive  (0),`
`  operatorBarring  (1) }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CallBarringFeature
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`  basicService  BasicServiceCode  OPTIONAL,`
`  ss_Status  [4] IMPLICIT SS_Status  OPTIONAL }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CallBarringFeatureList
Comments:	GSM 09.02, 14.7.4
Type Definition	
`SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF CallBarringFeature`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CallBarringInfo
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`  ss_Code  SS_Code  OPTIONAL,`
`  callBarringFeatureList  CallBarringFeatureList  }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CliRestrictionOption
Comments:	GSM 09.02, 14.7.4
Type Definition	
ENUMERATED { permanent (0), temporaryDefaultRestricted (1), temporaryDefaultAllowed (2) }	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_Feature
Comments:	GSM 09.02, 14.7.4
Type Definition	
SEQUENCE { basicService BasicServiceCode OPTIONAL, preferentialCUG_Indicator CUG_Index OPTIONAL, interCUG_Restrictions InterCUG_Restrictions }	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_FeatureList
Comments:	GSM 09.02, 14.7.4
Type Definition	
SEQUENCE SIZE (1.. maxNumOfBasicServiceGroups) OF CUG_Feature	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_Index
Comments:	GSM 09.02, 14.7.4
Type Definition	
INTEGER (0..32767)	
Detailed Comments:	The internal structure is defined in ETS 300 138.

ASN.1 Type Definition	
Type Name:	CUG_Interlock
Comments:	GSM 09.02, 14.7.4
Type Definition	
OCTET STRING (SIZE (4))	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_Info
Comments:	GSM 09.02, 14.7.4
Type Definition	
SEQUENCE { cug_SubscriptionList CUG_SubscriptionList, cug_FeatureList CUG_FeatureList OPTIONAL }	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	CUG_Subscription
Comments:	GSM 09.02, 14.7.4

Type Definition

```
SEQUENCE {
        cug_Index                       CUG_Index,
        cug_Interlock                  CUG_Interlock,
        intraCUG_Options         IntraCUG_Options,
        basicServiceGroupList    BasicServiceGroupList    OPTIONAL }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	CUG_SubscriptionList
Comments:	GSM 09.02, 14.7.4

Type Definition

```
SEQUENCE SIZE (1.. maxNumOfCUG ) OF CUG_Subscription
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	ErrorLocalValues
Comments:	GSM 04.80, 4.5

Type Definition

```
INTEGER {
  unknownSubscriber                                 (1),
  illegalSubscriber                                      (9),
  bearerServiceNotProvisioned                    (10),
  teleserviceNotProvisioned                         (11),
  illegalEquipment                                       (12),
  callBarred                                                (13),
  illegalSS_Operation                                  (16),
  ss_ErrorStatus                                        (17),
  ss_NotAvailable                                       (18),
  ss_SubscriptionViolation                           (19),
  ss_Incompatibility                                    (20),
  facilityNotSupported                                 (21),
  absentSubscriber                                      (27),
  systemFailure                                            (34),
  dataMissing                                              (35),
  unexpectedDataValue                                (36),
  pw_RegistrationFailure                              (37),
  negativePW_Check                                   (38),
  numberOfPW_AttemptsViolation                (43),
  unknownAlphabet                                      (71),
  ussd_Busy                                                 (72),
  maxNumberOfMPTY_ParticipantsExceeded  (126),
  resourcesNotAvailable                               (127)
 }
```

Detailed Comments: Elements of INTEGER are global for the ATS.

ASN.1 Type Definition	
Type Name:	ForwardingFeature
Comments:	GSM 09.02, 14.7.4

Type Definition

```
SEQUENCE {
        basicService                    BasicServiceCode                    OPTIONAL,
        ss_Status                        [4] IMPLICIT SS_Status
OPTIONAL,
        forwardedToNumber         [5] IMPLICIT ISDN_AddressString          OPTIONAL,
        forwardedToSubaddress   [8] IMPLICIT ISDN_SubaddressString     OPTIONAL,
        forwardingOptions           [6] IMPLICIT ForwardingOptions             OPTIONAL,
        noReplyConditionTime      [7] IMPLICIT NoReplyConditionTime         OPTIONAL }
```

Detailed Comments:

ASN.1 Type Definition	
Type Name:	ForwardingFeatureList
Comments:	GSM 09.02, 14.7.4
Type Definition	
`SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF ForwardingFeature`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	ForwardingInfo
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`  ss_Code  SS_Code  OPTIONAL,`
`  forwardingFeatureList  ForwardingFeatureList }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	ForwardingOptions
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	GeneralProblem
Comments:	ITU-T Q.767
Type Definition	

`TCAP_Problems ( unrecognizedComponent`
`  | mistypedComponent`
`  | badlyStructuredComponent )`	

Detailed Comments:	Type restricted to these three.

ASN.1 Type Definition	
Type Name:	InterCUG_Restrictions
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	bits: 876543: 000000 (unused) bits 21: 00 CUG only facilities 01 CUG with outgoing access 10 CUG with incoming access 11 CUG with both outgoing and incoming access

ASN.1 Type Definition	
Type Name:	IntraCUG_Options
Comments:	GSM 09.02, 14.7.4
Type Definition	

`ENUMERATED {`
`  noCUG_Restrictions (0),`
`  cugIC_CallBarred (1),`
`  cugOG_CallBarred (2) }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	InvokeIDType
Comments:	ITU-T, Q.773
Type Definition	
`INTEGER (-128 .. 127)`	
Detailed Comments:	Values: Sending Components: If it is an invoke component then use Test Case Variable (with default) to set value. If another invoke component is sent the TCV should be incremented beforehand. If it is a return result, error or reject component in response to a received invoke component then use TCV also, making sure the value is set to the value of the received component beforehand. Receiving Components: If it is an invoke comp then use '?'. If it is a return result, error or reject component in response to a sent invoke component then use TCV value (as used in sent invoke component).

ASN.1 Type Definition	
Type Name:	InvokeProblem
Comments:	ITU-T Q.767
Type Definition	

`TCAP_Problems ( duplicateInvokeID`
`| unrecognizedOperation`
`| mistypedArgument`
`| resourceLimitation`
`| initiatingRelease`
`| unrecognizedLinkedID`
`| linkedResponseUnexpected`
`| unexpectedLinkedOperation )`	

Detailed Comments:	Type restricted to these 8.

ASN.1 Type Definition	
Type Name:	ISDN_AddressString
Comments:	GSM 09.02, 14.7.8
Type Definition	
`AddressString (SIZE ( 1 .. maxISDN_AddressLength ))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	ISDN_SubaddressString
Comments:	GSM 09.02, 14.7.8
Type Definition	
`OCTET STRING (SIZE ( 1 .. maxISDN_SubaddressLength ))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	NetworkResource
Comments:	GSM 09.02, 14.7.8,
Type Definition	

`ENUMERATED {`
`plmn (0),`
`hlr (1),`
`vlr (2),`
`pvlr (3),`
`controllingMSC (4),`
`vmsc (5),`
`eir (6),`
`rss (7) }`	

Detailed Comments:	Elements of INTEGER are global for the ATS.

ASN.1 Type Definition	
Type Name:	NoReplyConditionTime
Comments:	GSM 09.02, 14.7.4
Type Definition	
`INTEGER (5 .. 30 )`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	Operation
Comments:	ITU-T Q.767
Type Definition	

`CHOICE {`
`  localValue  INTEGER,`
`  globalValue OBJECT IDENTIFIER}`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	OverrideCategory
Comments:	GSM 09.02, 14.7.4
Type Definition	

`ENUMERATED {`
`  overrideEnabled  (0),`
`  overrideDisabled  (1) }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	Password
Comments:	GSM 09.02, 14.7.4
Type Definition	
`NumericString (FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"))  (SIZE (4))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	PW_RegistrationFailureCause
Comments:	GSM 09.02, 14.7.7
Type Definition	

`ENUMERATED {`
`  undetermined  (0),`
`  invalidFormat  (1),`
`  newPasswordsMismatch  (2) }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	ReturnErrorProblem
Comments:	ITU-T Q.767
Type Definition	

`TCAP_Problems ( unrecognizedInvokeID`
`| returnErrorUnexpected`
`| unrecognizedError`
`| unexpectedError`
`| mistypedParameter )`	

Detailed Comments:	Type restricted to these 5.

ASN.1 Type Definition	
Type Name:	ReturnResultProblem
Comments:	ITU-T Q.767
Type Definition	

`TCAP_Problems ( unrecognizedInvokeID`
`| returnResultUnexpected`
`| mistypedResult )`	

Detailed Comments:	Type restricted to these three.

ASN.1 Type Definition	
Type Name:	SS_Code
Comments:	GSM 09.02, 14.7.5
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	group (bits 8765), and specific service (bits 4321)

ASN.1 Type Definition	
Type Name:	SS_Data
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`ss_Code SS_Code OPTIONAL,`
`ss_Status [4] IMPLICIT SS_Status OPTIONAL,`
`ss_SubscriptionOption SS_SubscriptionOption OPTIONAL,`
`basicServiceGroupList BasicServiceGroupList OPTIONAL }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_ForBS_Code
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`ss_Code SS_Code,`
`basicService BasicServiceCode OPTIONAL }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_IncompatibilityCause
Comments:	GSM 09.02, 14.7.8
Type Definition	

`SEQUENCE {`
`ss_Code [1] IMPLICIT SS_Code OPTIONAL,`
`basicService BasicServiceCode OPTIONAL,`
`ss_Status [4] SS_Status OPTIONAL }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_Info
Comments:	GSM 09.02, 14.7.4
Type Definition	

`CHOICE {`
`forwardingInfo [0] IMPLICIT ForwardingInfo,`
`callBarringInfo [1] IMPLICIT CallBarringInfo,`
`cug_Info [2] IMPLICIT CUG_Info,`
`ss_Data [3] IMPLICIT SS_Data }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_Status
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	bits 8765: 0000 unused, bits 4: SS state information Q bit, bits 3: SS state information P bit, bits 2: SS state information R bit, bits 1: SS state information A bit.

ASN.1 Type Definition	
Type Name:	SS_SubscriptionOption
Comments:	GSM 09.02, 14.7.4
Type Definition	

`CHOICE {`
`  overrideCategory  [1] IMPLICIT OverrideCategory,`
`  cliRestrictionOption  [2] IMPLICIT CliRestrictionOption }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	SS_UserData
Comments:	GSM 09.02, 14.7.4
Type Definition	
`IA5String (SIZE (1 .. maxSignalInfoLength))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	TCAP_Problems
Comments:	ITU-T Q.767
Type Definition	

`INTEGER {`
`  unrecognizedComponent  (0),  -- GeneralProblem`
`  unrecognizedInvokeID  (0),  -- ReturnResultProblem, ReturnErrorProblem`
`  duplicateInvokeID  (0),  -- InvokeProblem`
`  mistypedComponent  (1),  -- GeneralProblem`
`  returnErrorUnexpected  (1),  -- ReturnErrorProblem`
`  returnResultUnexpected  (1),  -- ReturnResultProblem`
`  unrecognizedOperation  (1),  -- InvokeProblem`
`  badlyStructuredComponent  (2),  -- GeneralProblem`
`  unrecognizedError  (2),  -- ReturnErrorProblem`
`  mistypedArgument  (2),  -- InvokeProblem, originally called mistypedParameter in TCAP`
`  mistypedResult  (2),  -- ReturnResultProblem`
`  resourceLimitation  (3),  -- InvokeProblem`
`  unexpectedError  (3),  -- ReturnErrorProblem`
`  mistypedParameter  (4),  -- ReturnErrorProblem`
`  initiatingRelease  (4),  -- InvokeProblem`
`  unrecognizedLinkedID  (5),  -- InvokeProblem`
`  linkedResponseUnexpected  (6),  -- InvokeProblem`
`  unexpectedLinkedOperation  (7)  -- InvokeProblem`
`  }`	

Detailed Comments:	Errors of the same integer value are distinguished by their different parent types (General, Invoke, ReturnResult, ReturnError).

ASN.1 Type Definition	
Type Name:	TeleserviceCode
Comments:	GSM 09.02, 14.7.9
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	USSD_Arg
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`ussd_DataCodingScheme USSD_DataCodingScheme,`
`ussd_String USSD_String }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	USSD_Res
Comments:	GSM 09.02, 14.7.4
Type Definition	

`SEQUENCE {`
`ussd_DataCodingScheme USSD_DataCodingScheme,`
`ussd_String USSD_String }`	

Detailed Comments:	

ASN.1 Type Definition	
Type Name:	USSD_DataCodingScheme
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1))`	
Detailed Comments:	

ASN.1 Type Definition	
Type Name:	USSD_String
Comments:	GSM 09.02, 14.7.4
Type Definition	
`OCTET STRING (SIZE (1..maxUSSD_StringLength))`	
Detailed Comments:	The structure of the contents of the USSD-String is dependent on the USSD-DataCodingScheme as described in TS GSM03.38.

Test suite operation definitions

Test Suite Operation Definition	
Operation Name:	OC_Asn1intToOct(n:Asn1Integer; l: INTEGER)
Result Type:	OCTETSTRING
Comments:	
Description	
OC_Asn1intToOct converts the ASN.1 INTEGER `n` into OCTETSTRING with length = `l`. for example: OC_Asn1intToOct(14,1) = '0E'O; OC_Asn1intToOct(18,1) = '12'O; OC_Asn1intToOct(18,2) = '0012'O; OC_Asn1intToOct(-128,1) = '80'O (MSB (position p out of 1...p) represents -2exp(p-1)); OC_Asn1intToOct(-32768,2) = '8000'O (MSB (position p out of 1...p) represents -2exp(p-1)).	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_Bcap(setup:SETUP_MO_PDU; callproc:CALL_PROC_PDU; mem:INTEGER)
Result Type:	BCAP
Comments:	

Description

OC_Bcap operation returns a bearer capability IE according the following rule:

- for `mem` = 1:
  - if bearer capability IE were presented in `callproc` the returned BC is the bearer capability 1 of the `callproc`;
  - otherwise, the returned bearer capability is the bearer capability 1 of the `setup`.

- for `mem` = 2:
  - if bearer capability IE were presented in `callproc` the returned BC is the bearer capability 2 of the `callproc`;
  - otherwise, the returned bearer capability is the bearer capability 2 of the `setup`.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_BinAdd(bitstr1:BITSTRING; bitstr2:BITSTRING)
Result Type:	BITSTRING
Comments:	

Description

OC_BinAdd operation performs binary addition of two input parameters `bitstr1` and `bitstr2`, then returns the result of the addition. These two input parameters shall have the same length, the result of the operation has the same length as the input parameters.

for example:

OC_BinAdd('01000'B, '00110'B) = '01110'B;
OC_BinAdd('01000'B, '00011'B) = '01011'B;
OC_BinAdd('00100'B, '00010'B) = '00110'B.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_Bit7(bstring: B_8)
Result Type:	BITSTRING
Comments:	The input parameter bstring is of type BITSTRING[8].

Description

OC_Bit7(bstring) returns the value of bit 7 in the `bstring`.

for example:

OC_Bit7('01010101'B) = '1'B,
OC_Bit7('10101010'B) = '0'B

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_BCDtoInt(bcdstring:HEXSTRING; relevant_digits: INTEGER)
Result Type:	INTEGER
Comments:	

Description

The operation OC_BCDtoInt converts last 'relevant_digits' of an HEXSTRING containing BCD coded digits to an integer representation of these relevant digits.

Example: OC_BCDtoInt('12345'H, 3) := 345

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_CalledNumCHK(callednum:OCTETSTRING; dialnum:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
the operation OC_CalledNumCHK to check whether the called party number `callednum`, which is represented by OCTETSTRING, is the same as the dialed number 'dialnum`, which is represented by IA5String. It returns TRUE if they are the same, otherwise FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_ChkSRES(sres:OCTETSTRING; ki: BITSTRING; rand:BITSTRING)
Result Type:	BOOLEAN
Comments:	sres is 32 bits value.
Description	
OC_ChkSRES checks the input parameter `sres` according to the authentication algorithm defined in the following procedure. It returns TRUE if the `sres` is correct, otherwise it returns FALSE. - firstly the `ki` XOR to the `rand` results in RES1; - then compare the most significant 32 bits of the RES1 with the `sres`; - if they are equal, the `sres` is correct and the operation returns TRUE; - if they are not equal, the `sres` is wrong and the operation returns FALSE. NOTE: this procedure is the test algorithm for authentication defined by GSM.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_CnvtMax(max:INTEGER)
Result Type:	B_2
Comments:	
Description	
The allowed integer values for max are 1, 2, 4, 7 (maximum number of retransmissions). The result BITSTRING is 2 bits long. OC_CnvtMax converts the input integer `max` into a BITSTRING according to the following rule: 1. the result is '00'B if the `max` = 1, 2. the result is '01'B if the `max` = 2, 3. the result is '10'B if the `max` = 4, 4. the result is '11'B if the `max` = 7. for example: OC_CnvtMax(1) = '00'B, OC_CvntMax(7) = '11'B.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_CnvtTx(tx:INTEGER)
Result Type:	B_4
Comments:	The result BITSTRING is 4 bits long.

Description

OC_CnvtTx converts the input integer `tx` into a BITSTRING according the following rule:

1. the result is '0000'B if the `tx` = 3,
2. the result is '0001'B if the `tx` = 4,
3. the result is '0010'B if the `tx` = 5,
4. the result is '0011'B if the `tx` = 6,
5. the result is '0100'B if the `tx` = 7,
6. the result is '0101'B if the `tx` = 8,
7. the result is '0110'B if the `tx` = 9,
8. the result is '0111'B if the `tx` = 10,
9. the result is '1000'B if the `tx` = 11,
10. the result is '1001'B if the `tx` = 12,
11. the result is '1010'B if the `tx` = 14,
12. the result is '1011'B if the `tx` = 16,
13. the result is '1100'B if the `tx` = 20,
14. the result is '1101'B if the `tx` = 25,
15. the result is '1110'B if the `tx` = 32,
16. the result is '1111'B if the `tx` = 50.

for example:

OC_CnvtTx(3) = '0000'B,
OC_CvntTx(5) = '0010'B.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_CodeSMSCBMessage(firstoct: INTEGER; lastoct: INTEGER)
Result Type:	OCTETSTRING
Comments:	

Description

The operation codes a part of the contents for a cell broadcast short message. The cell broadcast short message, is 82 octets long, i.e. 93 characters, each represented by 7 bits. As many as possible different characters are sent, the characters are those corresponding to the 7-bit representation of the integers 0, 1, 2, ..., 92. The bits are arranged acc. to GSM 03.38, clause 6.1.2.1.1. The result of this operation is the octetstring of the octets 'firstoct' to 'lastoct' (16 octets for the first message block, 22 octets for the 2nd, 3rd and 4th blocks), with the octets of the cell broadcast short message being numbered from 1 to 82.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_CodingOfUssdString(text: IA5String)
Result Type:	OCTETSTRING
Comments:	

Description

This operation provides the coding for a USSD String 'text' in the default alpabet, each character represented by 7 bits. The bits are arranged acc. to GSM 03.38, clause 6.1.2.2.1.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_ComputeSMContents(NumberOfCharacters: INTEGER)
Result Type:	OCTETSTRING
Comments:	max. 160 characters, i.e. 140 octets.
Description	
This operation provides a short message's contents with a specified number of characters 'NumberOfCharacters', each represented by 7 bits. As possible different characters are sent, the characters are those corresponding to the 7-bit representation of 0, 1, 2, ... up to ('NumberOfCharacters' - 1). If more than 128 characters are sent, the rest of the characters is the corresponding to 0, 1, ... up to (NumberOfCharacters - 128 - 1), e.g. for 160 characters: 0, 1, ..., 127, 0, 1, ..., 31. The bits are arranged acc. to GSM 03.38, clause 6.1.2.1.1.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_ComputeSMContentsSpecText(NumbOfIA5chara: INTEGER; text: IA5String)
Result Type:	OCTETSTRING
Comments:	max. 160 characters, i.e. 140 octets.
Description	
This operation provides a short message's contents with a specified number of characters 'NumbOfIA5chara', each represented by 7 bits. 'text' is used as contents of the short message. If "text' contains less than 'NumberOfCharacters' characters, 'text' is repeated until the short message reaches the 'NumberOfCharacters' characters long. The bits packing is arranged according. to GSM 03.38, clause 6.1.2.1.1.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_CphKeyGen(ki:BITSTRING; rand:BITSTRING)
Result Type:	BITSTRING
Comments:	both ki and rand are 128 bits values, the result of the operation is 64 bits value
Description	
OC_CphKeyGen generates the ciphering key from the input parameters according to the following procedure: - firstly the `ki` XOR to the `rand` results in RES1; - then discard the most significant 32 bits of the RES; - the next 64 bits of RES1 are the ciphering key, the operation returns this value. - the 32 least singnificant bits are not used.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_FirstDigi(bcddigits:HEXSTRING)
Result Type:	B_4
Comments:	
Description	
The input parameter bcddigits shall be a BCD string (subset of HEXSTRING), the resut is a BITSTRING[4] of a binary representation of one BCD digit. The function of the OC_FirstDigi is to return the first (most significant) digit of the input parameter `bcddigits`. for example: OC_FirstDigi('12345') = '0001'B, OC_FirstDigi('012345678') = '0000'B.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_FnInc(fn:FN; inc:INTEGER)
Result Type:	FN
Comments:	

Description

The function of OC_FnInc operation to increase frame number with `inc`. The frame number to be incremented is the input parameter `fn` in FN type and the increment `inc` is in INTEGER type, the incremented frame number is returned in FN type.

```
OC_FnInc(fn, inc)
FN     fn;
INTEGER  inc;
{
int     fmrin;
FN    frmout;

   frmin = 51 * ((fn.t3 - fn.t2) MOD 26) + fn.t3 + 1326 * fn.t1_ ;

   frmout.t1_ = ((frmin + inc) DIV 1326) MOD 32;
   frmout.t2   = (frmin + inc) MOD 26;
   frmout.t3   = (frmin + inc) MOD 51;
   return (frmout);
}
```

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_GetSCTimeStamp(timezone:TZONES)
Result Type:	TPSCTS
Comments:	TPSCTS is HEXSTRING[14]

Description

This Operation provides the hexstring containing the service center time stamp (SCTS) according to GSM 03.40, clauses 9.2.2.1 and 9.2.3.11. The TSO reads the current time of the test systems clock and transforms the time in conbination with the input parameter 'timezone' into a service center time stamp.

Example:

1996 April 18, 15:32:46, timezone=4

OC_GetSCTimeStamp returns 69408151236440

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_IncTmsi(tmsi:TMSI_V; inc:OCTETSTRING)
Result Type:	TMSI_V
Comments:	the `tmsi` is 4 OCTETs long

Description

OC_IncTmsi operation adds the two input parameters and returns the result. An overflow of addition is allowed.

For example :

OC_IncTmsi('33542140'O + '01'O) = '33542141'O;
OC_IncTmsi('21322140'O + '08'O) = '21322148'O.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_InRang(tx: INTEGER; maxret:INTEGER; m: INTEGER)
Result Type:	BOOLEAN
Comments:	
Description	
The operation returns TRUE if (m DIV (maxret * ((230+maxret -1)/maxret))) is inside the following interval: [0.8 - ((tx+1)/2) DIV tx; 1.2 - ((tx+1)/2) DIV tx] where / is integer division, DIV is float division, m, tx and maxret are input parameters of the operation.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_IntToOct(n:INTEGER; l: INTEGER)
Result Type:	OCTETSTRING
Comments:	
Description	
OC_IntToOct converts the INTEGER `n` into OCTETSTRING with length = `l`. for example: OC_IntToOct(14,1) = '0E'O; OC_IntToOct(18,1) = '12'O; OC_IntToOct(18,2) = '0012'O.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LeastBits(bstring:BITSTRING; lg:INTEGER)
Result Type:	BITSTRING
Comments:	
Description	
OC_LeastBits operation returns the `lg` least significant bits of the original `bstring`. for example: OC_LeastBits('110011000101010'B, 3) = '010'B, OC_LeastBits('110011000101010'B, 6) = '101010'B.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LengthOf(identity_field :MI)
Result Type:	INTEGER
Comments:	
Description	
OC_LengthOf operation returns the actual length of the `identity_field`, the unit of length is in OCTET.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LengthOfBCDN(bcdn :BCDN)
Result Type:	INTEGER
Comments:	
Description	
OC_LengthOfBCDN operation returns the actual length of an IE of type BCDN 'bcdn', the unit of length is in OCTET.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LengthOfComp(comp: Component_T)
Result Type:	LENGTH
Comments:	
Description	
OC_LengthOfComp operation returns the actual length of the Component_T `comp`, the unit of length is in OCTET.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LengthOfString(strg: IA5String)
Result Type:	INTEGER
Comments:	
Description	
OC_LengthOfString operation returns the actual length (number of characters) of the string `strg`.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_LookupS(t: INTEGER; combined :BOOLEAN)
Result Type:	INTEGER
Comments:	the algorithm is derived from Table 3.1/GSM 04.08 for values of parameter S.

Description

This operation returns an INTEGER according the following algorithm:

```
OC_LookupS( t, comb)
INTEGER t;
BOOLEAN combined;
{
    switch (t)
        {  case 3:
           case 8:
           case 14:
           case 50:
              if (combined) then
                     return(41)
                else   return(55);
              break;
           case 4:
           case 9:
           case 16:
              if (combined) then
                     return(52)
                else  return(76);
              break;
           case 5:
           case 10:
           case 20:
              if (combined) then
                     return(58)
                else  return(109);
              break;
           case 6:
           case 11:
           case 25:
              if (combined) then
                     return(86)
                else  return(163);
              break;
           case 7:
           case 12:
           case 32:
              if (combined) then
                     return(115)
                else  return(217);
              break;
     }
}
```

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_MostBits(bstring:BITSTRING; lg :INTEGER)
Result Type:	BITSTRING
Comments:	
Description	
OC_MostBits operation returns the `lg` most significant bits of the original `bstring`. for example: OC_LeastBits('110011000101010'B, 3) = '110'B, OC_LeastBits('110011000101010'B, 6) = '110011'B.	
Detailed Comments:	

Test Suite Operation Definition

Operation Name:	OC_MsrReptChk(msrres: MSRR; index:INTEGER)
Result Type:	BOOLEAN
Comments:	parameter index is used for selecting different match pattern

Description

OC_MsrReptChk operation checks whether the received measurement report contains correct values:

1. when the `index` = 1:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00000'B, '001011'B}, {'01111'B, '001001'B},
   {'10101'B, '001011'B}, {'10110'B, '001101'B},
   {'11100'B, '001111'B}, {'11111'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

2. when the `index` = 2:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00000'B, '001011'B}, {'00100'B, '001001'B},
   {'00110'B, '001011'B}, {'00101'B, '001101'B},
   {'00010'B, '001111'B}, {'00111'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

3. when the `index` = 3:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00001'B, '001011'B}, {'00101'B, '001001'B},
   {'00111'B, '001011'B}, {'00110'B, '001101'B},
   {'00011'B, '001111'B}, {'01000'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

4. when the `index` = 4 :
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   }
   equals to
   set { {'00000'B, '001011'B}, {'00010'B, '001111'B},
   {'00001'B, '001111'B}, {'00011'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

5. when the `index` = 5 :
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   }
   equals to
   set { {'00000'B, '001011'B}, {'00010'B, '001111'B},
   {'00001'B, '001111'B}, {'00100'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

6. when the `index` = 6:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00000'B, '001011'B}, {'00011'B, '001001'B},
   {'00100'B, '001011'B}, {'00101'B, '001101'B},
   {'00110'B, '001111'B}, {'00111'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

7. when the `index` = 7:
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2},
   { msrres.bcchfrq_nc3, msrres.bsic_nc3}, { msrres.bcchfrq_nc4, msrres.bsic_nc4},
   { msrres.bcchfrq_nc5, msrres.bsic_nc5}, { msrres.bcchfrq_nc6, msrres.bsic_nc6}
   }
   equals to
   set { {'00000'B, '001011'B}, {'00100'B, '001001'B},
   {'00110'B, '001011'B}, {'00101'B, '001101'B},
   {'00010'B, '001111'B}, {'00111'B, '001001'B}
   }
   then the operation returns TRUE otherwise FALSE.

8. when the `index` = 8 :
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2}
   }
   equals to
   set { {'00000'B, '001011'B}, {'00010'B, '001101'B}
   }
   then the operation returns TRUE otherwise FALSE.

9. when the `index` = 9 :
   if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2}
   }
   equals to
   set { {'00110'B, '001011'B}, {'00111'B, '001101'B}
   }
   then the operation returns TRUE otherwise FALSE.

10. when the `index` = 10 :
    if set { { msrres.bcchfrq_nc1, msrres.bsic_nc1}, { msrres.bcchfrq_nc2, msrres.bsic_nc2}
    }
    equals to
    set { {'00011'B, '001011'B}, {'00101'B, '001101'B}
    }
    then the operation returns TRUE otherwise FALSE.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_OeBit(bcddigits:HEXSTRING)
Result Type:	BITSTRING
Comments:	The input parameter `bcddigits` is really BCD string (subset of HEXSTRING), the result is BITSTRING[1].

Description

The function of the OC_OeBit is as the following:

1. it returns '1'B, if the length of the `bcddigits` is odd,
2. it returns '0'B, if the length of the `bcddigits` is even.

for example:

OC_OeBit('12583') = '1'B,
OC_OeBit('87259957') ='0'B.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_OctToInt(ostr: OCTETSTRING)
Result Type:	INTEGER
Comments:	

Description

OC_OctToInt converts the OCTETSTRING `ostr` into INTEGER.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_OctToInvokeIDType(o:OCTETSTRING)
Result Type:	InvokeIDType
Comments:	

Description

OC_OctToInvokeIDType converts the OCTETSTRING `o` into InvokeIDType, with the MSB of 'o' representing the negative value -2exp(n-1), for example '80'O->'10000000'B-> -128. The rest of the bits can code positive values up to +2exp(n-2)-1, for example '4F'O->'01111111'B-> +127.

for example:

OC_OctToInvokeIDType('80'O) = -128;
OC_OctToInvokeIDType('81'O) = -128+1 = -127;
OC_OctToInvokeIDType('40'O) = +64.
OC_OctToInvokeIDType('7F'O) = +127.
OC_OctToInvokeIDType('C0'O) = -128+64 = -64.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_OtherDigi(bcddigits:HEXSTRING)
Result Type:	OCTETSTRING
Comments:	The input parameter` bcddigits` is really BCD string (subset of HEXSTRING), the result is an even BCD digits except that the next to last may either be 'F'H or a BCD digit.
Description	
The function of the OC_OtherDigi is as the following: 1. If the number of the `bcddigits` is odd, the operation removes the most significant digit, and then reverses the order of each pair of digits; 2. If the number of the `bcddigits` is even, first the operation suffixes the `bcddigits` with 'F'H, then removes the most significant digit, and then reverses the order of each pair of digits. for example: OC_OtherDigi('12345') = '3254', OC_OtherDigi('12345678') ='325476F8'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OC_PosinSet(set: Components; comp: Component)
Result Type:	INTEGER
Comments:	
Description	
The operation OC_PosinSet returns the position of component `comp` within the SET `set`. for example: if the set = { registerSSComponents, eraseSSComponents, activateSSComponents, deactivateSSComponents } OC_PosinSet(set, registerSSComponents) = 0, OC_PosinSet(set, activateSSComponents) = 2.	
Detailed Comments:	

Test Suite Operation Definition

Operation Name: OC_RachSlots(fn1:FN; fn2:FN; cmbn: BOOLEAN; mode:INTEGER)

Result Type: INTEGER

Comments:

Description

OC_RachSlots calculates the number of RACH slots between frame number `fn1` and `fn2` excluding the slots in `fn1` and `fn2`, then return the result of the calculation.

If mode=0, 'fn1' denotes the (first) frame number for sending Paging Request, while 'fn2' denotes the (first) frame number for the received Channel Request. If mode=1 'fn1' and 'fn2' denote the (first) frame number for the received two subsequent Channel Requests. Any other values for mode is not allowed.

Since the number of RACH slots depends on the RACH being combined or not with dedicated channels, the parameter 'cmbn' is needed: TRUE --> combined, FALSE--> not combined.

It is assumed that the distance between fn1 and fn2 is not more than one group of 42432 TDMA frames (modulo 42432 is used to calculate t1' of fn1 and fn2). This is equivalent to approximately 196 seconds. The 51 TDMA frames in a multiframe are numbered 0 to 50.

fn1' := fn1 mod 51

fn2' := fn2 mod 51

CASE 1: cmbn is FALSE ('not combined')

When 'not combined', all slots are RACH slots.

1.1 mode=0 (sending paging message at fn1)

The paging uses the numbered frames 6 - 9, 12 -19, 22-29, 32-39 and 42-49.

1.1.1 If fn1 is not in the paging TDMA frame mapping, it leads testing to a test system error.

IF (fn1' < 6) OR (9 < fn1' < 12) OR (19 < fn1' < 22) OR (29 < fn1' < 32) OR (39 < fn1' < 42) OR (fn1' >49),

A test system error! The calling test case shall re-run again.

OTHERWISE

1.1.2 The number of RACH slots is equal to the number of frames between fn1 + 3 and fn2. The fn1 is added by 3 because a page message occupies 4 slots. The fn1 indicates the first slot of the paging message.

IF (fn1 + 3) < fn2, OC_RachSlots := fn2 - (fn1 + 3) -1 = fn2 - fn1 - 4,

1.1.3 If fn1+3 is equal to or greater than fn2, then this is due to fn2 being in the next group of 42432 frames. In this case 42432 frames have to be added.

IF fn2 <= (fn1 + 3), OC_RachSlots := fn2 -fn1 - 3 + 42432 -1 = fn2 -fn1 + 42428

1.2 mode=1 (receiving channel request at fn1)

The number of RACH slots is equal to the number of frames between fn1 and fn2

1.2.1 IF fn1 < fn2, OC_RachSlots := fn2 - fn1 - 1,

1.2.2 IF fn2 <= fn1, OC_RachSlots := fn2 -fn1 + 42432 -1 = fn2 -fn1 + 42431

CASE 2: cmbn is TRUE ('combined')

When combined only the slots of the numbered frames 4, 5, 14 to 36, 45 and 46 in each multiframe are RACH slots, i.e.total 27 frames per multiframe.

2.1 mode=0 (sending paging message at fn1)

The paging uses the numbered frames 6 - 9 and 12 -19.

2.1.1 If fn1 is not in the paging TDMA frame mapping, it leads testing to a test system error. If fn2 is not in the RACH TDMA frame mapping, it leads testing to fail.

IF (fn2` < 4) OR (5 < fn2` < 14) OR (36 < fn2` < 45) OR (fn2' > 46), OC_RachSlots := -9999

IF (fn1' < 6) OR (9 < fn1' < 12) OR (fn1' >19), A test system error! The calling test case shall re-run again.

OTHERWISE

2.1.2 Calculation of the number 'c' multiframes between fn1 + 3 and fn2. If fn1 + 3 is equal to or greater than fn2, then this is due to fn2 being in the next group of 42432 frames. In this case 42432 frames have to be added. '/' shall be the integer division, i.e. the result is also integer. Fractions are discarded.

2.1.2.1 IF (fn1 + 3) < fn2, c := fn2 / 51 - (fn1 + 3) / 51

2.1.2.2 IF fn2 <= (fn1 + 3), c := fn2 / 51 - (fn1 + 3) / 51 + 42432 / 51 = fn2 / 51 - (fn1 + 3) / 51 + 832

2.1.3 Calculation of the number of frames 'a' to be subtracted according to the position of fn1' within the multiframe

IF (5 < fn1` < 9) , a := 2

IF (fn1` =9), a := 3
IF (11<fn1` <20), a := fn1'-8

2.2 mode=1 (receiving channel request at fn1)

2.2.1 If fn1 or fn2 are not in the RACH TDMA frame mapping, it leads testing to fail.
IF (fn2` < 4) OR (5 < fn2` < 14) OR (36 < fn2` < 45) OR (fn2' > 46)
OR (fn1' < 6) OR (9 < fn1' < 12) OR (fn1' >19), OC_RachSlots := -9999

2.2.2 Calculation of the number 'c' multiframes between fn1 and fn2
2.2.2.1 IF fn1 < fn2, c := fn2 / 51 - fn1 / 51
2.2.2.2 IF fn2 <= fn1 c := fn2 / 51 - fn1 / 51 + 42432 / 51 = fn2 / 51 - fn1/ 51 + 832

2.2.3 Calculation of the number of frames 'a' to be subtracted according to the position of fn1' within the multiframe
IF (3 < fn1' < 6:), a:= fn1' - 3
IF (13 < fn1' < 37), a:= fn1' - 11
IF (44 < fn1' < 47), a:= fn1' - 19

2.3 Calculation of the number of slots 'b' to be added according to the position of fn2' within the multiframe
IF (3 < fn2' < 6:), b:= fn2' - 4
IF (13 < fn2' < 37), b:= fn2' - 12
IF (44 < fn2' < 47), b:= fn2' - 20

2.4 Calculation of the number of RACH slots. There are 27 RACH slots in each multiframe.
OC_RachSlots := 27 * c + b - a

Detailed Comments: TC_26_2_1_1 uses mode=0. TC_26_2_1_2 uses mode=1.

Test Suite Operation Definition

Operation Name: OC_Random(n1:INTEGER; n2:INTEGER)
Result Type: INTEGER
Comments:

Description

This operation randomly returns one number from the following candidates:
'n1', 'n1'+1, ..., 'n2'

Detailed Comments:

Test Suite Operation Definition

Operation Name: OC_RcsdPresent(msg:MODIFY_PDU)
Result Type: BOOLEAN
Comments: To check if RCSD IE is present or not in Modify PDU

Description

IF RCSD IE is present in the PDU passed on input paremeter return TRUE ELSE return FALSE.

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OC_SaveAndProc1(val:INTEGER; mode :INTEGER; cnt:INTEGER; cmbnd:BOOLEAN)
Result Type:	BOOLEAN
Comments:	This operation is used for storing and analysing the CHANNEL REQUEST message distribution.

Description

The function of this operation is defined as an pseudo C code:

```
#define SAVE 0;
#define PROC 1;

OC_SaveAndProc(val, mode,cnt)
INTEGER       val, mode,cnt;
BOOLEAN     combined;
{
  static INTEGER   buf[200];
   int i, j, n;
   if (mode == SAVE)
       { cnt = cnt mod 200;
         if (cmbnd == C_NotCombined) AND (val<(151 + 8)) then
                 {  buf[cnt] = val;
                   return (TRUE);
                  }
         if (cmbnd == C_Combined) AND (val<(81 + 8)) then
                 {  buf[cnt] = val;
                   return (TRUE);
                  }
         else return (FALSE);
        }
   if (mode == PROC)
        {
          for (i=0,  i<200, i++)
                {  n=0;
                   for (j=0, j<200, j++)
                       if (buf[j] == buf[i])
                           n = n+1;
                   if (n > 41) return (FALSE);
                }
           return(TRUE);
         }
}
```

Detailed Comments:

This test suite operation has two operation modes :

- when the mode is SAVE and the val is less than 89 for combined or 159 for non-combined, it saves the value val into internal buffer and returns TRUE, otherwise returns FALSE. 200 values will be stored in the internal buffer when the operation is invoked 200 times.
- When the mode is PROC, it analyses the values stored in the internal buffer, if no more than 41 of them are equal the operation returns TRUE, otherwise returns FALSE. (i.e. for all n, CARD{ k|f(k) = n } <= 41)

Test Suite Operation Definition

Operation Name:	OC_SaveAndProc3(val:BITSTRING; mode :INTEGER; cnt:INTEGER)
Result Type:	BOOLEAN
Comments:	This operation is used for storing and analysing the random reference.

Description

The function of this operation is defined as an pseudo C code:

```
#define SAVE 0;
#define PROC 1;

OC_SaveAndProc(val, mode,cnt)
BITSTRING[8] val;
INTEGER        mode,cnt;
{
  static BITSTRING[8]   buf[7];
   int i, j, n=7;
   if (mode == SAVE)
       { cnt = cnt mod 7;
         buf[cnt] = val;
         return(TRUE);
        }
   if (mode == PROC)
        {
          for (i=0,  i<6, i++)
             { for (j=i+1, j<7, j++)
                  if (buf[j] == buf[i])
                    { n = n-1;
                      break;
                    }
             }
           if (n>=4) then
                   return(TRUE)
           else    return (FALSE);
         }

}
```

Detailed Comments:	The function of this test suite operation is : - it saves the value val into internal buffer when the mode = SAVE. 7 values will be stored in the internal buffer when the operation is invoked 7 times. - it compares the values stored in the internal buffer when the mode = PROC, if 4 or more than 4 of them are different the operation returns TRUE, otherwise returns FALSE.

Test Suite Operation Definition

Operation Name:	OC_SaveAndRetrv(val:RQR; mode, idx, ex: INTEGER)
Result Type:	RQR
Comments:	This operation is used for storing and retrieving the random reference.

Description

The function of this test suite operation is (in prose):

- it saves the value val into internal buffer buf[idx] when the mode = SAVE and ex = 0. Maximum 9 values can be stored. The idx is ranged between 1-9.
- it returns the value stored in the internal buffer buf[idx] when the mode = RETRV and ex = 0.
- it returns a value which is different from any value stored in the buf and also different from any values which have been returned by the consecutive invocation of this operation with ex <> 0, if ex <> 0. The idx is ranged between 10-13.

The function of this operation is defined as an pseudo C code:

```
#define SAVE 0;
#define RETRV 1;

OC_SaveAndRetrv(val, mode, idx, ex)
RQR             val;
INTEGER         mode, idx, ex;
{
  int i, j;
  static RQR   buf[14];
   if ((mode == SAVE) AND (ex == 0))
       {
        buf[idx] = val;
        buf[10].ra = '00000000'B;
        buf[10].fn = '00'O;
        buf[11].ra = '00000000'B;
        buf[11].fn = '00'O;
        buf[12].ra = '00000000'B;
        buf[12].fn = '00'O;
        buf[13].ra = '00000000'B;
        buf[13].fn = '00'O;
        return (buf[10];
       }
   if ((mode == RETRV) AND (ex == 0))
        return (buf[idx]);
   if (ex != 0)
        {
          buf[0].ra = buf[1].ra + '00000001'B;
          buf[0].fn = buf[1].fn;
          for (j=1, j <13, j++)
             {
               for (i=1, i<14, i++)
                  if( buf[0] == buf[i])
                      { buf[0].ra = buf[i].ra + '00000001'B;
                        break;
                      }
               if (i == 14)
                {
                   buf[idx].ra = buf[0].ra:
                   return (buf[0]);
                }
             }
         buf[idx].ra = buf[i].ra+'00000001'B;
         return (buf[idx]);
        }
}
```

Detailed Comments:	0. The pseudo C code gives one of the possible implementations for the OC description in prose. 1. '00'O should be understood as equivalence of Fn_01, a TTCN structured type constraint. 2. The OC can save maximum 13 RQR values. In SAVE mode, idx is ranged between 1-9.

	3. For ex<>0, there are maximum 13 scans. In each scan a new value is assigned to buf[0].ra. If there is a 'non-match' in one of the 13 scans the buf[0] is returned. In case of 'all-match' for the 13 scans, the pair of (buf[i].ra + '00000001' at the last scan, buf[idx].fn is returned (idx is ranged between 10 - 13).

Test Suite Operation Definition	
Operation Name:	OC_StartTime(frmn:FN; t, i:INTEGER)
Result Type:	STRT
Comments:	

Description

OC_StartTime operation generates the STARTING TIME IE according to the input parameters.

```
(* frmn is the current frame number,
   t is the "delay" in applying the new frequencies,
   i is the contents of the starting time IE. *)
OC_StartTime(frmn, t, i)
INTEGER  t, i;
FN          frmn;
{
int tmp;
STRT strt;

tmp = 51 * ((frmn.t3 - frmn.t2) MOD 26) + frmn.t3 + 1326 * frmn.t1_ ;
tmp = (tmp + t) MOD 42432;
strt.fn.t1_ = (tmp DIV 1326) MOD 32;
strt.fn.t2   = tmp MOD 26;
strt.fn.t3   = tmp MOD 51;
strt.iei = OMIT;
if (i == 1) strt.iei := '01111100'B  /* if i=1 the information element identifier shall be included \
```

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OC_SubOctet(src:OCTETSTRING; len :INTEGER)
Result Type:	OCTETSTRING
Comments:	

Description

OC_SubOctet(src, len) is the octetstring of length `len` starting from the leftmost position of the source octetstring `src`.

For example : OC_SubOctet('123456789ABCDEF'O, 4) = '12345678'O

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OC_SubchOfFacchh(subch: B_1; cell:CellID; inst:INTEGER)
Result Type:	LOGICCH
Comments:	subch is of type BITSTRING[1]

Description

OC_SubchOfFacchh operation returns a logic channel identifier for FACCHH subchannel indicated by the input parameters `subch`, `cell` and `inst`, where 'subch' is the subchannel number, `cell` is the cell identifier which the channel belongs to, `inst` is the instance of the channel.

for example:

OC_SubchOfFacchh('0'B, C_CellA, 1) = C_FACCHH0_A_1;
OC_SubchOfFacchh('1'B, C_CellA, 1) = C_FACCHH1_A_1;
OC_SubchOfFacchh('0'B, C_CellB, 1) = C_FACCHH0_B_1;
OC_SubchOfFacchh('1'B, C_CellB, 1) = C_FACCHH1_B_1.

pseudo C code definition for the operation as following :

```
OC_SubchOfFacchh(subch, cell, inst)
BITSTRING   subch;
CellID     cell;
INTEGER   inst;
{
  LOGICCH    logch;
  if (subch=='0'B && cell == C_CellA)
     switch (inst)
       {
        case 1 : logch = C_FACCHH0_A_1; break;
        case 2 : logch = C_FACCHH0_A_2; break;
        case 3 : logch = C_FACCHH0_A_3; break;
       }
  if (subch=='0'B && cell == C_CellB)
     switch (inst)
       {
        case 1 : logch = C_FACCHH0_B_1; break;
        case 2 : logch = C_FACCHH0_B_2; break;
        case 3 : logch = C_FACCHH0_B_3; break;
       }
  if (subch=='0'B && cell == C_CellC)
     switch (inst)
       {
        case 1 : logch = C_FACCHH0_C_1; break;
        case 2 : logch = C_FACCHH0_C_2; break;
        case 3 : logch = C_FACCHH0_C_3; break;
       }
   if (subch=='1'B && cell == C_CellA)
     switch (inst)
       {
        case 1 : logch = C_FACCHH1_A_1; break;
        case 2 : logch = C_FACCHH1_A_2; break;
        case 3 : logch = C_FACCHH1_A_3; break;
       }
  if (subch=='1'B && cell == C_CellB)
     switch (inst)
       {
        case 1 : logch = C_FACCHH1_B_1; break;
        case 2 : logch = C_FACCHH1_B_2; break;
        case 3 : logch = C_FACCHH1_B_3; break;
       }
  if (subch=='1'B && cell == C_CellC)
     switch (inst)
       {
        case 1 : logch = C_FACCHH1_C_1; break;
        case 2 : logch = C_FACCHH1_C_2; break;
```

```
            case 3 : logch = C_FACCHH1_C_3; break;
          }
  return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name: OC_SubchOfSacchh(subch: B_1; cell:CellID; inst:INTEGER)
Result Type: LOGICCH
Comments: subch is of type BITSTRING[1]

Description

OC_SubchOfSacchh operation returns a logic channel identifier for SACCHH subchannel indicated by the input parameters `subch`, `cell` and `inst`, where 'subch' is the subchannel number, `cell` is the cell identifier which the channel belongs to, `inst` is the instance of the channel.

for example:

OC_SubchOfSacchh('0'B, C_CellA, 1) = C_SACCHH0_A_1;
OC_SubchOfSacchh('1'B, C_CellA, 1) = C_SACCHH1_A_1;
OC_SubchOfSacchh('0'B, C_CellB, 1) = C_SACCHH0_B_1;
OC_SubchOfSacchh('1'B, C_CellB, 1) = C_SACCHH1_B_1.

pseudo C code definition for the operation as following :

```
OC_SubchOfSacchh(subch, cell, inst)
BITSTRING   subch;
CellID     cell;
INTEGER   inst;
{
  LOGICCH    logch;
  if (subch=='0'B && cell == C_CellA)
     switch (inst)
       {
        case 1 : logch = C_SACCHH0_A_1; break;
        case 2 : logch = C_SACCHH0_A_2; break;
        case 3 : logch = C_SACCHH0_A_3; break;
       }
  if (subch=='0'B && cell == C_CellB)
     switch (inst)
       {
        case 1 : logch = C_SACCHH0_B_1; break;
        case 2 : logch = C_SACCHH0_B_2; break;
        case 3 : logch = C_SACCHH0_B_3; break;
       }
  if (subch=='0'B && cell == C_CellC)
     switch (inst)
       {
        case 1 : logch = C_SACCHH0_C_1; break;
        case 2 : logch = C_SACCHH0_C_2; break;
        case 3 : logch = C_SACCHH0_C_3; break;
       }
   if (subch=='1'B && cell == C_CellA)
     switch (inst)
       {
        case 1 : logch = C_SACCHH1_A_1; break;
        case 2 : logch = C_SACCHH1_A_2; break;
        case 3 : logch = C_SACCHH1_A_3; break;
       }
  if (subch=='1'B && cell == C_CellB)
     switch (inst)
       {
        case 1 : logch = C_SACCHH1_B_1; break;
        case 2 : logch = C_SACCHH1_B_2; break;
        case 3 : logch = C_SACCHH1_B_3; break;
       }
  if (subch=='1'B && cell == C_CellC)
     switch (inst)
       {
        case 1 : logch = C_SACCHH1_C_1; break;
        case 2 : logch = C_SACCHH1_C_2; break;
```

```
          case 3 : logch = C_SACCHH1_C_3; break;
        }
  return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name: OC_SubchOfSacch4(subch: B_2; cell:CellID)
Result Type: LOGICCH
Comments:

Description

OC_SubchOfSacch4 operation returns a logic channel identifier for SACCHC4 subchannel indicated by the input parameters `subch ` and `cell`, where `subch` is the TDMA offset, `cell` is the cell identifier which the channel belongs to.

for example:

OC_SubchOfSacch4('00'B, C_CellA) = C_SACCHC40_A;
OC_SubchOfSacch4('01'B, C_CellA) = C_SACCHC41_A;
OC_SubchOfSacch4('10'B, C_CellA) = C_SACCHC42_A;
OC_SubchOfSacch4('11'B, C_CellA) = C_SACCHC43_A;
OC_SubchOfSacch4('00'B, C_CellB) = C_SACCHC40_B;
OC_SubchOfSacch4('01'B, C_CellB) = C_SACCHC41_B.

pseudo C code definition for the operation as following :

```
OC_SubchOfSacch4(subch, cell)
BITSTRING   subch;
CellID     cell;
{
   LOGICCH    logch;
   if (cell == C_CellA)
      switch (subch)
         {
           case '00'B : logch = C_SACCHC40_A; break;
           case '01'B : logch = C_SACCHC41_A; break;
           case '10'B : logch = C_SACCHC42_A; break;
           case '11'B : logch = C_SACCHC43_A; break;
         }
   if (cell == C_CellB)
      switch (subch)
         {
           case '00'B : logch = C_SACCHC40_B; break;
           case '01'B : logch = C_SACCHC41_B; break;
           case '10'B : logch = C_SACCHC42_B; break;
           case '11'B : logch = C_SACCHC43_B; break;
         }
   if (cell == C_CellC)
      switch (subch)
         {
           case '00'B : logch = C_SACCHC40_C; break;
           case '01'B : logch = C_SACCHC41_C; break;
           case '10'B : logch = C_SACCHC42_C; break;
           case '11'B : logch = C_SACCHC43_C; break;
         }
  return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OC_SubchOfSacch8(subch: B_3; cell:CellID; inst:INTEGER)
Result Type:	LOGICCH
Comments:	offset is of type BITSTRING[3]

Description

OC_SubchOfSacch8 operation returns a logic channel identifier for SACCHC8 subchannel indicated by the input parameters `subch`, `cell` and `inst`, where `subch` is the TDMA offset, `cell` is the cell identifier which the channel belongs to, `inst` is the instance of the channel.

for example:

OC_SubchOfSacch8('000'B, C_CellA, 1) = C_SACCHC80_A_1;
OC_SubchOfSacch8('001'B, C_CellA, 1) = C_SACCHC81_A_1;
OC_SubchOfSacch8('010'B, C_CellA, 1) = C_SACCHC82_A_1;
OC_SubchOfSacch8('011'B, C_CellA, 1) = C_SACCHC83_A_1;
OC_SubchOfSacch8('100'B, C_CellA, 1) = C_SACCHC84_A_1;
OC_SubchOfSacch8('101'B, C_CellA, 1) = C_SACCHC85_A_1;
OC_SubchOfSacch8('110'B, C_CellA, 1) = C_SACCHC86_A_1;
OC_SubchOfSacch8('111'B, C_CellA, 1) = C_SACCHC87_A_1;
OC_SubchOfSacch8('000'B, C_CellB, 1) = C_SACCHC80_B_1;
OC_SubchOfSacch8('001'B, C_CellB, 1) = C_SACCHC81_B_1;
OC_SubchOfSacch8('000'B, C_CellC, 2) = C_SACCHC80_C_2;
OC_SubchOfSacch8('001'B, C_CellC, 2) = C_SACCHC81_C_2.

pseudo C code definition for the operation as following :

```
OC_SubchOfSacch8(subch, cell, inst)
BITSTRING   subch;
CellID     cell;
INTEGER   inst;
{
  LOGICCH    logch;
  if (inst==1 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SACCHC80_A_1; break;
         case '001'B : logch = C_SACCHC81_A_1; break;
         case '010'B : logch = C_SACCHC82_A_1; break;
         case '011'B : logch = C_SACCHC83_A_1; break;
         case '100'B : logch = C_SACCHC84_A_1; break;
         case '101'B : logch = C_SACCHC85_A_1; break;
         case '110'B : logch = C_SACCHC86_A_1; break;
         case '111'B : logch = C_SACCHC87_A_1; break;
       }
  if (inst==2 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SACCHC80_A_2; break;
         case '001'B : logch = C_SACCHC81_A_2; break;
         case '010'B : logch = C_SACCHC82_A_2; break;
         case '011'B : logch = C_SACCHC83_A_2; break;
         case '100'B : logch = C_SACCHC84_A_2; break;
         case '101'B : logch = C_SACCHC85_A_2; break;
         case '110'B : logch = C_SACCHC86_A_2; break;
         case '111'B : logch = C_SACCHC87_A_2; break;
       }
  if (inst==3 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SACCHC80_A_3; break;
         case '001'B : logch = C_SACCHC81_A_3; break;
         case '010'B : logch = C_SACCHC82_A_3; break;
         case '011'B : logch = C_SACCHC83_A_3; break;
         case '100'B : logch = C_SACCHC84_A_3; break;
```

```
        case '101'B : logch = C_SACCHC85_A_3; break;
        case '110'B : logch = C_SACCHC86_A_3; break;
        case '111'B : logch = C_SACCHC87_A_3; break;
       }
  if (inst==1 && cell == C_CellB)
     switch (subch)
       {
        case '000'B : logch = C_SACCHC80_B_1; break;
        case '001'B : logch = C_SACCHC81_B_1; break;
        case '010'B : logch = C_SACCHC82_B_1; break;
        case '011'B : logch = C_SACCHC83_B_1; break;
        case '100'B : logch = C_SACCHC84_B_1; break;
        case '101'B : logch = C_SACCHC85_B_1; break;
        case '110'B : logch = C_SACCHC86_B_1; break;
        case '111'B : logch = C_SACCHC87_B_1; break;
       }
  if (inst==2 && cell == C_CellB)
     switch (subch)
       {
        case '000'B : logch = C_SACCHC80_B_2; break;
        case '001'B : logch = C_SACCHC81_B_2; break;
        case '010'B : logch = C_SACCHC82_B_2; break;
        case '011'B : logch = C_SACCHC83_B_2; break;
        case '100'B : logch = C_SACCHC84_B_2; break;
        case '101'B : logch = C_SACCHC85_B_2; break;
        case '110'B : logch = C_SACCHC86_B_2; break;
        case '111'B : logch = C_SACCHC87_B_2; break;
       }
  if (inst==3 && cell == C_CellB)
     switch (subch)
       {
        case '000'B : logch = C_SACCHC80_B_3; break;
        case '001'B : logch = C_SACCHC81_B_3; break;
        case '010'B : logch = C_SACCHC82_B_3; break;
        case '011'B : logch = C_SACCHC83_B_3; break;
        case '100'B : logch = C_SACCHC84_B_3; break;
        case '101'B : logch = C_SACCHC85_B_3; break;
        case '110'B : logch = C_SACCHC86_B_3; break;
        case '111'B : logch = C_SACCHC87_B_3; break;
       }
  if (inst==1 && cell == C_CellC)
     switch (subch)
       {
        case '000'B : logch = C_SACCHC80_C_1; break;
        case '001'B : logch = C_SACCHC81_C_1; break;
        case '010'B : logch = C_SACCHC82_C_1; break;
        case '011'B : logch = C_SACCHC83_C_1; break;
        case '100'B : logch = C_SACCHC84_C_1; break;
        case '101'B : logch = C_SACCHC85_C_1; break;
        case '110'B : logch = C_SACCHC86_C_1; break;
        case '111'B : logch = C_SACCHC87_C_1; break;
       }
  if (inst==2 && cell == C_CellC)
     switch (subch)
       {
        case '000'B : logch = C_SACCHC80_C_2; break;
        case '001'B : logch = C_SACCHC81_C_2; break;
        case '010'B : logch = C_SACCHC82_C_2; break;
        case '011'B : logch = C_SACCHC83_C_2; break;
        case '100'B : logch = C_SACCHC84_C_2; break;
        case '101'B : logch = C_SACCHC85_C_2; break;
        case '110'B : logch = C_SACCHC86_C_2; break;
        case '111'B : logch = C_SACCHC87_C_2; break;
       }
  if (inst==3 && cell == C_CellC)
```

```
      switch (subch)
        {
          case '000'B : logch = C_SACCHC80_C_3; break;
          case '001'B : logch = C_SACCHC81_C_3; break;
          case '010'B : logch = C_SACCHC82_C_3; break;
          case '011'B : logch = C_SACCHC83_C_3; break;
          case '100'B : logch = C_SACCHC84_C_3; break;
          case '101'B : logch = C_SACCHC85_C_3; break;
          case '110'B : logch = C_SACCHC86_C_3; break;
          case '111'B : logch = C_SACCHC87_C_3; break;
        }
return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name: OC_SubchOfSdcch4(subch: B_2; cell:CellID)
Result Type: LOGICCH
Comments:

Description

OC_SubchOfSdcch4 operation returns a logic channel identifier for SDCCH4 subchannel indicated by the input parameters `subch ` and `cell`, where `subch` is the TDMA offset, `cell` is the cell identifier which the channel belongs to.

for example:

OC_SubchOfSdcch4('00'B, C_CellA) = C_SDCCH40_A;
OC_SubchOfSdcch4('01'B, C_CellA) = C_SDCCH41_A;
OC_SubchOfSdcch4('10'B, C_CellA) = C_SDCCH42_A;
OC_SubchOfSdcch4('11'B, C_CellA) = C_SDCCH43_A;
OC_SubchOfSdcch4('00'B, C_CellB) = C_SDCCH40_B;
OC_SubchOfSdcch4('01'B, C_CellB) = C_SDCCH41_B.

pseudo C code definition for the operation as following :

```
OC_SubchOfSdcch4(subch, cell)
BITSTRING   subch;
CellID     cell;
{
   LOGICCH    logch;
   if (cell == C_CellA)
      switch (subch)
         {
           case '00'B : logch = C_SDCCH40_A; break;
           case '01'B : logch = C_SDCCH41_A; break;
           case '10'B : logch = C_SDCCH42_A; break;
           case '11'B : logch = C_SDCCH43_A; break;
         }
   if (cell == C_CellB)
      switch (subch)
         {
           case '00'B : logch = C_SDCCH40_B; break;
           case '01'B : logch = C_SDCCH41_B; break;
           case '10'B : logch = C_SDCCH42_B; break;
           case '11'B : logch = C_SDCCH43_B; break;
         }
   if (cell == C_CellC)
      switch (subch)
         {
           case '00'B : logch = C_SDCCH40_C; break;
           case '01'B : logch = C_SDCCH41_C; break;
           case '10'B : logch = C_SDCCH42_C; break;
           case '11'B : logch = C_SDCCH43_C; break;
         }
  return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OC_SubchOfSdcch8(subch: B_3; cell:CellID; inst:INTEGER)
Result Type:	LOGICCH
Comments:	offset is of type BITSTRING[3]

Description

OC_SubchOfSdcch8 operation returns a logic channel identifier for SDCCH8 subchannel indicated by the input parameters `subch`, `cell` and `inst`, where `subch` is the TDMA offset, `cell` is the cell identifier which the channel belongs to, `inst` is the instance of the channel.

for example:

OC_SubchOfSdcch8('000'B, C_CellA, 1) = C_SDCCH80_A_1;
OC_SubchOfSdcch8('001'B, C_CellA, 1) = C_SDCCH81_A_1;
OC_SubchOfSdcch8('010'B, C_CellA, 1) = C_SDCCH82_A_1;
OC_SubchOfSdcch8('011'B, C_CellA, 1) = C_SDCCH83_A_1;
OC_SubchOfSdcch8('100'B, C_CellA, 1) = C_SDCCH84_A_1;
OC_SubchOfSdcch8('101'B, C_CellA, 1) = C_SDCCH85_A_1;
OC_SubchOfSdcch8('110'B, C_CellA, 1) = C_SDCCH86_A_1;
OC_SubchOfSdcch8('111'B, C_CellA, 1) = C_SDCCH87_A_1;
OC_SubchOfSdcch8('000'B, C_CellB, 1) = C_SDCCH80_B_1;
OC_SubchOfSdcch8('001'B, C_CellB, 1) = C_SDCCH81_B_1;
OC_SubchOfSdcch8('000'B, C_CellC, 2) = C_SDCCH80_C_2;
OC_SubchOfSdcch8('001'B, C_CellC, 2) = C_SDCCH81_C_2.

pseudo C code definition for the operation as following :

```
OC_SubchOfSdcch8(subch, cell, inst)
BITSTRING   subch;
CellID     cell;
INTEGER   inst;
{
  LOGICCH    logch;
  if (inst==1 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SDCCH80_A_1; break;
         case '001'B : logch = C_SDCCH81_A_1; break;
         case '010'B : logch = C_SDCCH82_A_1; break;
         case '011'B : logch = C_SDCCH83_A_1; break;
         case '100'B : logch = C_SDCCH84_A_1; break;
         case '101'B : logch = C_SDCCH85_A_1; break;
         case '110'B : logch = C_SDCCH86_A_1; break;
         case '111'B : logch = C_SDCCH87_A_1; break;
       }
  if (inst==2 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SDCCH80_A_2; break;
         case '001'B : logch = C_SDCCH81_A_2; break;
         case '010'B : logch = C_SDCCH82_A_2; break;
         case '011'B : logch = C_SDCCH83_A_2; break;
         case '100'B : logch = C_SDCCH84_A_2; break;
         case '101'B : logch = C_SDCCH85_A_2; break;
         case '110'B : logch = C_SDCCH86_A_2; break;
         case '111'B : logch = C_SDCCH87_A_2; break;
       }
  if (inst==3 && cell == C_CellA)
     switch (subch)
       {
         case '000'B : logch = C_SDCCH80_A_3; break;
         case '001'B : logch = C_SDCCH81_A_3; break;
         case '010'B : logch = C_SDCCH82_A_3; break;
         case '011'B : logch = C_SDCCH83_A_3; break;
         case '100'B : logch = C_SDCCH84_A_3; break;
```

```
        case '101'B : logch = C_SDCCH85_A_3; break;
        case '110'B : logch = C_SDCCH86_A_3; break;
        case '111'B : logch = C_SDCCH87_A_3; break;
       }
 if (inst==1 && cell == C_CellB)
    switch (subch)
      {
       case '000'B : logch = C_SDCCH80_B_1; break;
       case '001'B : logch = C_SDCCH81_B_1; break;
       case '010'B : logch = C_SDCCH82_B_1; break;
       case '011'B : logch = C_SDCCH83_B_1; break;
       case '100'B : logch = C_SDCCH84_B_1; break;
       case '101'B : logch = C_SDCCH85_B_1; break;
       case '110'B : logch = C_SDCCH86_B_1; break;
       case '111'B : logch = C_SDCCH87_B_1; break;
      }
 if (inst==2 && cell == C_CellB)
    switch (subch)
      {
       case '000'B : logch = C_SDCCH80_B_2; break;
       case '001'B : logch = C_SDCCH81_B_2; break;
       case '010'B : logch = C_SDCCH82_B_2; break;
       case '011'B : logch = C_SDCCH83_B_2; break;
       case '100'B : logch = C_SDCCH84_B_2; break;
       case '101'B : logch = C_SDCCH85_B_2; break;
       case '110'B : logch = C_SDCCH86_B_2; break;
       case '111'B : logch = C_SDCCH87_B_2; break;
      }
 if (inst==3 && cell == C_CellB)
    switch (subch)
      {
       case '000'B : logch = C_SDCCH80_B_3; break;
       case '001'B : logch = C_SDCCH81_B_3; break;
       case '010'B : logch = C_SDCCH82_B_3; break;
       case '011'B : logch = C_SDCCH83_B_3; break;
       case '100'B : logch = C_SDCCH84_B_3; break;
       case '101'B : logch = C_SDCCH85_B_3; break;
       case '110'B : logch = C_SDCCH86_B_3; break;
       case '111'B : logch = C_SDCCH87_B_3; break;
      }
 if (inst==1 && cell == C_CellC)
    switch (subch)
      {
       case '000'B : logch = C_SDCCH80_C_1; break;
       case '001'B : logch = C_SDCCH81_C_1; break;
       case '010'B : logch = C_SDCCH82_C_1; break;
       case '011'B : logch = C_SDCCH83_C_1; break;
       case '100'B : logch = C_SDCCH84_C_1; break;
       case '101'B : logch = C_SDCCH85_C_1; break;
       case '110'B : logch = C_SDCCH86_C_1; break;
       case '111'B : logch = C_SDCCH87_C_1; break;
      }
 if (inst==2 && cell == C_CellC)
    switch (subch)
      {
       case '000'B : logch = C_SDCCH80_C_2; break;
       case '001'B : logch = C_SDCCH81_C_2; break;
       case '010'B : logch = C_SDCCH82_C_2; break;
       case '011'B : logch = C_SDCCH83_C_2; break;
       case '100'B : logch = C_SDCCH84_C_2; break;
       case '101'B : logch = C_SDCCH85_C_2; break;
       case '110'B : logch = C_SDCCH86_C_2; break;
       case '111'B : logch = C_SDCCH87_C_2; break;
      }
 if (inst==3 && cell == C_CellC)
```

```
    switch (subch)
      {
        case '000'B : logch = C_SDCCH80_C_3; break;
        case '001'B : logch = C_SDCCH81_C_3; break;
        case '010'B : logch = C_SDCCH82_C_3; break;
        case '011'B : logch = C_SDCCH83_C_3; break;
        case '100'B : logch = C_SDCCH84_C_3; break;
        case '101'B : logch = C_SDCCH85_C_3; break;
        case '110'B : logch = C_SDCCH86_C_3; break;
        case '111'B : logch = C_SDCCH87_C_3; break;
      }
return(logch);
}
```

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OM_2Msgs(ch:LOGICCH; pgg:PGG; mode:SENDINGMODE)
Result Type:	BOOLEAN
Comments:	

Description

OM_2Msgs puts the layer 2 emulator into a special operation mode then returns. In this special mode the layer 2 emulator sends the next two consecutive messages in the following way:

- send the first message on the paging subchannel indicated by the `pgg` and 'ch';

- if the `mode` = C_NxtButOne, send the second message in the next but one paging sub block;
- if the `mode` = C_FmrAGB, send the second message in a former access grant block;
- if the `mode` = C_BfReOcc, send the second message before the MS's original paging subchannel re-occurs but later than the next paging block of that CCCH (paging block not belong to the MS);
- if the `mode` = C_NxtButOneNxt, nothing is sent in the next but one paging sub block, then send the second message in the next paging subblock of the MS's paging subchannel.

Detailed Comments:

Test Suite Operation Definition

Operation Name: OM_Assoc(lgch1:LOGICCH; sublgch1:LOGICCH; sublgch2:LOGICCH; sublgch3:LOGICCH; sublgch4:LOGICCH; sublgch5:LOGICCH; sublgch6:LOGICCH; sublgch7: LOGICCH; sublgch8:LOGICCH; lgch2:LOGICCH; sublgch9:LOGICCH; sublgch10:LOGICCH; sublgch11:LOGICCH; sublgch12:LOGICCH; sublgch13:LOGICCH; sublgch14:LOGICCH; sublgch15: LOGICCH; sublgch16:LOGICCH)

Result Type: BOOLEAN

Comments:

Description

OM_Assoc operation associates the sub logic channel identifiers `sublgch1` to `sublgch8` and `sublgch9` to `sublgch16` with the generic "parent" channel identifier ` lgch1` and `lgch2` respectively, therefore the subchannel identifiers can refer to the corresponding channels configured by OM_ChConf operation.

The `lgch1` can be an identifier for SDCCH4, SDCCH8, FACCHH and the `lgch2` can be SACCHC4, SACCHC8, and SACCHH.

"dummy" in the actual parameter list means the corresponding parameter is not used.

Naming convention for logical channel constants:

C_<channel type>[<rate>][<subchannel>][_<cell>][_<instance>]

e.g.: C_FACCH H 0 _C _3 => C_FACCHH0_C_3

channel type: FCCH, SCH, BCCH, CBCH, PCH, RACH, FACCH, SACCH, SDCCH, DCCH

rate: F(ull), H(alf), 4 (fourths, C4 for SACCHC4), 8 (eighths, C8 for SACCHC8).

subchannel: H: 0, 1
(C)4: 0, ..., 3
(C)8: 0, ..., 7

cell: A, ..., H

instance: 1, ..., 3 When several channels of the same type can exist in one cell simultaneously (in different time slots and / or on different frequencies) they are identified by their instance. The parameters of ChConf are used to specify the frequency and slot of this instance of the channel type.

Not all possible combinations of this naming syntax have been used. According to the elements that are used / omitted the name can have a special meaning.

Examples:	Comments:
C_FCCH_A	FCCH of cell A.
C_SCH_A	SCH of cell A.
C_BCCH_A_1	First instance of a BCCH in cell A.
C_CBCH_A	CBCH of cell A.
C_PCH_A_1	First instance of a PCH in cell A (there is one instance of the PCH per BCCH in the cell).
C_AGCH_A_1	First instance of a AGCH in cell A (there is one instance of the AGCH per BCCH in the cell).
C_RACH_A_1	First instance of a RACH in cell A.
C_FACCHF_A_1	First instance of a full rate FACCH in cell A.
C_FACCHH_A_1	First instance of a half rate FACCH in cell A (all subchannels).
C_FACCHH0_A_1	Subchannel 0 of the first instance of a half rate FACCH in cell A.
C_SACCHF_A_1	First instance of a full rate SACCH in cell A.
C_SACCHH_A_1	First instance of a half rate SACCH in cell A (all subchannels).
C_SACCHH1_A_1	Subchannel 1 of the first instance of a half rate SACCH in cell A.
C_SACCHC4_A	SACCH/4 of cell A (all subchannels).
C_SACCHC43_A	Subchannel 3 of the SACCH/4 of cell A.
C_SACCHC8_A_1	First instance of a SACCH/8 in cell A (all subchannels).
C_SACCHC87_A_1	Subchannel 7 of the first instance of a SACCH/8 in cell A.
C_SDCCH4_A	SDCCH/4 of cell A (all subchannels).
C_SDCCH42_A	Subchannel 2 of the SDCCH/4 of cell A.
C_SDCCH87_A_1	First instance of a SDCCH/8 in cell A (all subchannels).
C_SDCCH87_A_1	Subchannel 7 of the first instance of a SDCCH/8 in cell A.

This list covers all types of combinations used in the TTCN for the time being.

When the meaning refers to 'all' subchannels of a channel (eg. C_FACCHH_A_1), it means that all active subchannels of this channel activated previously by OM_ChConf, are associated with a logical channel identifier in the OM_Assoc.

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OM_BmInfo(ch:LOGICCH; mode:B_8)
Result Type:	BOOLEAN
Comments:	mode is of type BITSTRING[8], imode is a part of the channel mode IE

Description

OM_BmInfo operation checks whether the MS starts transmiting Bm channel information according to the new channel mode `mode'. The operation returns TRUE if the result of the check is correct otherwise the TSO returns FALSE.

Detailed Comments:

Test Suite Operation Definition

Operation Name:	OM_ChangeRFOfCell(cellid1:CellID; bspwr1:INTEGER)
Result Type:	BOOLEAN
Comments:	

Description

This TSO is to change the RF level of the cell 'cellid1' to the RF level 'bspwr1', then the TSO returns TRUE.

The major purpose of the TSO is to force the MS to reselect a new cell according to the GSM 05.08, section 6.6.2. The following example shows the use of the TSO.

The default cell selection parameters in the ATS are set as:

Cell reselect hysteresis: 12db

MX TXPWR MX CCH: minimu level (0)

ACS: no additional cell selection parameter (0)

RXLEV ACCESS MIN: minimum (0).

Therefore the path loss criterion parameter C1 equals to the reselection criterion parameter C2. In the whole ATS the C1 and C2 keeps greater than 10 (e.g. TC_26_7_4_2_3) for a cell not suitable. The MS received level average from a not suitable cell keeps also greater than 10 (-100 dBm) corresponding to the reference sensitivity level of DCS 1800 MS.

For GSM arfcn = 20, F = 941 MHz , the fields strengths (E) on tester integral antenna:

E (dB microV/m) >= (-110 + 10) (dBm) + 20log(941) + 77.2 = 36.7,

For DCS arfcn = 590, F = (-110 + 10) (dBm) + 20log(1820.8) + 77.2 = 42.4 (the formula see GSM 05.05, section 5 and GSM 05.08, section 8.1.4).

In order to keep C1, C2 greater 10, a reasonable field strength making a cell not suitable

is set to C_E_notsuitable = 45 in comparison with the default field strength C_E_default = 63. Considering that the cell recelect hysteresis is set to 12 db, the field strength of a cell being reselected need to be set to greater than 57. In the ATS the default value C_E_default (63) and C_E_notsuitable (45) are used for this TSO to force MS making cell reselection.

C1 needs to be less than zero to force the MS reselect a better cell. For GSM, E <= 26.7, for DCS, E<=32.4. The value C_E_notsuitable_sameLAI = 24 applies to the TSO to force the MS making cell reselection.

Detailed Comments:

Test Suite Operation Definition

Operation Name: OM_ChangeRFOf2Cells(cellid1:CellID; bspwr1:INTEGER; cellid2:CellID; bspwr2:INTEGER)

Result Type: BOOLEAN

Comments:

Description

This TSO is used to change the RF level of the cell 'cellid1' to the RF level 'bspwr1' and the RF level of the cell 'cellid2' to the RF level 'bspwr2' [dBuVemf], then the TSO returns TRUE.

The main use of this TSO is to force the Mobile Station(MS) to select a new cell according to the GSM 05.08 and GSM 03.22.

For the MS to select the new cell then the path loss criterion parameter (C1) of the old will have to be less than zero (C1 < 0). This criterion is defined as:

C1 = (A - max(B, 0))

when

A - Received Level Average - (RXLEV_ACCESS_MIN)
B - MS_TXPWR_MAX_CCH - P {non DCS 1800}
MS_TXPWR_MAX_CCH + POWEROFFSET - P {DCS 1800}

RXLEV_ACCESS_MIN - Minimum received level at the MS required for access to the system.
MS_TXPWR_MAX_CCH - Maximum TX power level an MS may use when accessing the system until otherwise commanded.
POWER OFFSET - The power offset to be used in conjunction with the MS TXPWR MAX CCH
parameter by the class 3 DCS 1 800 MS.
P - Maximum RF output power of the MS.

All values are expressed in dBm.

In most test cases MS_TXPWR_MAX_CHH and the POWER_OFFSET will be set at their minimum values thus C1 = A (B always being negative). Suitable levels of cells will be either 63 dBuV(C_E_default) ,53 dBuV(C_E_neighbourdefault) or 45 dBuV(C_E_suitable). Assuming that the MS uses a RXLEV_ACCESS_MIN between -110 and -115 dbM then to ensure that C1 < 0 the test system's output needs to be set to ensure the level input detected by the MS is less than -123 dBm or lower (-115 plus an 8dB for discrimination between C1 or C2 values and 0, GSM 11.10-1 section 20). This value of -123 dBm corresponds to -10 dBuV (C_E_notsuitable)

Detailed Comments:

Test Suite Operation Definition

Operation Name: OM_ChConf(bspwr: INTEGER; mspwr:INTEGER; acttype:BITSTRING; chmod:CHMOD; ta:TA; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; td:INTEGER; fn:INTEGER; babr, cch_con, bpm:B_3; pgfil:PG1_RQ_PDU; dtxu:BITSTRING; dtxd:BITSTRING; cell:CellID; lgch1:LOGICCH; lgch2:LOGICCH; lgch3:LOGICCH; lgch4:LOGICCH; lgch5:LOGICCH; lgch6:LOGICCH; lgch7:LOGICCH; lgch8:LOGICCH; lgch9:LOGICCH)

Result Type: BOOLEAN

Comments:

Description

The operation sets the configuration of a basic physical channel according to the input parameters and map logic channel combination onto it:

- bspwr: base station power level in dBuv;
- mspwr: mobile station power level in "power control level";
- acttype: type of activation:
  - '000'B activation for intra-cell channel change ---- immediate assignment;
  - '001'B activation for intra-cell channel change ---- assignment;
  - '010'B activation for inter-cell channel change ---- asynchronous handover;
  - '011'B activation for inter-cell channel change ---- synchronous handover;
  - '100'B activation for additional assignment;
    NOTE: the types defined above are all the same, no real difference
  - '101'B activation for receiving only;
- chmod: channel mode;
- ta: timing advance ---- receive timing ta bit periods in advance of the normal receive timing if the channel is a dedicated channel ;
- slot: time slot;
- tsc: Training sequence code for normal bursts;
- rf: ARFCN or hopping parameters;
- chcmbn: logic channel combination mapping to the physical channel;
- td: timing difference between the cell and conceptual timing reference ---- to make 2 cells with different timebase;
- fn: initial frame number offset to the timing base counter ---- to make 2 cells with different frame numbers;
- babr number of blocks reserved for access grant
- cch-con CCCH/SDCCHs configuration
- bpm multiframe period for transmission of PAGING REQUEST
- pgfil: paging filling contents;
- dtxu: mobile station discontinuous transmission:
  - '0'B mobile station discontinuous transmission is not applied;
  - '1'B mobile station discontinuous transmission is applied;
- dtxd: base station discontinuous transmission:
  - '0'B base station discontinuous transmission is not applied;
  - '1'B base station discontinuous transmission is applied;
- cell: cell identifier;
- lgch1: logic channel identifier for the channel FCCH;
- lgch2: logic channel identifier for the channel SCH;
- lgch3: logic channel identifier for the channel BCCH;
- lgch4: logic channel identifier for the channel PCH;
- lgch5: logic channel identifier for the channel AGCH;
- lgch6: logic channel identifier for the channel RACH;
- lgch7: logic channel identifier for the channel FACCH or SDCCH;
- lgch8: logic channel identifier for the channel SACCH;
- lgch9: logic channel identifier for the channel CBCH;

For parameters lgch1 to lgch9 "dummy" in the actual parameter list means that the corresponding parameters are not used.

In the test cases the logic channel identifiers lgch1 to lgch9 are used to refer the logic channels configured by the operation. There are only generic identifiers for SDCCH4 or SDCCH8 or FACCHH channel and SACCHC4 or SACCHC8 or SACCHH, identifiers for subchannels of SDCCH4 or SDCCH8 or FACCHH and SACCHC4 or SACCHC8 or SACCHH are linked to the generic identifiers by OM_Assoc operation and in turn refer to the sub logic channels configured by the operation.

Channel re-use whithiin OM_ChConf:

When an OM_ChConf calls is made then the following rules can be applied for deciding whether or not to re-use existing physical channel.

Decode the 'chcmbn' (channel combination) parameter to determine the new channel type.

If new channel = "C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4" (combined) or

if new channel = "C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4" (combined + SMS) then re-use any existing control channel configuration with same CellID.

If new channel = "C_FCCH_SCH_BCCH_CCCH" (non-combined) then use any combined, combined + SMS or non-combined control channel configuration with the same CellID. If new channel = "C_BCCH_CCCH_2", "C_BCCH_CCCH_3" or "C_BCCH_CCCH_4" (ccch-group) then re-use any existing ccch-group control channel configuration with the same CellID and the slot, or re-use any existing combined, combined + SMS with the same CellID. If new channel = "C_TCHF_ACCHF_1", "C_TCHF_ACCHF_2", "C_TCHH_ACCHH_1", "C_TCHH_ACCHH_2" (traffic) or "C_SDCCH8_SACCHC8_1", "C_SDCCH8_SACCHC8_2", "C_SDCCH8_SACCHC8_3" (dedicated) then re-use any existing traffic or dedicated channel with the same channel description.
Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OM_ChMdModi(ch:LOGICCH; chmod: CHMOD)
Result Type:	BOOLEAN
Comments:	
Description	
OM_ChMdModi operation passes channel mode IE for channel `ch` to lower layer emulator and requests the emulator to set the `ch` to the mode `chmod`. After the mode changed, it returns.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_CphMdChg(ch:LOGICCH; cphmod: CPHMS; key:BITSTRING)
Result Type:	BOOLEAN
Comments:	
Description	
OM_CphMdChg operation passes ciphering parameters for channel `ch` to lower layer and sets the lower layer emulator into a special operation mode in which lower layer starts the 3-step ciphering mode setting sequence when the next L3 message on the channel `ch` arrives.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_CphMd(ch:LOGICCH; cphmod: CPHMS; key:BITSTRING)
Result Type:	BOOLEAN
Comments:	
Description	
OM_CphMd operation passes ciphering parameters for channel `ch` to lower layer and sets the channel `ch` in the ciphering mode `cphmod`. This operation puts the channel `ch` into specified ciphering mode `cphmod` immediately without the 3-step ciphering mode setting sequence.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_ComingFn(ch:LOGICCH)
Result Type:	FN
Comments:	
Description	
OM_ComingFn operation returns the frame number (FN modulo 42432) which is about 5 seconds later than current frame number and is able to carry L3 message on the channel `ch`.	
Detailed Comments:	The delay of about 5 seconds ensures that there is still enough time left for higher layer controller to prepare next TTCN send event after the OM_ComingFn returns. the exact delay value is up to the implementor.

Test Suite Operation Definition	
Operation Name:	OM_CorruptPgBlks(pch:LOGICCH; pgg:PGG; blks:INTEGER)
Result Type:	BOOLEAN
Comments:	
Description	
This TSO corrupts the next 'blks' number of paging blocks of the paging channel 'pch' for paging group 'pgg'. The major purpose of the TSO is to force the MS to reselect a new cell by forcing the downlink signalling failure counter DSC to be < 0. The initial DSC value for the MS shall be initialised to a value equal to the nearest integer to 90/N where N is the BS_PA_MFRMS parameter for the cell the MS is camped on (see GSM 05.02). The DSC counter is decreased by 4 whenever it cannot decode a message in its paging subchannel.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_Deactivate(ch1, ch2:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
OM_Deactivate operation deactivates the logical channels `ch1` and 'ch2', then returns TRUE. The ch1 can be a DCCH or a TCH, the ch2 is the associated SACCH. Deactivation of an active SDCCH channel can be done by stopping transmission; this means that for the bursts on the BCCH carriers, dummy bursts shall be transmitted, for other bursts nothing shall be transmitted. Alternatively the deactivation of a SDCCH can be done by transmitting bad blocks on the SACCH with the parity bits set incorrectly. This will lead to a radio link failure in the MS. Deactivation of a TCH channel can be done by stopping transmission if it is non-BCCH. Deactivation of a SACCH_T can be done by transmitting bad blocks on the SACCH_T with the parity bits set incorrectly. The wrong channel coding on the SACCH will lead to a radio link failure in the MS after several seconds depending on the system information messages..	
Detailed Comments:	The TSO is called in TC_26_7_3_2, TC_26_7_4_3_2, TC_26_7_4_3_3, TC_26_7_4_3_4 on a SDCCH/4/SACCH/C4 channel in downlink, TC_26_2_4_1, TC_26_2_4_2, TC_26_8_2_1, TC_26_8_2_2, TC_26_8_2_3 on TCH/SACCH_T channel in both directions.

Test Suite Operation Definition	
Operation Name:	OM_FHCHK(ch:LOGICCH; ca:CCHD; ma:MA; cd:CHD; t :INTEGER; fn:FN)
Result Type:	BOOLEAN
Comments:	
Description	
After the SS has been configured to start / change frequency hopping at a particular start time, this TSO instructs the SS to check whether the MS correctly uses the hopping frequencies on and after the expected start time according to the frequency parameters. If `t` = 0 or 32024 <= `t` <= 42431 the operation checks that the new frequencies starts "without delay". If 0 < `t` <= 31000 the operation checks that the new frequencies starts "on" the frame number of (X + t) mod 42432, where x = 51 * ((fn.t3 - fn.t2) MOD 26) + fn.t3 + 1326 * fn.t1. If 31000 < `t` < 32024, or `t` < 0 or `t` > 42431 then the test has been configured incorrectly and should be re-run with a permitted value for `t`. This check is performed at the uplink RF burst level by sampling for a period of 3 seconds from the start time and should therefore be called immediately after the SS has been configured to start / change hopping. The operation returns TRUE if the MS behaviour is correct, otherwise it returns FALSE. Note: before this TSO is called, a layer 3 message must be sent to the MS to command the MS to start / change hopping using the same frequency hopping parameters.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_FreeResource(ch :LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
OM_FreeResource operation instructs the lower layer emulator to release the resources which have been allocated to the logical channel "ch". After the resources are released the operation returns TRUE.	
Detailed Comments:	In some test cases (most of them are hanhover test cases) diffirent channels are needed while the test goes on. the purpose of the operation is to reuse lower layer resources (e.g. tranceivers), which have been allocated to the previous channel, for configing another channel required in the next testing step.

Test Suite Operation Definition	
Operation Name:	OM_FreqDef(st:STRT; ma:MA; ch :LOGICCH; chd: CHD; ca:CCHD; frmn: FN)
Result Type:	BOOLEAN
Comments:	
Description	
OM_FreqDef operation passes the frequency hopping parameters for channel `ch` to lower layer emulator. The emulator starts using these frequency parameters at the frame number indicated by starting time `st`. The parameter 'frmn' indicates a reference frame number at which the first burst of the message, FREQUENCY REDEFINITION, ASSIGNMENT COMMAND or HANDOVER COMMAND, containing starting time is sent in the concerned cell. If 0 <= (st - frmn) mod 42432 <= 31623, the indicated time is the next time when frame number mod 42432 is equal to st. If 32024 <= (st - frmn) mod 42432 <= 42431, the indicated time has already elapsed. After completely sending out the message containing the parameter starting time the tester starts frequency hopping.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_GetHoaccPara(ch:LOGICCH)
Result Type:	HOACC_PARA
Comments:	
Description	
OM_GetHoaccPara instructs the L 2 emulator to return the number of handover access bursts received on the channel 'ch', along with the minimum and maximum measured timing advance in bits modulo 256, and the minimum and maximum measured powers in dBm modulo 256.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_GetL1Hd(ch:LOGICCH)
Result Type:	L1HD
Comments:	
Description	
OM_GetL1Hd instructs the L 2 emulator to read the L 1 header of the next SACCH frame received on channel `ch` and return this value.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_LowerLayerFail(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
This operation is to force the lower layer failure, then it returns. The lower layer failure can be any one of layer 1 failures or layer 2 failures. - layer 1 failure, see GSM 05.08; - layer 2 failure, see GSM 04.06.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_NoL2Ack(i:INTEGER; ch: LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
When the parameter i>0 the layer 2 emulator is put into a special operation mode then the TSO returns. In this special mode the layer 2 emulator shall not aknowledge the 'i'th occurence of the L2 I frame which has the more data bit "M" = 0 on the channel 'ch'. The layer 2 emulator resumes normal operation when - either the TSO is called again with i = 0 on the same 'ch', - or a data link SABM message is received on the 'ch'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_NotAckSetup(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
This operation puts the layer 2 emulator into a special operation mode then returns. In this special mode the layer 2 emulator will not acknowledge the last L 2 frame which carries the SETUP message, and the layer 2 emulator resumes normal operation after this L 2 frame.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_NoUAforDISC(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
OM_NoUAforDISC and OM_ResumUAforDISC are a pair of TSOs. OM_NoUAforDISC forces the L2 emulator of the tester to enter a specific mode. OM_ResumUAforDISC resumes the L2 emulator to a normal L2 operational mode. In the specific mode on receiving a DISC on the channel 'ch', the L2 emulator - does not respond with a UA, - remains in the multiple-frame established state, - indicates that the DISC has been received (by means of subsequent DL_RELEASE_INDICATION ASPs). Then the TSO returns TRUE.	
Detailed Comments:	The TSO is called in TC_26_6_12_2 and TC_26_6_12_4.

Test Suite Operation Definition	
Operation Name:	OM_PgFill(cellid:CellID; msg:PG1_RQ_PDU)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
OM_PgFill operation sets up the contents of paging filling frame to be sent on all paging subchannels continuously. If the contents of paging filling frame in cell `cellid` have not been set up, the operation sets the contents to `msg`. If paging filling frame in cell `cellid` has been set up, the operation changes the paging filling message to `msg`. After the required action has been taken it returns.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_Reactivate(ch1, ch2:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
The TSO OM_Reactivate reactivates the logical channels `ch1` and 'ch2' which were previously deactivated by the OM_Deactivate. Then the TSO returns TRUE. The ch1 is an SDCCH, the ch2 is the associated SACCH. Reactivation of deactivated SDCCH/SACCH channels means to restart normal transmissions both on the SDCCH and the associated SACCH as specified in the ETS 300 574 (GSM 05.02).	
Detailed Comments:	The TSO is called in TC_26_7_3_2, TC_26_7_4_3_2, TC_26_7_4_3_3, TC_26_7_4_3_4 on the SDCCH/4 / SACCH/C4 channels.

Test Suite Operation Definition	
Operation Name:	OM_ReturnFn(ch:LOGICCH)
Result Type:	FN
Comments:	
Description	
OM_ReturnFn operation returns the frame number (FN modulo 42432) on which the last L3 message was sent on logic channel `ch`.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_ResumUAforDISC(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
OM_NoUAforDISC and OM_ResumUAforDISC are a pair of TSOs. OM_NoUAforDISC forces the L2 emulator to enter a specific mode. OM_ResumUAforDISC resumes the L2 emulator to a normal L2 operational mode. OM_ResumeUAforDISC cancels the effect of OM_NoUAforDISC, returning L2 on the channel 'ch' to normal operation in the multiple-frame established state. Then the TSO returns TRUE.	
Detailed Comments:	The TSO is called in TC_26_6_12_2 and TC_26_6_12_4.

Test Suite Operation Definition	
Operation Name:	OM_SendSMSCBWhilePaging(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The TSO makes the lower layers send the second SMSCB message at the same time as the MS is paged. This shall be achieved by paging the MS immediately after the first block of the CB message has been sent. The SS shall ensure that the page is transmitted on the radio interface prior to the transmission of the 4th block of the CB message.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_SendNextOn(ch:LOGICCH; fn:FN)
Result Type:	BOOLEAN
Comments:	
Description	
OM_SendNextOn operation sets the lower layer emulator into a special operation mode then returns. In the special mode the lower layer emulator sends the next L3 message on the indicated frame number `fn` on the 'ch' channel. After the next L3 message is sent, the lower layer resumes normal operation automatically.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_SendNextOnHO(ch:LOGICCH; cnt:INTEGER)
Result Type:	BOOLEAN
Comments:	
Description	
OM_SendNextOnHO operation sets the lower layer emulator into a special operation mode then returns. In the special mode the lower layer emulator waits for the number of Handover access bursts specified by the parameter 'cnt' on channel 'ch' before sending the next L 3 message. After the message is sent, the lower layer resumes normal operation automatically.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_StartMsrReport(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The operation starts the reporting of received MEASUREMENT REPORT message to layer 3 emulator, then it returns with value TRUE. Measurement Reports shall be disable by default. By default means that the tester filters them out from the TTCN in-buffer. This operation enables the reports entering the TTCN in-buffer. The OM_StopMsrReport operation disables the reports entering the TTCN in-buffer. When starting each test case, the measurement reports are filtered out by default.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_StopAllBCCH(cell1, cell2, cell3, cell4, cell5, cell6, cell7, cell8:CellID)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The TSO stops the RF transmission on all BCCH channels including dummy bursts in the indicated cells, but keeps the uplink CCCH reception active. After all transmission have been stopped the TSO returns TRUE. If a cell in the parameter list is not in use a "dummy" value needs to be assigned to it by the calling test case. The TSO is called in TC_26_3_3.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_StopCell(cell:CellID)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The TSO stops completely all RF transmission of the specific cell `cell`, including the dummy bursts on other slots. After all RF transmission have been stopped the TSO returns TRUE. The cell shall be deleted from the lower layer tester and therefore if the cell 'cell' is required again by the calling test case it needs to be re-configured. The TSO is called in TC_26_3_4, TC_26_6_6_1, TC_26_7_4_2_2_1, TC_26_7_4_2_2_2, TC_26_7_4_6, TC_31_6_2_3.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_StopMsrReport(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	The result returned by the operation is not used.
Description	
The operation stops the reporting of received MEASUREMENT REPORT message to layer 3 emulator, after measurement report stops it returns with value TRUE. Measurement Reports shall be disable by default. By default means that the tester filters them out from the TTCN in-buffer. This operation disables the reports entering the TTCN in-buffer. The OM_StartMsrReport operation enables the reports entering the TTCN in-buffer. When starting each test case, the measurement reports are filtered out by default.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_StopVaryRFOfCell(ch:LOGICCH)
Result Type:	BOOLEAN
Comments:	
Description	
This TSO tells channel 'ch' to stop varying the RF signal level, leaving the RF signal level for that channel at its current setting.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OM_VaryRFOfCell(ch:LOGICCH; delay1, bspwr1, delay2, bspwr2:INTEGER)
Result Type:	BOOLEAN
Comments:	
Description	
This TSO tells channel 'ch' to change the RF level to 'bspwr1' after 'delay1' milliseconds, then to change the RF level to 'bspwr2' after a further 'delay2' milliseconds. This sequence is repeated until "OM_VaryRFOfCell" is called again or "OM_StopVaryRFOfCell" is called.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ACMIncCHK(para:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
the OO_ACMIncCHK operation requests (e.g. from the control console of the test system) the test operator to read the value of ACM on SIM and check the increment of the value. Then the operator informs the test system whether the increment is as description `para`, if it is as the description the operation returns TRUE, otherwise returns FALSE. The value of ACM can be read either via MMI or by removing the SIM and using SIM reader.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ACMReading
Result Type:	BOOLEAN
Comments:	
Description	
the OO_ACMReading operation requests (e.g. from the control console of the test system) the test operator to read and note the value of ACM on SIM. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE. The value of ACM can be read either via MMI or by removing the SIM and using SIM reader.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ACMSetting
Result Type:	BOOLEAN
Comments:	
Description	
the OO_ACMSetting operation requests (e.g. from the control console of the test system) the test operator to reset the ACM to zero and set the ACMmax to 2 units. After the operator finishes the action and informs the test system it returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_AddPwrAmp
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to add power amplification at the Mobile Station under test. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_AltIndCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the alerting indication is generated. If the operator informs (e.g. from the control console of the test system) the test system that the alerting indication is generated, the operation returns TRUE. If the operator informs the test system that the alerting indication is not generated, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CalledNumCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the called party number is correctly displayed on the mobile station. If the operator informs (e.g. from the control console of the test system) the test system that the display is correct, the operation returns TRUE. If the operator informs the test system that the display is wrong, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CalledPtyNumCHK(num:CDPN)
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the called party number `num` is the one that was entered into the MS. If the operator informs (e.g. from the control console of the test system) the test system that the number is correct, the operation returns TRUE. If the operator informs the test system that the number is wrong, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckAllSMPresentBut4th
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the mobile station under test has stored all short messages during the test case but not the 4th one which should have been replaced by the 5th(cf. GSM 11.10, clause 34.2.7.4). The messages are displayed. Then the operation returns. TRUE: All SM present but not the 4th FALSE: 4th SM present or 1st, 2nd, 3rd or 5th missing.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckCBSMReceived(NumbOfIA5chara: INTEGER; msg:SMSCBpack)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check whether the MS under test has correctly received the cell broadcast short message represented by 'msg' . Then the TSO returns TRUE: if the CBSM has been correctly received, FALSE: if non or an incorrect CBSM has been received. The TSO needs to convert the 'msg' into IA5 string according to the packing character rules specified in the section 6.1.2 of ETS 300 628 (GSM 03.38). The 'NumbOfIA5chara' indicates the number of IA5 characters in the 'msg'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckMCEFOnSIM
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the Memory Capacity Exceeded Flag has been set on the SIM simulator and to inform the test system of the result of the checking. If the checking succeds the operation returns TRUE, FALSE otherwise.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckMCEFOnSIMUnset
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the Memory Capacity Exceeded Flag has been unset on the SIM simulator and to inform the test system of the result of the checking. If the checking succeds the operation returns TRUE, FALSE otherwise.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckMessageDisplayed(NumbOfIA5chara:INTEGER; msg:OCTETSTRING)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the mobile station under test indicates that an SM has arrived. If the MS provides the functionality to display MT messages, it is checked that the message contents represented by 'msg' is correctly displayed. Then the operation returns TRUE: if the MS indicates SM reception (and displays correct message), FALSE: if the MS does not indicate SM reception (or displays incorrect message). The TSO needs to convert the 'msg' into IA5 string according to the packing character rules specified in the section 6.1.2 of ETS 300 628 (GSM 03.38). The 'NumbOfIA5chara' indicates the number of IA5 characters in the 'msg'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_CheckUssdStringDisplayed(strg: IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check if the mobile station under test displays the Ussd String 'strg'. Then the operation returns. TRUE: MS displays the correct Ussd String. FALSE: MS does not display the correct Ussd String.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ConnectSIMSimulator
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to connect the SIM simulator to the mobile station under test. For the contents of the SIM simulator the operator refers to the initial conditions of the test case in GSM 11.10-1. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_DepressEndKey
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to depress the 'END' key of the Mobile Stations keypad, then returns.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_DialCalledNum
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to dial the called party number at the Mobile Station under test but not press the "SEND" key (i.e. not initiate the call setup). After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_DisplaySMAndSendReplySM(n, NumbOfIA5chara: INTEGER; msg:OCTETSTRING)
Result Type:	OCTETSTRING
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to display the 'n'-th short message received in the arriving order and to check whether the short message contents represented by 'msg' is correctly displayed. Then the TSO asks the operator to send a reply short message from the mobile. After the operator finishes the action, the operation returns with the message contents sent in OCTETSTRING. The TSO needs to convert the 'msg' into IA5 string and the IA5 text entered by the operator into OCTETSTRING according to the packing character rules specified in the section 6.1.2 of ETS 300 628 (GSM 03.38). The 'NumbOfIA5chara' indicates the number of IA5 characters in the 'msg'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_DTMFIndCHK(character:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the `character` is indicated by the DTMF indicator of the MS. If the operator informs (e.g. from the control console of the test system) the test system that the MS does not give the indication of `character`, the operation returns FALSE. If the operator informs the test system that the MS gives the indication, the operation returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_EmptyMessageStorage
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to empty the message storage of the mobile station under test. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_EnterPswd(pswd:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This TSO displays a message on the control console to inform the operator that he is to enter a given password (the actual password is given by 'pswd') on the mobile. The operation waits for the operator to acknowledge the request, which may be done before or after entering the password into the mobile, and then returns the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_HookOff
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to answer the mobile terminating call at the Mobile Station under test. This operation waits for the operator to acknowledge the request which may be done before or after the call has been answered and then returns the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_Telephony
Result Type:	BOOLEAN
Comments:	
Description	
This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or configuration of the MS in such a way that the MS is able to successfully receive the call for telephony service. If the MS supports both telephony without immediate connection mode and telephony with immediate connection mode, set the MS in telephony without immediate connection mode. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE. This TSO is used in MO call test cases except TC_11_2, where this TSO is call for single-numbering scheme MT testing.	
Detailed Comments:	used for TS11 and TS12

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS2xorBS3x_MO(srv:SERVICES; itc, sacp:B_3; ce:B_2; rate:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS, when trigged, is able to initiate the call for the BS2x or BS3x service `srv` with the specified itc, sacp, ce values. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE. This TSO is used in MO call test cases except TC_11_2, where this TSO is call for single-numbering scheme MT testing. Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT paramters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS. The possible values are: 1) srv ----- basic service - C_Async300 Basic Service 21, Asynchronous 300 bps - C_Async1200 Basic Service 22, Asynchronous 1.2 kbps - C_Async120075 Basic Service 23, Asynchronous 1200/75 bps - C_Async2400 Basic Service 24, Asynchronous 2.4 kbps - C_Async4800 Basic Service 25, Asynchronous 4.8 kbps - C_Async9600 Basic Service 26, Asynchronous 9.6 kbps - C_Sync1200 Basic Service 31, Synchronous 1.2 kbps - C_Sync2400 Basic Service 32, Synchronous 2.4 kbps - C_Sync4800 Basic Service 33, Synchronous 4.8 kbps - C_Sync9600 Basic Service 34, Synchronous 9.6 kbps 2) itc ------ information transfer capability - '001'B unrestricted digital information - '010'B 3.1 kHz audio, ex PLMN 3) sacp ---- signalling access protocol - '001'B I.440/450 - '010'B X.21 - '101'B X.28 - non dedicated PAD - '110'B X.32 4) ce ------ connection element - '00'B transparent - '01'B non transparent 5) rate ----- channel rate - C_Full full rate channel - C_Half half rate channel	
Detailed Comments:	Used for BS21, BS22, BS23, BS24, BS25, BS26, BS31, BS32, BS33, BS34

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS4x_MO(srv:SERVICES; ce:B_2; rate:IA5String)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, orthe MS configuration in such a way that the MS, when trigged, is able to initiate the call for the BS4x service `srv` with specified ur and ce values. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:

1) srv ----- basic service

- C_PAD300 Basic Service 41, PAD access 300 bps
- C_PAD1200 Basic Service 42, PAD access 1.2 kbps
- C_PAD120075 Basic Service 43, PAD access 1200/75 bps
- C_PAD2400 Basic Service 44, PAD access 2.4 kbps
- C_PAD4800 Basic Service 45, PAD access 4.8 kbps
- C_PAD9600 Basic Service 46, PAD access 9.6 kbps

2) ce ------ connection element

- '00'B transparent
- '01'B non transparent

3) rate ----- channel rate

- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: Used for BS41, BS42, BS43, BS44, BS45, BS46

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS5x_MO(srv:SERVICES; ur:B_4; rate:IA5String)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, orthe MS configuration in such a way that the MS, when trigged, is able to initiate the call for the BS5x service `srv` with the specifyied ur value. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:

1) srv ----- basic service

- C_Packet2400 Basic Service 51, Packet access 2.4 kbps
- C_Packet4800 Basic Service 52, Packet access 4.8 kbps
- C_Packet9600 Basic Service 53, Packet access 9.6 kbps

2) ur ------ user rate

- '0011'B 2400 bit/s
- '0100'B 4800 bit/s
- '0101'B 9600 bit/s

3) rate ----- channel rate

- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: Used for BS51, BS52, BS53

Test Suite Operation Definition

Operation Name: OO_IFsetup_BS61orBS81_MO(srv:SERVICES; ur:B_4; sa:B_1; ce:B_2; rate:IA5String)

Result Type: BOOLEAN

Comments:

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS, when trigged, is able to initiate the call for BS61 or BS81 with the specified ur, ce and sa values, and the first bearer capability is speech. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

This TSO is used in MO call test cases except TC_11_2, where this TSO is call for single-numbering scheme MT testing.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:

1) srv ----- basic service

- C_AltSpchData Basic Service 61, Alternatie Speech/data
- C_SpchData Basic Service 81, Speech followed by data

2) ur ------ user rate

- '0001'B 300 bit/s
- '0010'B 1200 bit/s
- '0111'B 1200/75 bit/s
- '0011'B 2400 bit/s
- '0100'B 4800 bit/s
- '0101'B 9600 bit/s

3) sa ------ synchronous/asynchronous

- '0'B synchronous
- '1'B asynchronous

4) ce ------ connection element

- '00'B transparent
- '01'B non transparent

5) rate ----- channel rate

- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: used for BS61 and BS81

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_TS6x_MO(srv:SERVICES; ur:B_4; ce:B_2; rate:IA5String)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS, when trigged, is able to initiate the call for TS61 and TS62 with the specified ur and ce values, and the first bearer capability is speech. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

This TSO is used in MO call test cases except TC_11_2, where this TSO is call for single-numbering scheme MT testing.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

The possible values are:
1) srv ----- basic service
- C_AltSpchFax Teleservice 61, Alternatie Speech and Facsimile group 3
- C_AutoFax Teleservice 62, Automatic Facsimile group 3

2) ur ------ user rate
- '0011'B 2400 bit/s
- '0100'B 4800 bit/s
- '0101'B 9600 bit/s

3) ce ------ connection element
- '00'B transparent
- '01'B non transparent

4) rate ----- channel rate
- C_Full full rate channel
- C_Half half rate channel

Detailed Comments: used for TS61 and TS62

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS2x_MT(srv:SERVICES; itc:B_3)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS is able to successfully receive the call for the BS2x service `srv` with the specified itc value. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT paramters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

Detailed Comments: Used for BS21, BS22, BS24, BS25, BS26

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS3x_MT(srv:SERVICES; itc, sacp:B_3)
Result Type:	BOOLEAN
Comments:	

Description

This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, orthe MS configuration in such a way that the MS is able to successfully receive the call for the BS3x service `srv` with specified itc and sacp values. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE.

Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.

Detailed Comments: Used for BS31, BS32, BS33, BS34

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_BS61orBS81_MT(srv:SERVICES; ur:B_4; sa:B_1)
Result Type:	BOOLEAN
Comments:	
Description	
This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS is able to successfully receive the call for BS61 or BS81 with the specified ur and sa values. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE. Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.	
Detailed Comments:	used for BS61 and BS81

Test Suite Operation Definition	
Operation Name:	OO_IFsetup_TS6x_MT(srv:SERVICES; ur:B_4)
Result Type:	BOOLEAN
Comments:	
Description	
This TSO requests (e.g. from the control console of the test system) the test operator to setup the external interface, or the MS configuration in such a way that the MS is able to successfully receive the call for TS61 and TS62 with the specified ur value. After the operator finishes the required action and informs the test system, the TSO returns with value TRUE. Three negotiable parameters, number of stop bit, number of data bit, parity information, are asked as global IXIT parameters for all TE configurations. If necessary, the test operator refers to the manual provided by manufacturer to configure the MS.	
Detailed Comments:	used for TS61 and TS62

Test Suite Operation Definition	
Operation Name:	OO_InCallModi(srv:SERVICES)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to initiate a in-call modification at the Mobile Station under test, which is triggered by the calling tone identification (CNG) received by the MS, for the basic service 'srv', then returns TRUE. The calling test step can pass the 'srv' value either as TSPX or constant to the TSO. The possible service values include (see also test suite constant declarations): 1) Teleservices - C_Telephony - C_EmgCall - C_AltSpchFax - C_AutoFax 2) Bearer services - C_Async300, C_Async1200, C_Async120075, C_Async2400, C_Async4800, C_Async9600 - C_Sync1200, C_Sync2400, C_Sync4800, C_Sync9600 - C_PAD300, C_PAD1200, C_PAD120075, C_PAD2400, C_PAD4800, C_PAD9600 - C_Packet2400, C_Packet4800, C_Packet9600 - C_AltSpchData - C_SpchData	
Detailed Comments:	

Test Suite Operation Definition

Operation Name: OO_InitCall(srv:SERVICES)
Result Type: BOOLEAN
Comments:

Description

This operation requests (e.g. from the control console of the test system) the test operator to initiate an MS originating call for the basic service `srv`, then returns TRUE. The calling test step can pass the 'srv' value either as TSPX or constant to the TSO. The necessary MS configuration has been done in the corresponding TSO OO_IFsetup_xx...xx_MO before this TSO is called.

The possible service values include (see also test suite constant declarations):

1) Teleservices

- C_Telephony ----- TS11
- C_EmgCall ----- TS12
- C_AltSpchFax ----- TS61
- C_AutoFax ----- TS62

2) Bearer services

- C_Async300 ----- BS21
- C_Async1200 ----- BS22
- C_Async120075 ----- BS23
- C_Async2400 ----- BS24
- C_Async4800 ----- BS25
- C_Async9600 ----- BS26
- C_Sync1200 ----- BS31
- C_Sync2400 ----- BS32
- C_Sync4800 ----- BS33
- C_Sync9600 ----- BS34
- C_PAD300 ----- BS41
- C_PAD1200 ----- BS42
- C_PAD120075 ----- BS43
- C_PAD2400 ----- BS44
- C_PAD4800 ----- BS45
- C_PAD9600 ----- BS46
- C_Packet2400 ----- BS51
- C_Packet4800 ----- BS52
- C_Packet9600 ----- BS53
- C_AltSpchData ----- BS61
- C_SpchData ----- BS81

Detailed Comments:

Test Suite Operation Definition

Operation Name: OO_InitNonCallSS
Result Type: BOOLEAN
Comments:

Description

This operation requests (e.g. from the control console of the test system) the test operator to initiate a non call related supplementary service which is supported by the MS.

After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.

Detailed Comments:

Test Suite Operation Definition	
Operation Name:	OO_InitSS(action: IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to initiate the required supplementary service by the MMI sequence `action`. After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_InServiceCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the Mobile Station under test is in normal service state ("Idle, updated")-- listening to CCCH and BCCH and with U1 UPDATED status. If the operator informs (e.g. from the control console of the test system) the test system that the MS does not give any service indication, the operation returns FALSE. If the operator informs the test system that the MS gives service indication, the operation returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_Key
Result Type:	BOOLEAN
Comments:	
Description	
This operation returns TRUE if a key of the SS' keyboard has been pressed. Otherwise it returns FALSE. After OO_PressKeyWhenInService was called, the TSO is then continuously called until the test operator hits a key or a timer expires.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_MptyCall
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to enter MultiParty MMI command at the Mobile Station under test, then it returns with the value TRUE after the operator finishes the action and informs the test system.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_MSSetupStoreClass1SMInMEMemory
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to setup the MS under test to store class 1 SM in the ME memory (by way of MMI, as described in PICS/PIXIT statement). After the operator finishes the action, he informs the test system and the operation returns with the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PLMNsCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the Mobile Station presents a list of available PLMNs. The list shall include the MNC and MCC of cells 1 to 7, but not cell 8 (for GSM900) or cells 1 to 6, but not cell 7 (for DCS1800). If the operator informs (e.g. from the control console of the test system) the test system that the MS correctly presents the list, the operation returns TRUE. If the operator informs the test system that the list is incorrect, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PLMNselModeAuto
Result Type:	BOOLEAN
Comments:	
Description	
This operation switches the PLMN selection mode of the MS to automatic selection. After correctly switching of the mode the operation returns TRUE otherwise FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PLMNselModeMan
Result Type:	BOOLEAN
Comments:	
Description	
This operation switches the PLMN selection mode of the MS to manual selection. After correctly switching of the mode the operation returns TRUE otherwise FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PowerUp
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to power up the Mobile Station under test and to inform after the test system, then it returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PowerDown
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to power down the Mobile Station under test and to inform after the test system, then it returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_PressKeyWhenInService
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to press any key when the MS shows the service indicator, then returns. The return value is always TRUE. Working with OO_Key together, the TSO displays a prompt to the test operator and then immediately returns.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_RecallAndDisplaySM(NumbOfIA5chara: INTEGER; msg: OCTETSTRING)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to recall and display an SM stored in the ME of the MS under test e.g. by means of the MMI. Then the operator checks whether the short message contents represented by 'msg' is correctly displayed. The TSO returns TRUE: if SM can be recalled and displayed correctly in comparison with the 'msg', FALSE: if SM can not be recalled or displayed, or displayed incorrectly in comparison with the 'msg'. The TSO needs to convert the 'msg' into IA5 string according to the packing character rules specified in the section 6.1.2 of ETS 300 628 (GSM 03.38). The 'NumbOfIA5chara' indicates the number of IA5 characters in the 'msg'.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_RemoveOneSM
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to remove one of the SM from the message store and then to inform the test system. It does not matter which short message has been removed. After that the operation returns with value TRUE. The returned value is not used in the ATS.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_RemvPwrAmp
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to remove the added power amplification at the Mobile Station under test and to inform after the test system, then the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_RFoutputCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the Mobile Station produces any RF output. If the operator informs (e.g. from the control console of the test system) the test system that the MS does not produce any RF output, the operation returns FALSE. If the operator informs the test system that the MS produces RF output, the operation returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SelPLMN(par_plmn:OCTETSTRING)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to select the given PLMN in par_plmn manually, and to inform after the test system then the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SendMOShortMessage
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to send a short message from the mobile station under test. The operation waits for the operator to acknowldge the request which may be done before or after sending the MO Short Message, and then returns the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SendSMSCOMMANDDel
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to send a SMS COMMAND message from the mobile station under test containing requiring to delete the previouly submited SM and then to inform the test system. Then, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SendSMSCOMMANDEnq
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to send a SMS COMMAND message from the mobile station under test containing an enquiry about the previouly submited SMand then to inform the test system. Then, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SetCellTestMode
Result Type:	BOOLEAN
Comments:	
Description	
Instructs operator to switch MS to 'Cell Test Mode', used in test case 20.7.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SetRefuseCall
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to enable call refusal on the MS. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ShortKeyDepr(ch:IA5String)
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to cause a DTMF tone to be generated, e.g. by short depression of the key `ch` on the Mobile Station under test. The operation waits for the operator to acknowldge the request which may be done before or after generating the DTMF tone, and then returns the value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIMIns
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to insert the SIM card into the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIM2Ins
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to insert the SIM card 2 into the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE. The SIM Card 2 shall contain the following parameter values, which are different from dafault values(SIM Card 1): IMSI= '001011234' HPLMN_search_period=6min.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIM3Ins
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to insert the SIM card 3 into the Mobile Station under test and turn the MS power on, after the test operator finishes the action and informs the test system it returns with value TRUE. The SIM Card 3 shall contain fixed dialling number allocated and activated.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIMRmv
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to remove the SIM card from the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIMSimulAttIndOK
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check whether the SIM simulator indicates an attempt made by the ME to store the short message in the SIM. The SIM simulator returns the status response "OK" ('90 00"). Then returns. Attempt indicated: TRUE Attempt NOT indicated: FALSE	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SIMSimulAttIndMemProblem
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to check whether the SIM simulator indicates an attempt made by the ME to store the short message in the SIM. The SIM simulator returns the status response "Memory Problem" ('92 40"). Then returns. Attempt indicated: TRUE Attempt NOT indicated: FALSE	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SSresultCHK(svc:INTEGER)
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check the user indication of the result of the supplementary service `svc`. If the operator informs (e.g. from the control console of the test system) the test system that the indication is correct, the operation returns TRUE. If the operator informs the test system that the indication is wrong, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SwitchOn
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to switch on (or if there is no switch then to restore the power to) the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_SwitchOff
Result Type:	BOOLEAN
Comments:	
Description	
This operation requests (e.g. from the control console of the test system) the test operator to switch off (or if there is no switch then to remove the power from) the Mobile Station under test. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, the operation returns with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_TCHThroConnCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the TCH is through connected. The SS has to generate a noise pattern so that the operator can check this. If the operator informs (e.g. from the control console of the test system) the test system that the TCH is through connected, the operation returns TRUE. If the operator informs the test system that the TCH is not through connected, the operation returns FALSE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_TermCall
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to terminate the ongoing call. After the operator completes the action and informs (e.g. from the control console of the test system) the test system, it returns TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_TguardTimeOut
Result Type:	BOOLEAN
Comments:	
Description	
This operation informs (e.g. from the control console of the test system) the test operator that the T_guard timer has expired during the test case leading to an inconclusive or fail verdict and returns immediately with value TRUE.	
Detailed Comments:	

Test Suite Operation Definition	
Operation Name:	OO_ToneStopCHK
Result Type:	BOOLEAN
Comments:	
Description	
This operation first requests (e.g. from the control console of the test system) the test operator to check whether the MS stops the tone generation If the operator informs (e.g. from the control console of the test system) the test system that the MS does not stop tone generation, the operation returns FALSE. If the operator informs the test system that the MS stops tone generation, the operation returns TRUE. Definition of "tone generation": -During MO call: MS generates alerting tone/presents inband information sent by the network. -During MT call: MS generates ringing tone.	
Detailed Comments:	

Test suite parameter declarations

Test Suite Parameter Declarations			
Parameter Name	**Type**	**PICS/PIXIT Ref**	**Comments**
TSPC_24DataF	BOOLEAN	PICS Table A.25 Item 12	2.4 k full rate data mode supported
TSPC_24DataH	BOOLEAN	PICS Table A.25 Item 13	2.4 k half rate data mode supported
TSPC_48DataF	BOOLEAN	PICS Table A.25 Item 14	4.8 k full rate data mode supported
TSPC_48DataH	BOOLEAN	PICS Table A.25 Item 15	4.8 k half rate data mode supported
TSPC_96Data	BOOLEAN	PICS Table A.25 Item 16	9.6 k full rate data mode supported
TSPC_Feat_A51	BOOLEAN	PICS Table A.2 Item 17	ciphering algorithm A5/1 supported
TSPC_Feat_A52	BOOLEAN	PICS Table A.2 Item 18	ciphering algorithm A5/2 supported
TSPC_AddCharSet	BOOLEAN	PICS Table A.25 Item 47	A, B, C, D chars supported
TSPC_AddInfo_PseudoSynch	BOOLEAN	PICS	Pseudo synchronised supported
TSPC_AlertInd	BOOLEAN	PICS Table A.25 Item 49	alerting indication to the user supported
TSPC_SvcOnTCH	BOOLEAN	PICS Table A.25 Item 22	at least one service on traffic channel supported
TSPC_SMS	BOOLEAN	PICS Table A.25 Item 31	at least one short message service supported
TSPC_SS	BOOLEAN	PICS Table A.25 Item 29	at least one supplementary service supported
TSPC_AutoAutoMode	BOOLEAN	PICS Table A.25 Item 48	automatically enter automatic selection of PLMN mode supported
TSPC_BasCharSet	BOOLEAN	PICS Table A.25 Item 46	Chars 0-9, *, # supported
TSPC_CalledNumDisp	BOOLEAN	PICS Table A.2 Item 1	called number display supported
TSPC_DCS	BOOLEAN	PICS Table A.1 Item 3	testing DCS1800
TSPC_DCS_Class3	BOOLEAN	PICS	testing DCS1800 class 3
TSPC_DetachOnPwrDn	BOOLEAN	PICS Table A.25 Item 38	detach on power down supported
TSPC_DetachOnSIMRmv	BOOLEAN	PICS Table A.25 Item 39	detach on SIM remove supported
TSPC_DispRcvSMS	BOOLEAN	PICS Table A.25 Item 34	display of received SMS supported
TSPC_DualRate	BOOLEAN	PICS Table A.25 Item 23	dual rate channel types supported
TSPC_EGSM	BOOLEAN	PICS Table A.1 Item 2	both standard and extended GSM band supported
TSPC_EmgOnly	BOOLEAN	PICS Table A.25 Item 27	the only circuit switched basic service is emergency call
TSPC_followOnReq	BOOLEAN	PICS Table A.25 Item 53	follow-on request procedure supported
TSPC_HalfRateData	BOOLEAN	PICS Table A.25 Item 6	half rate data supported
TSPC_HalfRateSpeech	BOOLEAN	PICS Table A.25 Item 3	half rate speech mode supported
TSPC_InCallMod	BOOLEAN	PICS Table A.25 Item 52	In-Call modification supported
TSPC_NoimmConn	BOOLEAN	PICS Table A.25 Item 51	at least one service not support immediate connection
TSPC_Feat_FND	BOOLEAN	PICS Table A.2 Item 21	FND feature supported
TSPC_NonCallSS	BOOLEAN	PICS Table A.25 Item 30	non call related supplementary service supported
TSPC_MTsvc	BOOLEAN	PICS Table A.25 Item 19	at least one MT circuit switched basic service supported
TSPC_MOsvc	BOOLEAN	PICS Table A.25 Item 20	at least one MO circuit switched basic service supported
TSPC_PGSM	BOOLEAN	PICS Table A.1 Item 1	only standard GSM band supported
TSPC_RefusalCall	BOOLEAN	PICS Table A.25 Item 54	refusal of call supported
TSPC_ReplaceSMS	BOOLEAN	PICS Table A.25 Item 33	replace SMS supported
TSPC_ReplyProc	BOOLEAN	PICS Table A.25 Item 32	(SMS) reply procedures supported
TSPC_RFAmp	BOOLEAN	PICS Table A.25 Item 55	RF amplification supported
TSPC_SDCCHOnly	BOOLEAN	PICS Table A.25 Item 21	only SDCCH supported
TSPC_Serv_SS_AoCC	BOOLEAN	PICS Table A.5 Item 14	AOCC SS supported
TSPC_Serv_SS_BAIC	BOOLEAN	PICS Table A.5 Item 18	BAIC SS supported

TSPC_Serv_SS_BI	BOOLEAN	PICS	BI SS supported
TSPC_Serv_SS_BICRoam	BOOLEAN	PICS Table A.5 Item 19	BICRoam SS supported
TSPC_Serv_SS_BAOC	BOOLEAN	PICS Table A.5 Item 15	BAOC SS supported
TSPC_Serv_SS_BOIC	BOOLEAN	PICS Table A.5 Item 16	BOIC SS supported
TSPC_Serv_SS_BOICexHC	BOOLEAN	PICS Table A.5 Item 17	BOICExHC SS supported
TSPC_Serv_SS_CFB	BOOLEAN	PICS Table A.5 Item 6	CFB supported
TSPC_Serv_SS_CFNRy	BOOLEAN	PICS Table A.5 Item 7	CFNry supported
TSPC_Serv_SS_CFNRc	BOOLEAN	PICS Table A.5 Item 8	CFNRc supported
TSPC_Serv_SS_CFU	BOOLEAN	PICS Table A.5 Item 5	CFU supported
TSPC_Serv_SS_HOLD	BOOLEAN	PICS Table A.5 Item 10	Call Hold SS supported
TSPC_Serv_SS_MPTY	BOOLEAN	PICS Table A.5 Item 11	multiparty SS supported
TSPC_Serv_SS_unstruct	BOOLEAN	PICS	USSD supported
TSPC_Serv_TS11	BOOLEAN	PICS Table A.3 Item 1	telephony supported
TSPC_Serv_TS12	BOOLEAN	PICS Table A.3 Item 2	emergency call supported
TSPC_Serv_TS21	BOOLEAN	PICS Table A.3 Item 3	SMS MT/PP supported
TSPC_Serv_TS22	BOOLEAN	PICS Table A.3 Item 4	SMS MO/PP supported
TSPC_Serv_TS23	BOOLEAN	PICS Table A.3 Item 5	SMS cell broadcast supported
TSPC_Serv_TS61	BOOLEAN	PICS	alternate speech and G3 fax (TS61) supported
TSPC_Serv_TS62	BOOLEAN	PICS	automatic G3 fax (TS62) supported
TSPC_Serv_BS21	BOOLEAN	PICS Table A.4 Item 1	data circuit duplex async 300 bit/s supported
TSPC_Serv_BS22	BOOLEAN	PICS Table A.4 Item 2	data circuit duplex async 1200 bit/s supported
TSPC_Serv_BS23	BOOLEAN	PICS Table A.4 Item 3	data circuit duplex async 1200/75 bit/s supported
TSPC_Serv_BS24	BOOLEAN	PICS Table A.4 Item 4	data circuit duplex async 2400 bit/s supported
TSPC_Serv_BS25	BOOLEAN	PICS Table A.4 Item 5	data circuit duplex async 4800 bit/s supported
TSPC_Serv_BS26	BOOLEAN	PICS Table A.4 Item 6	data circuit duplex async 9600 bit/s supported
TSPC_Serv_BS31	BOOLEAN	PICS Table A.4 Item 7	data circuit duplex sync 1200 bit/s supported
TSPC_Serv_BS32	BOOLEAN	PICS Table A.4 Item 8	data circuit duplex sync 2400 bit/s supported
TSPC_Serv_BS33	BOOLEAN	PICS Table A.4 Item 9	data circuit duplex sync 4800 bit/s supported
TSPC_Serv_BS34	BOOLEAN	PICS Table A.4 Item 10	data circuit duplex sync 9600 bit/s supported
TSPC_Serv_BS41	BOOLEAN	PICS Table A.4 Item 11	PAD access 300 bit/s supported
TSPC_Serv_BS42	BOOLEAN	PICS Table A.4 Item 12	PAD access 1200 bit/s supported
TSPC_Serv_BS43	BOOLEAN	PICS Table A.4 Item 13	PAD access 1200/75 bit/s supported
TSPC_Serv_BS44	BOOLEAN	PICS Table A.4 Item 14	PAD access 2400 bit/s supported
TSPC_Serv_BS45	BOOLEAN	PICS Table A.4 Item 15	PAD access 4800 bit/s supported
TSPC_Serv_BS46	BOOLEAN	PICS Table A.4 Item 16	PAD access 9600 bit/s supported
TSPC_Serv_BS51	BOOLEAN	PICS Table A.4 Item 17	packet access 2400 bit/s supported
TSPC_Serv_BS52	BOOLEAN	PICS Table A.4 Item 18	packet access 4800 bit/s supported
TSPC_Serv_BS53	BOOLEAN	PICS Table A.4 Item 19	packet access 9600 bit/s supported
TSPC_Serv_BS61	BOOLEAN	PICS	alternate speech/data (BS61) supported
TSPC_Serv_BS81	BOOLEAN	PICS	speech followed data (BS81) supported
TSPC_SMSStatusRepCap	BOOLEAN	PICS Table A.25 Item 35	SMS status report capabilities supported
TSPC_StoreRcvSMSME	BOOLEAN	PICS Table A.25 Item 37	Storage of received SMS in ME supported
TSPC_StoreRcvSMSSIM	BOOLEAN	PICS Table A.25 Item 36	Storage of received SMS in SIM supported
TSPC_SIMRmv	BOOLEAN	PICS Table A.25 Item 40	SIM removable without power down supported
TSPC_SwitchOnOff	BOOLEAN	PICS Table A.2 Item 15	switch on/off supported
TSPC_TranspDataOnly	BOOLEAN	PICS Table A.25 Item 9	only transparent data service supported

TSPC_CC	BOOLEAN	PICS Table A.25 Item 26	CC protocol for at least one BC supported
TSPC_TeleSvc	BOOLEAN	PICS Table A.25 Item 25	at least one teleservice supported
TSPC_EFR	BOOLEAN		Set to TRUE for GSM-900 Phase 2+ (EFR) MS which supports any EFR features, and FALSE otherwise.
TSPC_EFR_Speech_v2	BOOLEAN		Set to TRUE for GSM-900 Phase 2+ (EFR) MS which supports Speech Version 2 and FALSE otherwise.
TSPC_EFR_Speech_v3	BOOLEAN		Set to TRUE for GSM-900 Phase 2+ (EFR) MS which supports Speech Version 3 and FALSE otherwise.
TSPC_EFR_EmgCallBcap	BOOLEAN		Set to TRUE if received ESETUP msg contains bearer capability IE, otherwise FALSE.
TSPC_FullRateSpeech	BOOLEAN	PICS Table A.25 Item 2	full rate speech mode supported
TSPC_FullRateOnly	BOOLEAN	PICS Table A.25 Item 24	only full rate channel type supported
TSPX_AltNb	BOOLEAN	PIXIT	use alternative neighbour cells description. default : FALSE
TSPX_TE_stopbit	B_1	PIXIT	TE conf. number stop bits. default value : '0'B, 1 bit
TSPX_TE_databit	B_1	PIXIT	TE conf. number data bits. default value : '1'B, 8 bits
TSPX_TE_FLCT	FLWCNTL	PIXIT	TE flow control. default : 0, outband flow control
TSPX_TE_parity	B_3	PIXIT	TE conf. parity. default : '001'B, no parity
TSPX_BS_21_itc1	B_3	PIXIT	itc value supported for BS21. default value : '001'B, UDI
TSPX_BS_21_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS21. default value : FALSE, only one itc
TSPX_BS_21_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS21. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_21_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_21_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent mode
TSPX_BS_21_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_21_itc2	B_3	PIXIT	other itc value supported by BS21, it shall be differenet from the previous one if the answer to TSPX_BS_21_more_itc is TRUE. default value :'001'B, UDI
TSPX_BS_22_itc1	B_3	PIXIT	itc value supported by BS22. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_22_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS22. default value : FALSE, only one itc
TSPX_BS_22_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS22. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_22_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_22_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent
TSPX_BS_22_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450

TSPX_BS_22_itc2	B_3	PIXIT	other itc value supported by BS22, it shall be differenet from the previous one if the answer to TSPX_BS_22_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_23_itc	B_3	PIXIT	itc value supported by BS23. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_23_T_NT	BOOLEAN	PIXIT	both T and NT supported for BS23. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_23_ce	B_2	PIXIT	ce value used in MO call. default value : '00'B, transparent
TSPX_BS_23_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_24_itc1	B_3	PIXIT	itc value supported by BS24. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_24_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS24. default value : FALSE, only one itc
TSPX_BS_24_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS24. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_24_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_24_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_24_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_24_itc2	B_3	PIXIT	other itc value supported by BS24, it shall be differenet from the previous one if the answer to TSPX_BS_24_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_25_itc1	B_3	PIXIT	itc value supported by BS25. default value : '001'B, UDI
TSPX_BS_25_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS25. default value : FALSE, only one itc
TSPX_BS_25_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS25. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_25_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_25_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_25_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_25_itc2	B_3	PIXIT	other itc value supported by BS25, it shall be differenet from the previous one if the answer to TSPX_BS_25_more_itc is TRUE. default value : '001'B, UDI
TSPX_BS_26_itc1	B_3	PIXIT	itc value supported by BS26. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_26_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS26. default value : FALSE, only one itc
TSPX_BS_26_T_NT	BOOLEAN	PIXIT	both T and NT supported by BS26. default value : FALSE, not support both transparent and non

			transparent modes
TSPX_BS_26_ce	B_2	PIXIT	ce value used in MO call and if the answer to TSPX_BS_26_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_26_sacp	B_3	PIXIT	sacp value used in MO call. default value : '001'B, I440/450
TSPX_BS_26_itc2	B_3	PIXIT	other itc value supported by BS26, it shall be differenet from the previous one if the answer to TSPX_BS_26_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_31_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS31. default value : FALSE, only one itc
TSPX_BS_31_more_sacp	BOOLEAN	PIXIT	more than one sacp value supported by BS31. default value : FALSE, only one sacp
TSPX_BS_31_itc1	B_3	PIXIT	itc value supported by BS31. default value : '001'B, UDI
TSPX_BS_31_sacp1	B_3	PIXIT	sacp value supported by BS31. default value : '001'B, I440/450
TSPX_BS_31_itc2	B_3	PIXIT	other itc value supported by BS31, it shall be differenet from the previous one if the answer to TSPX_BS_31_more_itc is TRUE. default value : '001'B, UDI
TSPX_BS_31_sacp2	B_3	PIXIT	other sacp value supported by BS31, it shall be differenet from the previous one if the answer to TSPX_BS_31_more_sacp is TRUE. default value : '001'B, I440/450
TSPX_BS_32_itc1	B_3	PIXIT	itc value supported by BS32. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_32_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS32. default value : FALSE, only one itc
TSPX_BS_32_more_sacp	BOOLEAN	PIXIT	more than one sacp value supported by BS32. default value : FALSE, only one sacp
TSPX_BS_32_sacp1	B_3	PIXIT	sacp value supported by BS32. default value : '110'B, X.32
TSPX_BS_32_X32_T_NT	BOOLEAN	PIXIT	BS 32 both T and NT supported for X32 3.1kHz. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_32_X32_ce	B_2	PIXIT	if the answer to TSPX_BS_32_X32_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent
TSPX_BS_32_itc2	B_3	PIXIT	other itc value supported by BS32, it shall be differenet from the previous one if the answer to TSPX_BS_32_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_32_sacp2	B_3	PIXIT	other sacp value supported by BS32, it shall be differenet from the previous one if the answer to TSPX_BS_32_more_sacp is TRUE. default value : '110'B, X.32
TSPX_BS_33_itc1	B_3	PIXIT	itc value supported by BS33. default value : '001'B, UDI
TSPX_BS_33_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS33. default value : FALSE,

			only one itc
TSPX_BS_33_more_sacp	BOOLEAN	PIXIT	more than one sacp value supported by BS33. default value : FALSE, only one sacp
TSPX_BS_33_sacp1	B_3	PIXIT	sacp value supported by BS33. default value : '010'B, X.21
TSPX_BS_33_X32_T_NT	BOOLEAN	PIXIT	BS 33 both T and NT supported for X32 3.1kHz. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_33_X32_ce	B_2	PIXIT	if the answer to TSPX_BS_33_X32_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_33_itc2	B_3	PIXIT	other itc value supported by BS32, it shall be differenet from the previous one if the answer to TSPX_BS_33_more_itc is TRUE. default value : '001'B, UDI
TSPX_BS_33_sacp2	B_3	PIXIT	other sacp value supported by BS33, it shall be differenet from the previous one if the answer to TSPX_BS_33_more_sacp is TRUE. default value : '010'B, X.21
TSPX_BS_34_itc1	B_3	PIXIT	itc value supported by BS34. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_34_more_itc	BOOLEAN	PIXIT	more than one itc value supported by BS34. default value : FALSE, only one itc
TSPX_BS_34_more_sacp	BOOLEAN	PIXIT	more than one sacp value supported by BS34. default value : FALSE, only one sacp
TSPX_BS_34_sacp1	B_3	PIXIT	sacp value supported by BS34. default value : '001'B, I440/450
TSPX_BS_34_X32_T_NT	BOOLEAN	PIXIT	BS 34 both T and NT supported for X32 3.1kHz. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_34_X32_ce	B_2	PIXIT	if the answer to TSPX_BS_34_X32_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_34_itc2	B_3	PIXIT	other itc value supported by BS34, it shall be differenet from the previous one if the answer to TSPX_BS_34_more_itc is TRUE. default value : '010'B, 3.1kHz audio, exPLMN
TSPX_BS_34_sacp2	B_3	PIXIT	other sacp value supported by BS34, it shall be differenet from the previous one if the answer to TSPX_BS_34_more_sacp is TRUE. default value : '001'B, I440/450
TSPX_BS_41_T_NT	BOOLEAN	PIXIT	BS41 both T and NT supported . default value : FALSE, not support both
TSPX_BS_41_ce	B_2	PIXIT	ce value used in BS41 MO call. default value : '00'B, transparent
TSPX_BS_42_T_NT	BOOLEAN	PIXIT	BS42 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_42_ce	B_2	PIXIT	ce value used in BS42 MO call. default value : '010'B, non transparent
TSPX_BS_43_T_NT	BOOLEAN	PIXIT	BS43 both T and NT supported .

			default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_43_ce	B_2	PIXIT	ce value used in BS43 MO call. default value : '00'B, transparent
TSPX_BS_44_T_NT	BOOLEAN	PIXIT	BS44 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_44_ce	B_2	PIXIT	ce value used in BS44 MO call. default value : '00'B, transparent
TSPX_BS_45_T_NT	BOOLEAN	PIXIT	BS45 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_45_ce	B_2	PIXIT	ce value used in BS45 MO call. default value : '01'B, non transparent
TSPX_BS_46_T_NT	BOOLEAN	PIXIT	BS46 both T and NT supported . default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_46_ce	B_2	PIXIT	ce value used in BS46 MO call. default value : '00'B, transparent
TSPX_BS_61_S	BOOLEAN	PIXIT	synchronous data supported by BS 61 . default value : TRUE, synchronous mode
TSPX_BS_61_S_more_ur	BOOLEAN	PIXIT	more than one user rate for synchronous data service supported by BS61. default value : FALSE, only ine user rate
TSPX_BS_61_S_ur1	B_4	PIXIT	usrer rate value supported for synchronous data service of BS61. default value : '0100'B, 4.8 kbit/s
TSPX_BS_61_S_ur2	B_4	PIXIT	another usrer rate value supported for synchronous data service of BS61. it shall be different from the previous one if the answer to TSPX_BS_61_S_more_ur is TRUE. default value : '0100'B, 4.8 kbit/s
TSPX_BS_61_A	BOOLEAN	PIXIT	asynchronous data service supported by BS61 . default value : TRUE, asynchronous
TSPX_BS_61_A_ur1	B_4	PIXIT	usrer rate value supported for asynchronous data service of BS61. default value : '0101'B, 9.6 kbit/s
TSPX_BS_61_A_ur1_T_NT	BOOLEAN	PIXIT	both T and NT supported for the rate TSPX_BS_61_A_ur1 of BS61. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_61_A_ur1_ce	B_2	PIXIT	if the answer to TSPX_BS_61_A_ur1_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_BS_61_A_more_ur	BOOLEAN	PIXIT	more than one user rate for asynchronous data service supported by BS61. default value : FALSE, only one user rate
TSPX_BS_61_A_ur2	B_4	PIXIT	another usrer rate value supported for asynchronous data service of BS61. it shall be different from the previous one if the answer to TSPX_BS_61_A_more_ur is TRUE. default value : '0101'B, 9.6 kbit/s
TSPX_BS_61_A_ur2_T_NT	BOOLEAN	PIXIT	both T and NT supported for the rate TSPX_BS_61_A_ur2 of

			BS61. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_81_S	BOOLEAN	PIXIT	synchronous data supported by BS 81. default value : TRUE, synchronous mode
TSPX_BS_81_S_more_ur	BOOLEAN	PIXIT	more than one user rate for synchronous data service supported by BS81. default value : FALSE, only one user rate
TSPX_BS_81_S_ur1	B_4	PIXIT	usrer rate value supported for synchronous data service of BS81. default value : '0101'B, 9.6 kbit/s
TSPX_BS_81_S_ur2	B_4	PIXIT	another usrer rate value supported for synchronous data service of BS81. it shall be different from the previous one if the answer to TSPX_BS_81_S_more_ur is TRUE. default value : '0101'B, 9.6 kbit/s
TSPX_BS_81_A	BOOLEAN	PIXIT	asynchronous data service supported by BS81. default value : TRUE, asynchronous mode
TSPX_BS_81_A_ur1	B_4	PIXIT	usrer rate value supported for asynchronous data service of BS81. default value : '0100'B, 4.8 kbit/s
TSPX_BS_81_A_ur1_T_NT	BOOLEAN	PIXIT	both T and NT supported for the rate TSPX_BS_81_A_ur1 of BS81. default value : FALSE, not support both transparent and non transparent modes
TSPX_BS_81_A_ur1_ce	B_2	PIXIT	if the answer to TSPX_BS_81_A_ur1_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent
TSPX_BS_81_A_more_ur	BOOLEAN	PIXIT	more than one user rate for asynchronous data service supported by BS81. default value : FALSE, only one user rate
TSPX_BS_81_A_ur2	B_4	PIXIT	another usrer rate value supported for asynchronous data service of BS81. it shall be different from the previous one if the answer to TSPX_BS_81_A_more_ur is TRUE. default value : '0100'B, 4.8 kbit/s
TSPX_BS_81_A_ur2_T_NT	BOOLEAN	PIXIT	both T and NT supported for the rate TSPX_BS_81_A_ur2 of BS81. default value : FALSE, not support both transparent and non transparent modes
TSPX_TS_61_T_NT	BOOLEAN	PIXIT	both T and NT supported for TS61(fax). default value : FALSE, not support both transparent and non transparent modes
TSPX_TS_61_ce	B_2	PIXIT	if the answer to TSPX_TS_61_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '01'B, non transparent
TSPX_TS_61_ur1	B_4	PIXIT	user rate for TS61(fax). default value : '0101'B, 9.6 kbit/s
TSPX_TS_61_more_ur	BOOLEAN	PIXIT	more than one user rate supported for TS61. default value : FALSE, only one user rate
TSPX_TS_61_ur2	B_4	PIXIT	another user rate for TS61, it shall be different from the previous one if the answer to

			TSPX_TS_61_more_ur is TRUE. default value : '0101'B, 9.6 kbit/s
TSPX_TS_62_T_NT	BOOLEAN	PIXIT	both T and NT supported for TS62. default value : FALSE, not support both transparent and non transparent modes
TSPX_TS_62_ce	B_2	PIXIT	if the answer to TSPX_TS_62_T_NT is FALSE, the manufacturer must precise which mode is supported by the MS. default value : '00'B, transparent
TSPX_TS_62_ur1	B_4	PIXIT	user rate for TS62. default value : '0100'B, 4.8 kbit/s
TSPX_TS_62_more_ur	BOOLEAN	PIXIT	more than one user rate supported by TS62. default value : FALSE, only one user rate
TSPX_TS_62_ur2	B_4	PIXIT	another user rate for TS62, it shall be different from the previous one if the answer to TSPX_TS_62_more_ur is TRUE. default value : '0100'B, 4.8 kbit/s
TSPX_DTMF	B_1	PIXIT	call control capabilities: value '1'B means MS supports DTMF
TSPX_modF	CHMOD_VAL	PIXIT	any non signalling full rate channel mode for TC_26_6_4_1
TSPX_modH	CHMOD_VAL	PIXIT	any non signalling half channel mode for TC_26_6_4_1
TSPX_anymod	CHMOD_VAL	PIXIT	channel mode for TC_62_10_2_4_2. Any supported value except signalling and Full rate speech. possible values are '00000011'B' 12k data, '00001011'B 6k data, '0010011'B 3.6k data
TSPX_CKSNA	CKSN	PIXIT	cipher key sequence number
TSPX_CKSNB	CKSN	PIXIT	cipher key sequence number
TSPX_CKSNC	CKSN	PIXIT	cipher key sequence number
TSPX_CKSNDef	CKSN	PIXIT	default cipher key sequence number
TSPX_RfPwrCap	B_3	PIXIT	RF power capability
TSPX_RevLevel	B_2	PIXIT	Revision level of classmark 1
TSPX_CiphAlgA5_1	B_1	PIXIT	A5/1 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_2	B_1	PIXIT	A5/2 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_3	B_1	PIXIT	A5/3 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_4	B_1	PIXIT	A5/4 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_5	B_1	PIXIT	A5/5 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_6	B_1	PIXIT	A5/6 algorithm: '1'B : available, '0'B: not available
TSPX_CiphAlgA5_7	B_1	PIXIT	A5/7 algorithm: '1'B : available, '0'B: not available
TSPX_pSyncCap	B_1	PIXIT	pseudo synchronisation capability, '0'B: not present '1'B: present
TSPX_SSscrnInd	B_2	PIXIT	SS screen indicator
TSPX_SMCap	B_1	PXIIT	short message capability, '0'B: not present '1'B: present
TSPX_frqCap	B_1	PIXIT	frequency capability. DCS: reserved, GSM: '0'B not support extension band, '1'B support extension band
TSPX_ClsMk3	B_1	PIXIT	classmark 3 indicator, '0'B no addition capability info, '1'B additional capability info in class mark 3
TSPX_RfPwrCapAmp	B_3	PIXIT	RF power capability with external RF amplifier

TSPX_CphAlgA	CPHALG	PIXIT	any supported ciphering algorithm
TSPX_CphAlgB	CPHALG	PIXIT	ciphering algorithm supported
TSPX_CphAlgC	CPHALG	PIXIT	ciphering algorithm supported
TSPX_CphAlgD	CPHALG	PIXIT	ciphering algorithm supported
TSPX_CphAlgE	CPHALG	PIXIT	ciphering algorithm supported
TSPX_CphAlgDef	CPHALG	PIXIT	default ciphering algorithm
TSPX_DTMFInd	BOOLEAN	PIXIT	DTMF indication to user supported
TSPX_NoOfHoAccA	INTEGER	PIXIT	number of handover access bursts, value range 10 - 20
TSPX_NoOfHoAccB	INTEGER	PIXIT	number of handover access bursts, value range 10 - 20
TSPX_NoOfHoAccC	INTEGER	PIXIT	number of handover access bursts, value range 10 - 20
TSPX_NoOfHoAccD	INTEGER	PIXIT	number of handover access bursts, value range 5 - 10
TSPX_NoOfHoAccE	INTEGER	PIXIT	number of handover access bursts, value range 5 - 10
TSPX_NoOfHoAccF	INTEGER	PIXIT	number of handover access bursts, value range 5 - 10
TSPX_NoOfHoAccG	INTEGER	PIXIT	number of handover access bursts, value range 2 - 5
TSPX_NoOfHoAccH	INTEGER	PIXIT	number of handover access bursts, value range 2 - 5
TSPX_NoOfHoAccI	INTEGER	PIXIT	number of handover access bursts, value range 2 - 5
TSPX_HoRefA	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefB	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefC	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefD	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefE	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefF	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefG	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefH	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefI	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HoRefJ	HORF	PIXIT	Hand over reference, GSM 04.08, 10.5.2.15
TSPX_HSN	INTEGER	PIXIT	Hopping sequence number value range: 0 - 63
TSPX_IMSI	HEXSTRING	PIXIT	IMSI of the MS
TSPX_IMEI	HEXSTRING	PIXIT	IMEI of the MS.
TSPX_IMEISV	HEXSTRING	PIXIT	IMEISV of the MS. Used in: TC_26_6_8_5, TC_26_7_3_1
TSPX_Ki	BITSTRING	PIXIT	default authentication key used in testing
TSPX_MAIO	INTEGER	PIXIT	mobile allocation index offset, value range: 0 - 63
TSPX_MaxRetrans	INTEGER	PIXIT	Max-Retrans
TSPX_MSTxpwrMax	MAXTXPOW	PIXIT	maximum output power from MS
TSPX_PwrlvlA	INTEGER	PIXIT	power level, value between 2 to 15
TSPX_PwrlvlB	INTEGER	PIXIT	power level, value between 2 to 15
TSPX_PwrlvlC	INTEGER	PIXIT	power level, value between 2 to 15
TSPX_PwrlvlD	INTEGER	PIXIT	power level, value between 2 to 15
TSPX_RANDA	RAND	PIXIT	challenge RAND

TSPX_RANDB	RAND	PIXIT	challenge RAND
TSPX_RANDC	RAND	PIXIT	challenge RAND
TSPX_RANDDef	RAND	PIXIT	default challenge RAND
TSPX_SDCCH4SubA	B_2	PIXIT	TDMA offset of SDCCH/4 subchannel
TSPX_SDCCH4SubB	B_2	PIXIT	TDMA offset of SDCCH/4 subchannel
TSPX_SDCCH4SubC	B_2	PIXIT	TDMA offset of SDCCH/4 subchannel
TSPX_SDCCH4SubDef	B_2	PIXIT	TDMA offset of default SDCCH/4 subchannel
TSPX_SDCCH8SubA	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubB	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubC	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubD	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubE	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubF	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubG	B_3	PIXIT	TDMA offset of SDCCH/8 subchannel
TSPX_SDCCH8SubDef	B_3	PIXIT	TDMA offset of default SDCCH/8 subchannel
TSPX_TC1M	INTEGER	PIXIT	timer value for GSM timer TC1M (for SMS)
TSPX_MaxCPDataRetx	INTEGER	PIXIT	max. number of CP data retransmissions for SMS
TSPX_Immconn	BOOLEAN	PIXIT	Immediate connect supported
TSPX_TimadvA	INTEGER	PIXIT	timing advance
TSPX_TimadvB	INTEGER	PIXIT	timing advance
TSPX_TimadvC	INTEGER	PIXIT	timing advance
TSPX_TmSltA	SN	PIXIT	time slot
TSPX_TmSltB	SN	PIXIT	time slot
TSPX_TmSltC	SN	PIXIT	time slot
TSPX_TmSltD	SN	PIXIT	time slot
TSPX_TmSltE	SN	PIXIT	time slot
TSPX_TmSltF	SN	PIXIT	time slot
TSPX_TmSltG	SN	PIXIT	time slot
TSPX_TmSltDef	SN	PIXIT	default time slot
TSPX_TmSltNotZero	SN	PIXIT	time slot, arbitrarily value, but not zero.
TSPX_TmSltNotZero1	SN	PIXIT	time slot, arbitrarily value, but not zero and not TSPX_TmSltNotZero
TSPX_Txint	INTEGER	PIXIT	Tx-Integer
TSPX_TscA	TSC	PIXIT	training sequence code
TSPX_TscB	TSC	PIXIT	training sequence code
TSPX_TscC	TSC	PIXIT	training sequence code
TSPX_TscD	TSC	PIXIT	training sequence code
TSPX_TscE	TSC	PIXIT	training sequence code
TSPX_TscF	TSC	PIXIT	training sequence code
TSPX_TscG	TSC	PIXIT	training sequence code
TSPX_TscDef	TSC	PIXIT	default TSC
TSPX_T3122	WI	PIXIT	value of timer T3122, HEXSTRING[2]
TSPX_TCHcarrierA_ho	INTEGER	PIXIT	the value can be choosen arbitrarily from cell allocation of cell A (GSM), but not BCCH carrier .possible value is : 10, 17, 26, 34, 42, 45, 46, 52, 59, 66, 73, 74, 75, 76, 108, 114
TSPX_TCHcarrierA_hod	INTEGER	PIXIT	the value can be choosen arbitrarily from cell allocation of cell A (DCS), but not BCCH carrier . possible value is :

			734,741,754,759,762,766,767,773,775,779,782,791,798,829,832,844
TSPX_TCHcarrierB	INTEGER	PIXIT	TCH and SDCCH channel frequency number of cell B (GSM)
TSPX_TCHcarrierBd	INTEGER	PIXIT	TCH and SDCCH channel frequency number of cell B (DCS)
TSPX_TCHcarrierB_ho	INTEGER	PIXIT	the value can be choosen arbitrarily from cell allocation of cell B (GSM), but not BCCH carrier . possibles value is : 14,18,22,24,30,31,38,60,66,73,74,75,76,108,114
TSPX_TCHcarrierB_hod	INTEGER	PIXIT	not BCCH carrier of cell B. the value can be choosen arbitrarily from cell allocation of cell B (DCS), which is (739,743,746,749,756,758,761,771,779,782,791,798,829,832,844)
TSPX_TCHcarrierB2_ho	INTEGER	PIXIT	Choosen arbitrarly from cell allocation B for GSM HO cases, but not BCCH carrier!
TSPX_TCHcarrierB2_hod	INTEGER	PIXIT	Choosen arbitrarly from cell allocation B for DCS HO cases, but not BCCH carrier!
TSPX_TCHHSubA	B_1	PIXIT	TDMA offset of half rate subchannel
TSPX_TCHHSubDef	B_1	PIXIT	TDMA offset of default half rate subchannel
TSPX_TMSI	TMSI_V	PIXIT	TMSI of the MS used in test
TSPX_TMSI1	TMSI_V	PIXIT	another TMSI used in test which shall differ from TSPX_TMSI, TSPX_TMSI + '01'O TSPX_TMSI + '02'O TSPX_TMSI + '03'O
TSPX_Uupd	B_8	PIXIT	user-user protocol discriminator
TSPX_UuInfo	OCTETSTRING	PIXIT	user-user information
TSPX_k	INTEGER	PIXIT	timing of cell A before cell B k bit periods for TC_26_6_5_1, TC_26_6_5_2_? TC_26_6_5_3_?, TC_26_6_5_4_?.
TSPX_y	INTEGER	PIXIT	timing advance for TC_26_6_5_3_?, TC_26_6_5_4_?, TC_26_6_5_5_? . The values of TSPX_k (or TSPX_k1) and TSPX_y: 0 < (2*TSPX_k + TSPX_y) MOD 256 < 60.
TSPX_k1	INTEGER	PIXIT	timing of cell A before cell B k bit periods for TC_26_6_5_5_?
TSPX_k2	INTEGER	PIXIT	timing of cell A before cell B k bit periods for TC_26_6_5_6
TSPX_y2	INTEGER	PIXIT	timing advance for TC_26_6_5_6, value range : 11 - 62.
TSPX_k3	INTEGER	PIXIT	timing of cell A before cell B k bit periods for TC_26_6_5_7
TSPX_y3	INTEGER	PIXIT	timing advance for TC_26_6_5_7
TSPX_nPara	INTEGER	PIXIT	the n'th ChReq for TC_26_6_1_2, shall be choose to [1..8], MAXRETRANS for TC_26_6_1_2 is 7 (See 11.10 for more explaination).
TSPX_kPara	INTEGER	PIXIT	the k'th ChReq for TC_26_6_1_2
TSPX_rPara	INTEGER	PIXIT	the r'th ChReq for TC_26_6_1_2
TSPX_i1Para	INTEGER	PIXIT	for TC_26_1_2
TSPX_i2Para	INTEGER	PIXIT	for TC_26_1_2
TSPX_i3Para	INTEGER	PIXIT	the r'th ChReq for TC_26_6_1_2
TSPX_n1Para	INTEGER	PIXIT	the n'th ChReq for TC_26_6_1_3
TSPX_i4Para	INTEGER	PIXIT	for TC_26_6_1_3
TSPX_xPara	INTEGER	PIXIT	t3122 for TC_26_6_1_3
TSPX_AGBLKS1	INTEGER	PIXIT	BS-AG-BLKS-RES for

			TC_26_6_2_3_1, TC_26_6_2_1_1
TSPX_PAMFRMS1	INTEGER	PIXIT	BS-PA-MFRMS for TC_26_6_2_3_1 (shall not be set to 9), TC_26_6_2_1_1
TSPX_CcchConf1	CCCH_CON	PIXIT	paging subchannel for TC_26_6_2_3_1, TC_26_6_2_1_1
TSPX_PgSubch	INTEGER	PIXIT	paging subchannel for TC_26_6_2_3_1
TSPX_AGBLKS2	INTEGER	PIXIT	BS-AG-BLKS-RES for TC_26_6_2_3_2, TC_26_6_2_1_2
TSPX_PAMFRMS2	INTEGER	PIXIT	BS-PA-MFRMS for TC_26_6_2_3_2, TC_26_6_2_1_2
TSPX_CcchConf2	CCCH_CON	PIXIT	paging subchannel for TC_26_6_2_3_2, TC_26_6_2_1_2
TSPX_AGBLKS3	INTEGER	PIXIT	BS-AG-BLKS-RES for TC_26_6_2_4, TC_26_6_2_2, TC_26_6_2_1_3
TSPX_PAMFRMS3	INTEGER	PIXIT	BS-PA-MFRMS for TC_26_6_2_4, TC_26_6_2_2, TC_26_6_2_1_3
TSPX_CcchConf3	CCCH_CON	PIXIT	paging subchannel for TC_26_6_2_4, TC_26_6_2_2, TC_26_6_2_1_3
TSPX_AGBLKS4	INTEGER	PIXIT	BS-AG-BLKS-RES for TC_26_6_2_5
TSPX_PAMFRMS4	INTEGER	PIXIT	BS-PA-MFRMS for TC_26_6_2_5
TSPX_CcchConf4	CCCH_CON	PIXIT	paging subchannel for TC_26_6_2_5 (shall be in the set ('010', '100', '110'))
TSPX_Chtp1	CH_TDMA	PIXIT	channel type for TC_26_6_13_1, any value supported by the MS
TSPX_ChMod1	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_1, any value for the channel type of TSPX_Chtp1
TSPX_Maio1	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma1, default : 2
TSPX_Ma1	MAC	PIXIT	mobile allocation for TC_26_6_13_1, its value shall indicate number of frequencies between 1 and 17. default : '010101'O
TSPX_Hsn1	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1. default: 1
TSPX_Maio2	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma2, default : 8
TSPX_Ma2	MAC	PIXIT	mobile allocation for TC_26_6_13_1, its value shall indicate number of frequencies between 1 and 17. default :'01362A5'O
TSPX_Hsn2	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, default : 6
TSPX_Maio3	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma3, default : 14
TSPX_Ma3	MAC	PIXIT	mobile allocation for TC_26_6_13_1, its value shall indicate number of frequencies between 1 and 17. default :'01FFFA'O

TSPX_Hsn3	INTEGER	PIXIT	hopping parameter for TC_26_6_13_1, default : 0
TSPX_Chtp2	CH_TDMA	PIXIT	channel type for TC_26_6_13_2, any value supported by the MS
TSPX_ChMod2	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_2
TSPX_Maio4	INTEGER	PIXIT	hopping parameter for TC_26_6_13_2, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma4, default : 3
TSPX_Ma4	MAC	PIXIT	mobile allocation for TC_26_6_13_2, its value shall indicate number of frequencies between 1 and 17. default : '001141'O
TSPX_Hsn4	INTEGER	PIXIT	hopping parameter for TC_26_6_13_2, default : 0
TSPX_Maio5	INTEGER	PIXIT	hopping parameter for TC_26_6_13_2, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma5, default : 9
TSPX_Ma5	MAC	PIXIT	mobile allocation for TC_26_6_13_2, its value shall indicate number of frequencies between 1 and 17. default : '01ABCB'O
TSPX_Hsn5	INTEGER	PIXIT	hopping parameter for TC_26_6_13_2, default : 16
TSPX_Chtp3	CH_TDMA	PIXIT	channel type for TC_26_6_13_3, any value supported by the MS
TSPX_Maio6	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma6, default : 4
TSPX_Ma6	MAC	PIXIT	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 1 and 17. default :'006248'O
TSPX_Hsn6	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, default : 4
TSPX_Maio7	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma7, default : 10
TSPX_Ma7	MAC	PIXIT	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 2 and 17. default : '009AFB'O
TSPX_Chtp4	CH_TDMA	PIXIT	channel type for TC_26_6_13_3, any value supported by the MS
TSPX_Maio8	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma8, default : 1
TSPX_Ma8	MAC	PIXIT	mobile allocation for TC_26_6_13_3, its value shall indicate number of frequencies between 2 and 17. default : '002800'O
TSPX_Hsn8	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, default : 40
TSPX_Maio9	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma9, default : 15
TSPX_Ma9	MAC	PIXIT	mobile allocation for TC_26_6_13_3, its value shall

			indicate number of frequencies between 2 and 17. default : '01FFFB'O
TSPX_Hsn9	INTEGER	PIXIT	hopping parameter for TC_26_6_13_3, default : 8
TSPX_Chtp5	CH_TDMA	PIXIT	SDCCH8 subchannel immediately assigned, TC_26_6_13_4
TSPX_Maio10	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma10, default : 5
TSPX_Ma10	MAC	PIXIT	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 1 and 17. default : '01D082'O
TSPX_Hsn10	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, default: 0
TSPX_Maio11	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma11, default : 11
TSPX_Ma11	MAC	PIXIT	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 2 and 17. default : '00CFF3'O
TSPX_Chtp6	CH_TDMA	PIXIT	channel type of a non- existing channel in Assignment Cmd, TC_26_6_13_4
TSPX_Maio12	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma12, default : 1
TSPX_Ma12	MAC	PIXIT	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 1 and 17. default : '010100'O
TSPX_Hsn12	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, default : 39
TSPX_Maio13	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma13, default : 6
TSPX_Ma13	MAC	PIXIT	mobile allocation for TC_26_6_13_4, its value shall indicate number of frequencies between 1 and 17. default : '00E690'O
TSPX_Hsn13	INTEGER	PIXIT	hopping parameter for TC_26_6_13_4, default: 42
TSPX_Chtp7	CH_TDMA	PIXIT	channel type for TC_26_6_13_5
TSPX_ChMod4	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_5
TSPX_Maio14	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma14, default : 5
TSPX_Ma14	MAC	PIXIT	mobile allocation for TC_26_6_13_5, its value shall indicate number of frequencies between 1 and 17. default : '014A28'O
TSPX_Hsn14	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, default : 50
TSPX_Maio15	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, its value between 0 and (the number of frequencies) -1, which is defined in

			TSPX_Ma15, default : 12
TSPX_Ma15	MAC	PIXIT	mobile allocation for TC_26_6_13_5, its value shall indicate number of frequencies between 1 and 16. default : '00FF79'O
TSPX_Hsn15	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, default : 33
TSPX_Maio16	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma16, default : 2
TSPX_Ma16	MAC	PIXIT	mobile allocation for TC_26_6_13_5, its value shall indicate number of frequencies between 1 and 16. default : '001110'O
TSPX_Hsn16	INTEGER	PIXIT	hopping parameter for TC_26_6_13_5, default : 21
TSPX_Chtp8	CH_TDMA	PIXIT	channel type for TC_26_6_13_6
TSPX_ChMod5	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_6
TSPX_Maio17	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma17, default : 7
TSPX_Ma17	MAC	PIXIT	mobile allocation for TC_26_6_13_6, its value shall indicate number of frequencies between 1 and 17. default : '00E6A1'O
TSPX_Hsn17	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, default : 22
TSPX_Maio18	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma18, default : 13
TSPX_Ma18	MAC	PIXIT	mobile allocation for TC_26_6_13_6, its value shall indicate number of frequencies between 1 and 16. default : '00FFF9'O
TSPX_Hsn18	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, default : 9
TSPX_Maio19	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma19, default : 3
TSPX_Ma19	MAC	PIXIT	mobile allocation for TC_26_6_13_6, its value shall indicate number of frequencies between 1 and 16. default : '001111'O
TSPX_Hsn19	INTEGER	PIXIT	hopping parameter for TC_26_6_13_6, default : 44
TSPX_Chtp9	CH_TDMA	PIXIT	channel type for TC_26_6_13_7
TSPX_ChMod6	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_7
TSPX_Maio20	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma20, default : 8
TSPX_Ma20	MAC	PIXIT	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 17. default : '016699'O
TSPX_Hsn20	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, default : 30
TSPX_Maio21	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, its value between

			0 and (the number of frequencies) -1, which is defined in TSPX_Ma21, default : 14
TSPX_Ma21	MAC	PIXIT	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 17. default : '01FEFB'O
TSPX_Chtp10	CH_TDMA	PIXIT	channel type for TC_26_6_13_7
TSPX_Maio22	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma22, default : 4
TSPX_Ma22	MAC	PIXIT	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 16. default : '006241'O
TSPX_Hsn22	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, default : 11
TSPX_Maio23	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma23, default : 11
TSPX_Ma23	MAC	PIXIT	mobile allocation for TC_26_6_13_7, its value shall indicate number of frequencies between 1 and 16. default : '00FF1B'O
TSPX_Hsn23	INTEGER	PIXIT	hopping parameter for TC_26_6_13_7, default : 60
TSPX_Chtp11	CH_TDMA	PIXIT	SDCCH8 subchannel immediately assigned, TC_26_6_13_8
TSPX_ChMod7	CHMOD_VAL	PIXIT	channel mode for TC_26_6_13_8
TSPX_Maio24	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma24, default : 9
TSPX_Ma24	MAC	PIXIT	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 17. default : '00E6E9'O
TSPX_Hsn24	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, default : 7
TSPX_Maio25	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma25, default : 15
TSPX_Ma25	MAC	PIXIT	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 17. default : '01FFFB'O
TSPX_Chtp12	CH_TDMA	PIXIT	channel type of a non- existing channel in Assignment Cmd, TC_26_6_13_8
TSPX_Maio26	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma26, default : 5
TSPX_Ma26	MAC	PIXIT	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 16. default : '009168'O
TSPX_Hsn26	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, default : 9
TSPX_Maio27	INTEGER	PIXIT	hopping parameter for

			TC_26_6_13_8, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma27, default : 1
TSPX_Ma27	MAC	PIXIT	mobile allocation for TC_26_6_13_8, its value shall indicate number of frequencies between 1 and 16. default : '004080'O
TSPX_Hsn27	INTEGER	PIXIT	hopping parameter for TC_26_6_13_8, default : 38
TSPX_Chtp13	CH_TDMA	PIXIT	channel type for TC_26_6_13_9
TSPX_Tm3	INTEGER	PIXIT	channel mode for TC_26_6_13_9, value between 60 -100
TSPX_Maio28	INTEGER	PIXIT	hopping parameter for TC_26_6_13_9, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma28, default : 10
TSPX_Ma28	MAC	PIXIT	mobile allocation for TC_26_6_13_9, its value shall indicate number of frequencies between 1 and 17. default : '00EE7A'O
TSPX_Hsn28	INTEGER	PIXIT	hopping parameter for TC_26_6_13_9, default : 11
TSPX_Maio29	INTEGER	PIXIT	hopping parameter for TC_26_6_13_9, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma29, default : 6
TSPX_Ma29	MAC	PIXIT	mobile allocation for TC_26_6_13_9, its value shall indicate number of frequencies between 1 and 17. default : '00F070'O
TSPX_Chtp14	CH_TDMA	PIXIT	channel type for TC_26_6_13_10
TSPX_Maio30	INTEGER	PIXIT	hopping parameter for TC_26_6_13_10, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma30, default : 11
TSPX_Ma30	MAC	PIXIT	mobile allocation for TC_26_6_13_10, its value shall indicate number of frequencies between 1 and 17. default : '017F69'O
TSPX_Hsn30	INTEGER	PIXIT	hopping parameter for TC_26_6_13_10, default : 62
TSPX_Maio31	INTEGER	PIXIT	hopping parameter for TC_26_6_13_10, its value between 0 and (the number of frequencies) -1, which is defined in TSPX_Ma31, default : 1
TSPX_Ma31	MAC	PIXIT	mobile allocation for TC_26_6_13_10, its value shall indicate number of frequencies between 1 and 17. default : '000101'O
TSPX_MOBscSvcA	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcB	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcC	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcD	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcE	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcF	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcG	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcH	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcI	SERVICES	PIXIT	any supported MO basic service
TSPX_MOBscSvcJ	SERVICES	PIXIT	any supported MO basic service
TSPX_MT_DualModSvc	SERVICES	PIXIT	any supported MT dual mode service
TSPX_MT_NonSptSvc	SERVICES	PIXIT	any not supported MT basic service

TSPX_MTBscSvcA	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcB	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcC	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcD	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcE	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcF	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcG	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcH	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcI	SERVICES	PIXIT	any supported MT basic service
TSPX_MTBscSvcJ	SERVICES	PIXIT	any supported MT basic service
TSPX_MTNIC_BscSvcA	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcB	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcC	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcD	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcE	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcF	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcG	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcH	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcI	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_MTNIC_BscSvcJ	SERVICES	PIXIT	any supported MT basic service without immediate connection
TSPX_Telephony_Rate	RATE	PIXIT	channel rate for TS11, default value : C_Full
TSPX_MOChRateA	RATE	PIXIT	channel rate for TSPX_MOBscSvcA, default value : C_Full
TSPX_MOChRateB	RATE	PIXIT	channel rate for TSPX_MOBscSvcB, default value : C_Full
TSPX_MOChRateC	RATE	PIXIT	channel rate for TSPX_MOBscSvcC, default value : C_Full
TSPX_MOChRateD	RATE	PIXIT	channel rate for TSPX_MOBscSvcD, default value : C_Full
TSPX_MOChRateE	RATE	PIXIT	channel rate for TSPX_MOBscSvcE, default value : C_Full
TSPX_MOChRateF	RATE	PIXIT	channel rate for TSPX_MOBscSvcF, default value : C_Full
TSPX_MOChRateG	RATE	PIXIT	channel rate for TSPX_MOBscSvcG, default value : C_Full
TSPX_MOChRateH	RATE	PIXIT	channel rate for TSPX_MOBscSvcH, default value : C_Full
TSPX_MOChRateI	RATE	PIXIT	channel rate for TSPX_MOBscSvcI, default value : C_Full
TSPX_MOChRateJ	RATE	PIXIT	channel rate for TSPX_MOBscSvcJ, default value : C_Full
TSPX_MTChRateA	RATE	PIXIT	channel rate for TSPX_MTBscSvcA, default value : C_Full
TSPX_MTChRateB	RATE	PIXIT	channel rate for TSPX_MTBscSvcB, default value : C_Full
TSPX_MTChRateC	RATE	PIXIT	channel rate for TSPX_MTBscSvcC, default value : C_Full

TSPX_MTChRateD	RATE	PIXIT	channel rate for TSPX_MTBscSvcD, default value : C_Full
TSPX_MTChRateE	RATE	PIXIT	channel rate for TSPX_MTBscSvcE, default value : C_Full
TSPX_MTChRateF	RATE	PIXIT	channel rate for TSPX_MTBscSvcF, default value : C_Full
TSPX_MTChRateG	RATE	PIXIT	channel rate for TSPX_MTBscSvcG, default value : C_Full
TSPX_MTChRateH	RATE	PIXIT	channel rate for TSPX_MTBscSvcH, default value : C_Full
TSPX_MTChRateI	RATE	PIXIT	channel rate for TSPX_MTBscSvcI, default value : C_Full
TSPX_MTChRateJ	RATE	PIXIT	channel rate for TSPX_MTBscSvcJ, default value : C_Full
TSPX_MTNIC_ChRateA	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcA, default value : C_Full
TSPX_MTNIC_ChRateB	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcB, default value : C_Full
TSPX_MTNIC_ChRateC	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcC, default value : C_Full
TSPX_MTNIC_ChRateD	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcD, default value : C_Full
TSPX_MTNIC_ChRateE	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcE, default value : C_Full
TSPX_MTNIC_ChRateF	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcF, default value : C_Full
TSPX_MTNIC_ChRateG	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcG, default value : C_Full
TSPX_MTNIC_ChRateH	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcH, default value : C_Full
TSPX_MTNIC_ChRateI	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcI, default value : C_Full
TSPX_MTNIC_ChRateJ	RATE	PIXIT	channel rate for TSPX_MTNIC_BscSvcJ, default value : C_Full
TSPX_EmgCallRate	RATE	PIXIT	Rate for the basic service supported for MO emergency calls
TSPX_MO_NonCallSS	SERVICES	PIXIT	any supported MO non-call releated supplementary service
TSPX_MO_BscSvc_SMS	SERVICES	PIXIT	any supported MO SMS calls
TSPX_MO_DualModSvc	SERVICES	PIXIT	any supported MO dual mode call
TSPX_MO_DualModRate	RATE	PIXIT	Rate for the supported MO dual mode call in TSPX_MO_DualModSvc
TSPX_aTav	INTEGER	PIXIT	averaging period a*Tav in milliseconds
TSPX_bTav	INTEGER	PIXIT	averaging period b*Tav in milliseconds
Detailed Comments:			

Test case selection expression definitions

Test Case Selection Expression Definitions		
Expression Name	**Selection Expression**	**Comments**
SelExpr_0000	TRUE	General test group always selected
SelExpr_0001	TRUE	always selected
SelExpr_0002	TSPC_MTsvc	MT circuit switched basic service supported
SelExpr_0003	NOT TSPC_Serv_SS_AoCC	AOCC not supported
SelExpr_0004	TSPC_Serv_SS_AoCC AND(NOT TSPC_Serv_SS_HOLD)	AOCC supported but Call Hold not supported
SelExpr_0005	TSPC_Serv_SS_AoCC AND TSPC_Serv_SS_HOLD AND(NOT TSPC_Serv_SS_MPTY)	AOCC and Call Hold supported but multiparty not supported
SelExpr_0006	NOT TSPC_Feat_FND	FND feature not supported
SelExpr_0007	TSPC_MOsvc	MO circuit switched basic service supported
SelExpr_0100	TRUE	initial test group always selected
SelExpr_0101	TRUE	always selected
SelExpr_0102	TSPC_SvcOnTCH	at least one service on traffic channel supported
SelExpr_0103	TSPC_DualRate	half rate channel supported
SelExpr_0104	TSPC_Serv_BS21 OR TSPC_Serv_BS22 OR TSPC_Serv_BS23 OR TSPC_Serv_BS24 OR TSPC_Serv_BS25 OR TSPC_Serv_BS26 OR TSPC_Serv_BS31 OR TSPC_Serv_BS32 OR TSPC_Serv_BS33 OR TSPC_Serv_BS34 OR TSPC_Serv_BS41 OR TSPC_Serv_BS42 OR TSPC_Serv_BS43 OR TSPC_Serv_BS44 OR TSPC_Serv_BS45 OR TSPC_Serv_BS46 OR TSPC_Serv_BS51 OR TSPC_Serv_BS52 OR TSPC_Serv_BS53 OR TSPC_Serv_BS61 OR TSPC_Serv_BS81 OR TSPC_Serv_TS61 OR TSPC_Serv_TS62	at least one data service supported
SelExpr_0105	TSPC_NonCallSS	non call related supplementary service supported
SelExpr_0106	TSPC_Serv_TS22	MO short message service supported
SelExpr_0107	TSPC_Serv_TS11 OR TSPC_Serv_TS12	speech supported
SelExpr_0200	TRUE	idleMode test group always selected
SelExpr_0201	TRUE	always selected
SelExpr_0300	TRUE	BiBo test group always selected
SelExpr_0301	TSPC_Serv_TS11 OR TSPC_Serv_TS12 OR SelExpr_0104	CC protocol for at least one BC supported
SelExpr_0302	TRUE	always selected
SelExpr_0400	TRUE	RR test group always selected
SelExpr_0401	TRUE	always selected
SelExpr_0402	SelExpr_0301	CC protocol for at least one BC supported
SelExpr_0403	SelExpr_0301	Full rate traffic channel supported
SelExpr_0404	TSPC_DualRate	Dual rate traffic channel supported
SelExpr_0405	SelExpr_0301 AND(TSPC_Feat_A51 OR TSPC_Feat_A52)	call control protocol and (A5/1 or A5/2) algorithm supported
SelExpr_0406	SelExpr_0301 AND TSPC_RFAmp	call control protocol and RF amplification supported
SelExpr_0407	SelExpr_0301	call control protocol and full rate traffic channel supported
SelExpr_0408	SelExpr_0301 AND TSPC_DualRate	call control protocol and dual rate traffic channel supported
SelExpr_0409	SelExpr_0301 AND TSPC_AddInfo_PseudoSynch	CC protocol for at least one BC supported and Pseudo synchronised supported
SelExpr_0410	TSPC_MOsvc	at least one MO circuit switched basic service supported
SelExpr_0500	TRUE	MM test group always selected
SelExpr_0501	TRUE	always selected
SelExpr_0502	TSPC_SIMRmv	SIM removable without power down supported
SelExpr_0503	TSPC_Serv_TS11 OR TSPC_Serv_TS12	speech supported

SelExpr_0504	TSPC_NonCallSS	The MS supports a non call related supplementary service operation during an active call
SelExpr_0600	TSPC_MOsvc OR TSPC_MTsvc	CC test group
SelExpr_0601	TSPC_MOsvc AND(NOT TSPC_EmgOnly)	at least one mobile originating circuit switched basic service supported and not only emergency call supported
SelExpr_0602	(TSPC_NoimmConn) AND TSPC_MTsvc	immediate connect not supported and at least one mobile terminated circuit switched basic service supported
SelExpr_0603	TSPC_RefusalCall AND TSPC_MTsvc	refusal of call supported and at least one mobile terminated circuit switched basic service supported
SelExpr_0604	TSPC_Serv_TS11 OR TSPC_Serv_TS12 OR TSPC_Serv_TS61 OR TSPC_Serv_BS61 OR TSPC_Serv_BS81	speech supported
SelExpr_0605	SelExpr_0604 AND TSPC_AlertInd	speech & alerting supported
SelExpr_0606	TSPC_MTsvc	at least one mobile terminated circuit switched basic service supported
SelExpr_0607	TSPC_Serv_TS61 OR TSPC_Serv_BS61 OR TSPC_Serv_BS81	dual mode services supported
SelExpr_0700	SelExpr_0301	StrcutureProc test group
SelExpr_0701	TSPC_Serv_TS11 OR TSPC_Serv_TS62	at least one teleservice (except emergency call and dual service) supported
SelExpr_0702	TSPC_Serv_TS12	emergency call supported
SelExpr_0703	TSPC_Serv_TS12 AND TSPC_DualRate	dual rate emergency call supported
SelExpr_0800	TSPC_SS	SS test group
SelExpr_0801	TSPC_Serv_SS_BOIC OR TSPC_Serv_SS_BAIC OR TSPC_Serv_SS_BOICexHC OR TSPC_Serv_SS_BICRoam OR TSPC_Serv_SS_BAOC	at least one of call restrictions supported
SelExpr_0802	TSPC_Serv_SS_AoCC	AOCC supported
SelExpr_0803	TSPC_Serv_SS_BOIC	BOIC supported
SelExpr_0804	TSPC_Serv_SS_BAIC	BAIC supported
SelExpr_0805	TSPC_Serv_SS_BOICexHC	BOICExHC supported
SelExpr_0806	TSPC_Serv_SS_BOICexHC OR TSPC_Serv_SS_BAIC	BOICExHC OR BAIC supported
SelExpr_0807	TSPC_Serv_SS_BOIC OR TSPC_Serv_SS_BICRoam	BOIC OR BICRoam supported
SelExpr_0808	TSPC_Serv_SS_BI	BI supported
SelExpr_0809	TSPC_Serv_SS_CFNRy OR TSPC_Serv_SS_CFU	CFNNRy or CFU supported
SelExpr_0810	TSPC_Serv_SS_CFB OR TSPC_Serv_SS_CFU OR TSPC_Serv_SS_CFNRc OR TSPC_Serv_SS_CFNRy	CFB or CFU or CFNRy or CFNRc supported
SelExpr_0811	TSPC_Serv_SS_CFB OR TSPC_Serv_SS_CFNRc OR TSPC_Serv_SS_CFNRy	CFNRy or CFNRc or CFB supported
SelExpr_0812	TSPC_Serv_SS_CFNRc OR TSPC_Serv_SS_CFB	CFNNRc or CFB supported
SelExpr_0813	TSPC_Serv_SS_CFB	CFB supported
SelExpr_0814	TSPC_Serv_SS_BICRoam OR TSPC_Serv_SS_BAOC	BAOC or BICRoam supported
SelExpr_0815	TRUE	always selected
SelExpr_0816	TSPC_Serv_SS_unstruct	USSD supported
SelExpr_0817	TSPC_Serv_SS_unstruct AND SelExpr_0301	USSD and CC protocol for at least one BC supported
SelExpr_0818	TSPC_Serv_SS_AoCC AND TSPC_SIMRmv	AOCC and SIM removable without power down supported
SelExpr_0819	TSPC_Serv_SS_AoCC AND TSPC_Serv_SS_HOLD	AOCC and Call Hold supported
SelExpr_0820	TSPC_Serv_SS_AoCC AND TSPC_Serv_SS_MPTY	AOCC and Multi Party service supported
SelExpr_0821	TSPC_Serv_SS_AoCC AND TSPC_SwitchOnOff	AOCC and switch on/off supported
SelExpr_0900	TSPC_SMS	SMS test group
SelExpr_0901	TSPC_Serv_TS21 AND SelExpr_0301	MT/PP supported and CC protocol for at least one BC supported

SelExpr_0902	TSPC_Serv_TS22 AND TSPC_Serv_TS21 AND SelExpr_0301	MO/PP supported and MT/PP supported and CC protocol for at least one BC supported
SelExpr_0903	TSPC_Serv_TS22 AND TSPC_StoreRcvSMSME AND TSPC_StoreRcvSMSSIM	MO/PP supported AND storage of SMS in the ME supported AND storage of SMS in the SIM supported
SelExpr_0904	TSPC_Serv_TS22 AND TSPC_Serv_TS21 AND TSPC_SMSStatusRepCap	MO/PP supported AND MT/PP supported AND SMS Status report capabilities supported
SelExpr_0905	TSPC_Serv_TS21 AND TSPC_DispRcvSMS	MT/PP supported AND display of received short message supported
SelExpr_0906	TSPC_Serv_TS21 AND TSPC_DispRcvSMS AND(TSPC_StoreRcvSMSME OR TSPC_StoreRcvSMSSIM)	MT/PP supported AND display of received short message supported AND (storage of SMS in the ME supported OR storage of SMS in the SIM supported)
SelExpr_0907	TSPC_Serv_TS21 AND TSPC_StoreRcvSMSSIM	MT/PP supported AND display of received short message supported AND storage of SMS in the SIM supported
SelExpr_0908	TSPC_Serv_TS21 AND TSPC_ReplaceSMS AND TSPC_DispRcvSMS	MT/PP supported AND "replace short message" and "display of received short message" supported
SelExpr_0909	TSPC_Serv_TS22 AND TSPC_Serv_TS21 AND TSPC_ReplyProc AND TSPC_DispRcvSMS	MO/PP supported AND MT/PP supported AND "reply procedure" and "display of received short message" supported
SelExpr_1000	TSPC_EGSM	E- band supported
SelExpr_1001	SelExpr_0301	CC protocol for at least one BC supported
SelExpr_1002	TSPC_Serv_TS11 OR TSPC_Serv_TS61 OR TSPC_Serv_TS62	telephony supported or alternate speech/data supported or Teleservice automatic G3 fax supported
SelExpr_1003	TSPC_Serv_TS12	emergency call supported
Detailed Comments:		

Test suite constant declarations

Test Suite Constant Declarations			
Constant Name	**Type**	**Value**	**Comments**
C_arfcnA	INTEGER	20	ARFCN 20
C_arfcnB	INTEGER	10	ARFCN of cell B
C_arfcnC	INTEGER	30	ARFCN 30
C_arfcnH	INTEGER	44	ARFCN 44
C_arfcnAd	INTEGER	590	ARFCN 590
C_arfcnBd	INTEGER	520	ARFCN 520
C_arfcnCd	INTEGER	700	ARFCN 700
C_arfcnHd	INTEGER	810	ARFCN 810
C_arfcn_tchA	INTEGER	30	ARFCN f30
C_arfcn_tchAd	INTEGER	650	ARFCN 650
C_arfcn_tchB	INTEGER	114	ARFCN 114
C_arfcn_tchBd	INTEGER	844	ARFCN 844
C_arfcnEgsm	INTEGER	990	ARFCN 990
C_arfcnEgsm_iacmd	INTEGER	20	ARFCN 20
C_arfcnEgsm_asscmd	INTEGER	20	ARFCN 20
C_BCCHcarrierB_ho	INTEGER	40	BCCH frequency number of cell B for Ho cases (GSM: 40)
C_BCCHcarrierB_hod	INTEGER	764	BCCH frequency number of cell B for Ho cases (DCS:764)
C_Noarfcn	INTEGER	65534	No arfcn defined (hopping channel)
C_ATT_0	INTEGER	0	Attach value 0
C_ATT_1	INTEGER	1	Attach value 1
C_NCC	NCC	'001'B	Network color code
C_NCC_0	NCC	'000'B	Network color code value '000'B
C_NCC_1	NCC	'001'B	Network color code value '001'B
C_NCC_2	NCC	'010'B	Network color code value '010'B
C_NCC_3	NCC	'011'B	Network color code value '011'B
C_NCC_4	NCC	'100'B	Network color code value '100'B
C_NCC_5	NCC	'101'B	Network color code value '101'B
C_NCC_6	NCC	'110'B	Network color code value '110'B
C_NCC_7	NCC	'111'B	Network color code value '111'B
C_NCCP_2	NCCP	'02'O	Network color code permitted value '02'O
C_NCCP_10	NCCP	'0A'O	Network color code permitted value '0A'O
C_BCC	BCC	'101'B	Base Station color code
C_BCC_0	BCC	'000'B	Base Station color code value 0
C_BCC_1	BCC	'001'B	Base Station color code value 1
C_BCC_3	BCC	'011'B	Base Station color code value 3
C_BCC_5	BCC	'101'B	Base Station color code value 5
C_BCC_7	BCC	'111'B	Base Station color code value 7
C_cch_1nonComb	B_3	'000'B	cch_con = 1 basic ch non combined
C_cch_1Comb	B_3	'001'B	cch_con = 1 basic ch combined
C_cch_2Comb	B_3	'011'B	cch_con = 2 basic ch combined
C_cch_3Comb	B_3	'101'B	cch_con = 3 basic ch combined
C_cch_4Comb	B_3	'111'B	cch_con = 4 basic ch combined
C_cch_2nonComb	B_3	'010'B	cch_con = 2 basic chs non combined
C_cch_3nonComb	B_3	'100'B	cch_con = 3 basic chs non combined
C_cch_4nonComb	B_3	'110'B	cch_con = 4 basic chs non combined
C_cch_XnonComb	B_3	'??0'B	cch_con = non combined all number value permitted
C_cch_NotAllowed	B_3	'??1'B	cch_con = value reserved after testing '001'B
C_cchd_e_407	OCTETSTRING	'8DEA0DF4CC6C4AFAFB100	f-list for cellchdescr using of 128

		00000000000'O	format. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}. Length = 16 TC_26_10_4, k=2, c=1 TC_26.10.6, c=1
C_cchd_e_408	OCTETSTRING	'8A0A1CFD3EF4610E2FFFFA0000000000'O	f-list for cellchdescr using of 256 format. The coded set is {20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114,115} Length = 16 TC_26_10_4, k=2, c=2 TC_26.10.6, c=2
C_cchd_e_409	OCTETSTRING	'89EA037F433C7B042BFEBFEC10000000'O	f-list for cellchdescr using of 512 format. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}. Length = 16 TC_26_10_4, k=2, c=3 TC: 26.10.6 c=3
C_cchd_e_410	OCTETSTRING	'841EEA893EF98143B161000000000000'O	f-list for cellchdescr using of 1024 format. Length = 16
C_cchd_e_411	OCTETSTRING	'8FEA703E084210000000000000000000'O	f-list for cellchdescr using of variable bit format. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}. Length = 16 TC_26_10_4, k=2, c=5 TC: 26.10.6 c=5
C_cchd_e_412	OCTETSTRING	'00020800000007F020000008000080000'O	f-list for cellchdescr using of bit map 0 format. The coded set is {20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114,115} Length = 16 TC_26_10_4, k=2, c=6
C_cchd_e_414	OCTETSTRING	'00000000000000002000002000020000000'O	f-list for cellchdescr using of bit map 0 format. Length = 16 f_list: 30, 50 ,70 TC: 26.10.6 c=6
C_cchd_e_415	OCTETSTRING	'841EEA893EF981438000000000000000'O	f-list for cellchdescr using range 1024 format. The coded set is {0, 30, 40, 66, 80, 1005, 1010, 1015}. Length = 16 TC_26_10_4, k=2, c=4 TC: 26.10.6 c=4
C_ChMod_s	CHMOD_VAL	'00000000'B	Mode : signalling only.
C_ChMod_r	CHMOD_VAL	'00000001'B	Mode : speech full or half rate.
C_ChMod_12k	CHMOD_VAL	'00000011'B	Mode : 12k radio rate.
C_ChMod_6k	CHMOD_VAL	'00001011'B	Mode : 6k radio rate.
C_ChMod_3k	CHMOD_VAL	'00010011'B	Mode : 3.6k radio rate.
C_Sap0	SAPID	'00'O	service access point 0
C_Sap3	SAPID	'03'O	service access point 3
C_CellA	CellID	"C_CellA"	cell A -- cell 1
C_CellB	CellID	"C_CellB"	cell B -- cell 2
C_CellC	CellID	"C_CellC"	cell C -- cell 3
C_CellD	CellID	"C_CellD"	cell D -- cell 4
C_CellE	CellID	"C_CellE"	cell E -- cell 5
C_CellF	CellID	"C_CellF"	cell F -- cell 6
C_CellG	CellID	"C_CellG"	cell G -- cell 7
C_CellH	CellID	"C_CellH"	cell H -- cell 8
C_ci_cellA	CI	'0001'O	Cell Id for cell A
C_ci_cellB	CI	'0002'O	Cell Id for cell B
C_ci_cellC	CI	'0003'O	Cell Id for cell C
C_ci_cellD	CI	'0004'O	Cell Id for cell D

C_ci_cellE	CI	'0005'O	Cell Id for cell E
C_ci_cellF	CI	'0006'O	Cell Id for cell F
C_ci_cellG	CI	'0007'O	Cell Id for cell G
C_ci_cellH	CI	'0008'O	Cell Id for cell H
C_cksn_nokey	BITSTRING	'111'B	No key available
C_E_default	INTEGER	63	Default field strength
C_E_suitable	INTEGER	45	Field strength of a suitable cell
C_E_notsuitable	INTEGER	0	Field strength of a not suitable cell for reselection
C_E_notsuitable_fSel	INTEGER	24	Field strength of a not suitable cell for selection
C_flist_e_401	OCTETSTRING	'8DF68AEC00'O	It includes the list of f's in EGSM test cases. The coded set is {1005, 1010, 1015}. range 128 format Length = 5 TC_26_10_4, k=1, c=1 TC: 26.10.5.1 k=1/2,c=1
C_flist_e_402	OCTETSTRING	'8A2481FF03F8'O	It includes the list of f's in EGSM test cases. The coded set is {73, 74, 75, 76, 77}, range 256 format. Length = 6 TC_26_10_4, k=1, c=2 TC: 26.10.5.1 k=1/2,c=2
C_flist_e_403	OCTETSTRING	'89EA00BFC040'O	It includes the list of f's in EGSM test cases. The coded set is {980, 981, 982, 983}, range 512 format. Length = 6 TC_26_10_4, k=1, c=3 TC: 26.10.5.1 k=1/2,c=3
C_flist_e_404	OCTETSTRING	'801EED02BEC0'O	It includes the list of f's in EGSM test cases. The coded set is {30, 40, 1010, 1015}, range 1024 format. Length = 6 TC_26_10_4, k=1, c=4 TC: 26.10.5.1 k=1/2,c=4
C_flist_e_405	OCTETSTRING	'8FEA001E000010'O	It includes the list of f's in EGSM test cases. The coded set is {980, 991, 992, 993, 994, 1015}. Length = 7 TC_26_10_4, k=1, c=5 TC: 26.10.5.1 k=1/2,c=5
C_flist_e_406	OCTETSTRING	'00000000000000020000008000080000'O	It includes the list of f's in EGSM test cases. The coded set is {20, 40, 66}, bitmap 0 format. Length = 16 TC_26_10_4, k=1, c=6 TC: 26.10.5.1 k=2,c=6
C_flist_e_407	OCTETSTRING	'8DEA0DF4CC6C4AFAFB100000000000000'O	It includes the list of f's in EGSM test cases. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}, range 128 format. Length = 16 TC_26_10_4, k=2, c=1 TC: 26.10.5.1 k=3,c=1 TC: 26.10.6 c=1
C_flist_e_408	OCTETSTRING	'8A0A1CFD3EF4610E2FFFFA0000'O	It includes the list of f's in EGSM test cases. The coded set is {20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114, 115}, range 256 format. Length = 13 TC_26_10_4, k=2, c=2 TC: 26.10.5.1 k=3,c=2 TC: 26.10.6 c=2
C_flist_e_409	OCTETSTRING	'89EA037F433C7B042BFEBFEC10000000'O	It includes the list of f's in EGSM test cases. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010,

			1015}, range 512 format. Length = 16 TC_26_10_4, k=2, c=3 TC: 26.10.5.2 k=3,c=3 TC: 26.10.6 c=3
C_flist_e_410	OCTETSTRING	'841EEA893EF98143B161'O	Length = 10 TC: 26.10.5.2 k=3,c=4
C_flist_e_411	OCTETSTRING	'8FEA703E084210'O	It includes the list of f's in EGSM test cases. The coded set is {980, 981, 982, 983, 990, 991, 992, 993, 994, 1000, 1005, 1010, 1015}, variable bitmap format. Length = 7 TC_26_10_4, k=2, c=5 TC: 26.10.5.1 k=3,c=5 TC: 26.10.6 c=5
C_flist_e_412	OCTETSTRING	'0002080000007F020000008000080000'O	It includes the list of f's in EGSM test cases. The coded set is {20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114, 115}, bitmap 0 format. Length = 16 TC_26_10_4, k=2, c=6 TC: 26.10.5.1 k=3,c=6
C_flist_e_415	OCTETSTRING	'841EEA893EF9814380'O	It includes the list of f's in EGSM test cases. The coded set is {0, 30, 40, 66, 80, 1005, 1010, 1015}, range 1024 format. Length = 9 Format: 1024 TC_26_10_4, k=2, c=4 TC: 26.10.6 c=4
C_flist_e_416	OCTETSTRING	'0002080000007F020000008000080000'O	bit map 0 format. The coded set is {20, 40, 66, 73, 74, 75, 76, 77, 78, 79, 108, 114} Length = 16
C_BABR_0	B_3	'000'B	BABR value 0
C_BABR_1	B_3	'001'B	BABR value 1
C_BABR_2	B_3	'010'B	BABR value 2
C_BPM_0	B_3	'000'B	BPM value 0, 2 multiframes
C_BPM_2	B_3	'010'B	BPM value 2, 4 multiframes
C_BPM_3	B_3	'011'B	BPM value 3, 5 multiframes
C_BPM_4	B_3	'100'B	BPM value 4, 6 muiframes
C_BPM_7	B_3	'111'B	BPM value 7, 9 multiframes
C_T3212_0	OCTETSTRING	'00'O	T3212 infinite
C_T3212_1	OCTETSTRING	'01'O	T3212 6 mn / 1 deci hours
C_T3212_2	OCTETSTRING	'02'O	T3212 12 mn / 2 deci hours
C_T3212_5	OCTETSTRING	'05'O	T3212 30 mn / 5 deci hours
C_IMSI	INTEGER	0	MI type: IMSI
C_TMSI	INTEGER	1	MI type: TMSI
C_IMEI	INTEGER	2	MI type: IMEI
C_IMEISV	INTEGER	3	MI type: IMEISV
C_shortIMSI	HEXSTRING	'001011234'H	A short IMSI
C_LAC_1	OCTETSTRING	'0001'O	lac 1.
C_LAC_2	OCTETSTRING	'0002'O	lac 2.
C_LAC_3	OCTETSTRING	'0003'O	lac 3.
C_LAC_4	OCTETSTRING	'0004'O	lac 4.
C_LAC_5	OCTETSTRING	'0005'O	lac 5.
C_LAC_6	OCTETSTRING	'0006'O	lac 6.
C_LAC_7	OCTETSTRING	'0007'O	lac 7.
C_LAC_8	OCTETSTRING	'0008'O	lac 8.
C_LAC_1111	OCTETSTRING	'1111'O	lac '1111'H
C_LAC_2222	OCTETSTRING	'2222'O	lac '2222'H
C_LAC_spe	OCTETSTRING	'5344'O	lac '5344'H.
C_lacdeleted	OCTETSTRING	'FFFE'O	lac value deleted.
C_MCC_1	OCTETSTRING	'00F1'O	MCC 1
C_MCC_2	OCTETSTRING	'00F2'O	MCC 2
C_MCC_3	OCTETSTRING	'00F3'O	MCC 3
C_MCC_4	OCTETSTRING	'00F4'O	MCC 4

C_MCC_5	OCTETSTRING	'00F5'O	MCC 5
C_MCC_6	OCTETSTRING	'00F6'O	MCC 6
C_MCC_7	OCTETSTRING	'00F7'O	MCC 7
C_MCC_8	OCTETSTRING	'00F8'O	MCC 8
C_MCC_9	OCTETSTRING	'013B'O	MCC 9
C_Max_1	INTEGER	1	Max Retrans value 1
C_Max_2	INTEGER	2	Max Retrans value 2
C_Max_7	INTEGER	7	Max Retrans value 7
C_NECI_0	INTEGER	0	NECI value 0, not supported
C_NECI_1	INTEGER	1	NECI value 1, supported
C_PLMN_1	OCTETSTRING	'10'O	PLMN 1 (mnc=01)
C_PLMN_2	OCTETSTRING	'20'O	PLMN 2 (mnc=02)
C_PLMN_3	OCTETSTRING	'30'O	PLMN 3 (mnc=03)
C_PLMN_4	OCTETSTRING	'F0'O	PLMN 4
C_PLMN_5	OCTETSTRING	'F2'O	PLMN 5
C_PLMN_6	OCTETSTRING	'F3'O	PLMN 6
C_PLMN_7	OCTETSTRING	'F4'O	PLMN 7
C_PLMN_8	OCTETSTRING	'F5'O	PLMN 8
C_PLMN_9	OCTETSTRING	'F6'O	PLMN 9
C_PLMN_10	OCTETSTRING	'F7'O	PLMN 10
C_PLMN_Home	OCTETSTRING	'10'O	PLMN 1 (mnc=01)
C_TxInt_5	INTEGER	5	Tx Integer value 5
C_TxInt_7	INTEGER	7	Tx Integer value 7
C_TxInt_10	INTEGER	10	Tx Integer value 10
C_NotCombined	BOOLEAN	FALSE	CCCH not combined with SDCCH
C_Combined	BOOLEAN	TRUE	CCCH combined with SDCCH
C_FCCH_A	LOGICCH	"C_FCCH_A"	frequency correction channel of cell A(1)
C_FCCH_B	LOGICCH	"C_FCCH_B"	frequency correction channel of cell B(2)
C_FCCH_C	LOGICCH	"C_FCCH_C"	frequency correction channel of cell C(3)
C_FCCH_D	LOGICCH	"C_FCCH_D"	frequency correction channel of cell D(4)
C_FCCH_E	LOGICCH	"C_FCCH_E"	frequency correction channel of cell E(5)
C_FCCH_F	LOGICCH	"C_FCCH_F"	frequency correction channel of cell F(6)
C_FCCH_G	LOGICCH	"C_FCCH_G"	frequency correction channel of cell G(7)
C_FCCH_H	LOGICCH	"C_FCCH_H"	frequency correction channel of cell H(8)
C_SCH_A	LOGICCH	"C_SCH_A"	sync channel of cell A(1)
C_SCH_B	LOGICCH	"C_SCH_B"	sync channel of cell B(2)
C_SCH_C	LOGICCH	"C_SCH_C"	sync channel of cell C(3)
C_SCH_D	LOGICCH	"C_SCH_D"	sync channel of cell D(4)
C_SCH_E	LOGICCH	"C_SCH_E"	sync channel of cell E(5)
C_SCH_F	LOGICCH	"C_SCH_F"	sync channel of cell F(6)
C_SCH_G	LOGICCH	"C_SCH_G"	sync channel of cell G(7)
C_SCH_H	LOGICCH	"C_SCH_H"	sync channel of cell H(8)
C_BCCH_A_1	LOGICCH	"C_BCCH_A_1"	1st broadcast channel of cell A
C_BCCH_A_2	LOGICCH	"C_BCCH_A_2"	2nd broadcast channel of cell A
C_BCCH_A_3	LOGICCH	"C_BCCH_A_3"	3rd broadcast channel of cell A
C_BCCH_A_4	LOGICCH	"C_BCCH_A_4"	4th broadcast channel of cell A
C_BCCH_B_1	LOGICCH	"C_BCCH_B_1"	1st broadcast channel of cell B
C_BCCH_C_1	LOGICCH	"C_BCCH_C_1"	1st broadcast channel of cell C
C_BCCH_D_1	LOGICCH	"C_BCCH_D_1"	1st broadcast channel of cell D
C_BCCH_E_1	LOGICCH	"C_BCCH_E_1"	1st broadcast channel of cell E
C_BCCH_F_1	LOGICCH	"C_BCCH_F_1"	1st broadcast channel of cell F
C_BCCH_G_1	LOGICCH	"C_BCCH_G_1"	1st broadcast channel of cell G
C_BCCH_H_1	LOGICCH	"C_BCCH_H_1"	1st broadcast channel of cell H
C_CBCH_A	LOGICCH	"C_CBCH_A"	Cell broadcast channel in cell A
C_AGCH_A_1	LOGICCH	"C_AGCH_A_1"	1st down link CCCH (AGCH) of cell A
C_AGCH_A_2	LOGICCH	"C_AGCH_A_2"	2nd down link CCCH (AGCH) of cell A

C_AGCH_A_3	LOGICCH	"C_AGCH_A_3"	3rd down link CCCH (AGCH) of cell A
C_AGCH_A_4	LOGICCH	"C_AGCH_A_4"	4th down link CCCH (AGCH) of cell A
C_AGCH_B_1	LOGICCH	"C_AGCH_B_1"	1st down link CCCH (AGCH) of cell B
C_AGCH_B_2	LOGICCH	"C_AGCH_B_2"	2nd down link CCCH (AGCH) of cell B
C_AGCH_B_3	LOGICCH	"C_AGCH_B_3"	3rd down link CCCH (AGCH) of cell B
C_AGCH_B_4	LOGICCH	"C_AGCH_B_4"	4th down link CCCH (AGCH) of cell B
C_AGCH_C_1	LOGICCH	"C_AGCH_C_1"	1st down link CCCH (AGCH) of cell C
C_AGCH_C_2	LOGICCH	"C_AGCH_C_2"	2nd down link CCCH (AGCH) of cell C
C_AGCH_C_3	LOGICCH	"C_AGCH_C_3"	3rd down link CCCH (AGCH) of cell C
C_AGCH_C_4	LOGICCH	"C_AGCH_C_4"	4th down link CCCH (AGCH) of cell C
C_AGCH_D_1	LOGICCH	"C_AGCH_D_1"	1st down link CCCH (AGCH) of cell D
C_AGCH_E_1	LOGICCH	"C_AGCH_E_1"	1st down link CCCH (AGCH) of cell E
C_AGCH_F_1	LOGICCH	"C_AGCH_F_1"	1st down link CCCH (AGCH) of cell F
C_AGCH_G_1	LOGICCH	"C_AGCH_G_1"	1st down link CCCH (AGCH) of cell G
C_AGCH_H_1	LOGICCH	"C_AGCH_H_1"	1st down link CCCH (AGCH) of cell H
C_PCH_A_1	LOGICCH	"C_PCH_A_1"	1st down link CCCH (PCH) of cell A
C_PCH_A_2	LOGICCH	"C_PCH_A_2"	2nd down link CCCH (PCH) of cell A
C_PCH_A_3	LOGICCH	"C_PCH_A_3"	3rd down link CCCH (PCH) of cell A
C_PCH_A_4	LOGICCH	"C_PCH_A_4"	4th down link CCCH (PCH) of cell A
C_PCH_B_1	LOGICCH	"C_PCH_B_1"	1st down link CCCH (PCH) of cell B
C_PCH_B_2	LOGICCH	"C_PCH_B_2"	2nd down link CCCH (PCH) of cell B
C_PCH_B_3	LOGICCH	"C_PCH_B_3"	3rd down link CCCH (PCH) of cell B
C_PCH_B_4	LOGICCH	"C_PCH_B_4"	4th down link CCCH (PCH) of cell B
C_PCH_C_1	LOGICCH	"C_PCH_C_1"	1st down link CCCH (PCH) of cell C
C_PCH_C_2	LOGICCH	"C_PCH_C_2"	2nd down link CCCH (PCH) of cell C
C_PCH_C_3	LOGICCH	"C_PCH_C_3"	3rd down link CCCH (PCH) of cell C
C_PCH_C_4	LOGICCH	"C_PCH_C_4"	4th down link CCCH (PCH) of cell C
C_PCH_D_1	LOGICCH	"C_PCH_D_1"	1st down link CCCH (PCH and AGCH) of cell D
C_PCH_E_1	LOGICCH	"C_PCH_E_1"	1st down link CCCH (PCH) of cell E
C_PCH_F_1	LOGICCH	"C_PCH_F_1"	1st down link CCCH (PCH) of cell F
C_PCH_G_1	LOGICCH	"C_PCH_G_1"	1st down link CCCH (PCH) of cell G
C_PCH_H_1	LOGICCH	"C_PCH_H_1"	1st down link CCCH (PCH) of cell H
C_RACH_A_1	LOGICCH	"C_RACH_A_1"	1st uplink CCCH (RACH) of cell A
C_RACH_A_2	LOGICCH	"C_RACH_A_2"	2nd uplink CCCH (RACH) of cell A
C_RACH_A_3	LOGICCH	"C_RACH_A_3"	3rd uplink CCCH (RACH) of cell A
C_RACH_A_4	LOGICCH	"C_RACH_A_4"	4th uplink CCCH (RACH) of cell A
C_RACH_B_1	LOGICCH	"C_RACH_B_1"	1st uplink CCCH (RACH) of cell B

C_RACH_B_2	LOGICCH	"C_RACH_B_2"	2nd uplink CCCH (RACH) of cell B
C_RACH_B_3	LOGICCH	"C_RACH_B_3"	3rd uplink CCCH (RACH) of cell B
C_RACH_B_4	LOGICCH	"C_RACH_B_4"	4th uplink CCCH (RACH) of cell B
C_RACH_C_1	LOGICCH	"C_RACH_C_1"	1st uplink CCCH (RACH) of cell C
C_RACH_C_2	LOGICCH	"C_RACH_C_2"	2nd uplink CCCH (RACH) of cell C
C_RACH_C_3	LOGICCH	"C_RACH_C_3"	3rd uplink CCCH (RACH) of cell C
C_RACH_C_4	LOGICCH	"C_RACH_C_4"	4th uplink CCCH (RACH) of cell C
C_RACH_D_1	LOGICCH	"C_RACH_D_1"	1st uplink CCCH (RACH) of cell D
C_RACH_E_1	LOGICCH	"C_RACH_E_1"	1st uplink CCCH (RACH) of cell E
C_RACH_F_1	LOGICCH	"C_RACH_F_1"	1st uplink CCCH (RACH) of cell F
C_RACH_G_1	LOGICCH	"C_RACH_G_1"	1st uplink CCCH (RACH) of cell G
C_RACH_H_1	LOGICCH	"C_RACH_H_1"	1st uplink CCCH (RACH) of cell H
C_FACCHF_A_1	LOGICCH	"C_FACCHF_A_1"	FACCH associated with 1st TCH/F of cell A
C_FACCHF_A_2	LOGICCH	"C_FACCHF_A_2"	FACCH associated with 2nd TCH/F of cell A
C_FACCHF_A_3	LOGICCH	"C_FACCHF_A_3"	FACCH associated with 3rd TCH/F of cell A
C_FACCHF_B_1	LOGICCH	"C_FACCHF_B_1"	FACCH associated with 1st TCH/F of cell B
C_FACCHF_B_2	LOGICCH	"C_FACCHF_B_2"	FACCH associated with 2nd TCH/F of cell B
C_FACCHF_B_3	LOGICCH	"C_FACCHF_B_3"	FACCH associated with 3rd TCH/F of cell B
C_FACCHF_C_1	LOGICCH	"C_FACCHF_C_1"	FACCH associated with 1sd TCH/F of cell C
C_FACCHF_C_2	LOGICCH	"C_FACCHF_C_2"	FACCH associated with 2nd TCH/F of cell C
C_FACCHF_C_3	LOGICCH	"C_FACCHF_C_3"	FACCH associated with 3rd TCH/F of cell C
C_FACCHF_H_1	LOGICCH	"C_FACCHF_H_1"	FACCH associated with 1st TCH/F of cell H
C_FACCHH_A_1	LOGICCH	"C_FACCHH_A_1"	FACCH associated with 1st TCH/H of cell A
C_FACCHH_A_2	LOGICCH	"C_FACCHH_A_2"	FACCH associated with 2nd TCH/H of cell A
C_FACCHH_A_3	LOGICCH	"C_FACCHH_A_3"	FACCH associated with 3rd TCH/H of cell A
C_FACCHH_B_1	LOGICCH	"C_FACCHH_B_1"	FACCH associated with 1st TCH/H of cell B
C_FACCHH_B_2	LOGICCH	"C_FACCHH_B_2"	FACCH associated with 2nd TCH/H of cell B
C_FACCHH_B_3	LOGICCH	"C_FACCHH_B_3"	FACCH associated with 3rd TCH/H of cell B
C_FACCHH_C_1	LOGICCH	"C_FACCHH_C_1"	FACCH associated with 1st TCH/H of cell C
C_FACCHH_C_2	LOGICCH	"C_FACCHH_C_2"	FACCH associated with 2nd TCH/H of cell C
C_FACCHH_C_3	LOGICCH	"C_FACCHH_C_3"	FACCH associated with 3rd TCH/H of cell C
C_FACCHH0_A_1	LOGICCH	"C_FACCHH0_A_1"	FACCH associated with 1st TCH/H0 of cell A
C_FACCHH1_A_1	LOGICCH	"C_FACCHH1_A_1"	FACCH associated with 1st TCH/H1 of cell A
C_FACCHH0_A_2	LOGICCH	"C_FACCHH0_A_2"	FACCH associated with 2nd TCH/H0 of cell A
C_FACCHH1_A_2	LOGICCH	"C_FACCHH1_A_2"	FACCH associated with 2nd TCH/H1 of cell A
C_FACCHH0_A_3	LOGICCH	"C_FACCHH0_A_3"	FACCH associated with 3rd TCH/H0 of cell A
C_FACCHH1_A_3	LOGICCH	"C_FACCHH1_A_3"	FACCH associated with 3rd TCH/H1 of cell A
C_FACCHH0_B_1	LOGICCH	"C_FACCHH0_B_1"	FACCH associated with 1st TCH/H0 of cell B
C_FACCHH1_B_1	LOGICCH	"C_FACCHH1_B_1"	FACCH associated with 1st TCH/H1 of cell B
C_FACCHH0_B_2	LOGICCH	"C_FACCHH0_B_2"	FACCH associated with 2nd TCH/H0 of cell B

C_FACCHH1_B_2	LOGICCH	"C_FACCHH1_B_2"	FACCH associated with 2nd TCH/H1 of cell B
C_FACCHH0_B_3	LOGICCH	"C_FACCHH0_B_3"	FACCH associated with 3rd TCH/H0 of cell B
C_FACCHH1_B_3	LOGICCH	"C_FACCHH1_B_3"	FACCH associated with 3rd TCH/H1 of cell B
C_FACCHH0_C_1	LOGICCH	"C_FACCHH0_C_1"	FACCH associated with 1st TCH/H0 of cell C
C_FACCHH1_C_1	LOGICCH	"C_FACCHH1_C_1"	FACCH associated with 1st TCH/H1 of cell C
C_FACCHH0_C_2	LOGICCH	"C_FACCHH0_C_2"	FACCH associated with 2nd TCH/H0 of cell C
C_FACCHH1_C_2	LOGICCH	"C_FACCHH1_C_2"	FACCH associated with 2nd TCH/H1 of cell C
C_FACCHH0_C_3	LOGICCH	"C_FACCHH0_C_3"	FACCH associated with 3rd TCH/H0 of cell C
C_FACCHH1_C_3	LOGICCH	"C_FACCHH1_C_3"	FACCH associated with 3rd TCH/H1 of cell C
C_SACCH_A	LOGICCH	"C_SACCH_A"	all SACCHs of cell A
C_SACCH_B	LOGICCH	"C_SACCH_B"	all SACCHs of cell B
C_SACCH_C	LOGICCH	"C_SACCH_C"	all SACCHs of cell C
C_SACCH_D	LOGICCH	"C_SACCH_D"	all SACCHs of cell D
C_SACCH_E	LOGICCH	"C_SACCH_E"	all SACCHs of cell E
C_SACCH_F	LOGICCH	"C_SACCH_F"	all SACCHs of cell F
C_SACCH_G	LOGICCH	"C_SACCH_G"	all SACCHs of cell G
C_SACCH_H	LOGICCH	"C_SACCH_H"	all SACCHs of cell H
C_SACCHF_A_1	LOGICCH	"C_SACCHF_A_1"	SACCH associated with 1st TCH/F of cell A
C_SACCHF_A_2	LOGICCH	"C_SACCHF_A_2"	SACCH associated with 2nd TCH/F of cell A
C_SACCHF_B_1	LOGICCH	"C_SACCHF_B_1"	SACCH associated with 1st TCH/F of cell B
C_SACCHF_B_2	LOGICCH	"C_SACCHF_B_2"	SACCH associated with 2nd TCH/F of cell B
C_SACCHF_H_1	LOGICCH	"C_SACCHF_H_1"	SACCH associated with 1st TCH/F of cell H
C_SACCHH_A_1	LOGICCH	"C_SACCHH_A_1"	SACCH associated with 1st TCH/H of cell A
C_SACCHH_A_2	LOGICCH	"C_SACCHH_A_2"	SACCH associated with 2nd TCH/H of cell A
C_SACCHH_A_3	LOGICCH	"C_SACCHH_A_3"	SACCH associated with 3rd TCH/H of cell A
C_SACCHH_B_1	LOGICCH	"C_SACCHH_B_1"	SACCH associated with 1st TCH/H of cell B
C_SACCHH_B_2	LOGICCH	"C_SACCHH_B_2"	SACCH associated with 2nd TCH/H of cell B
C_SACCHH_B_3	LOGICCH	"C_SACCHH_B_3"	SACCH associated with 3rd TCH/H of cell B
C_SACCHH_C_1	LOGICCH	"C_SACCHH_C_1"	SACCH associated with 1st TCH/H of cell C
C_SACCHH_C_2	LOGICCH	"C_SACCHH_C_2"	SACCH associated with 2nd TCH/H of cell C
C_SACCHH_C_3	LOGICCH	"C_SACCHH_C_3"	SACCH associated with 3rd TCH/H of cell C
C_SACCHH0_A_1	LOGICCH	"C_SACCHH0_A_1"	SACCH associated with 1st TCH/H_0 of cell A
C_SACCHH1_A_1	LOGICCH	"C_SACCHH1_A_1"	SACCH associated with 1st TCH/H_1 of cell A
C_SACCHH0_A_2	LOGICCH	"C_SACCHH0_A_2"	SACCH associated with 2nd TCH/H_0 of cell A
C_SACCHH1_A_2	LOGICCH	"C_SACCHH1_A_2"	SACCH associated with 2nd TCH/H_1 of cell A
C_SACCHH0_A_3	LOGICCH	"C_SACCHH0_A_3"	SACCH associated with 3rd TCH/H_0 of cell A
C_SACCHH1_A_3	LOGICCH	"C_SACCHH1_A_3"	SACCH associated with 3rd TCH/H_1 of cell A
C_SACCHH0_B_1	LOGICCH	"C_SACCHH0_B_1"	SACCH associated with 1st TCH/H_0 of cell B
C_SACCHH1_B_1	LOGICCH	"C_SACCHH1_B_1"	SACCH associated with 1st TCH/H_1 of cell B

C_SACCHH0_B_2	LOGICCH	"C_SACCHH0_B_2"	SACCH associated with 2nd TCH/H_0 of cell B
C_SACCHH1_B_2	LOGICCH	"C_SACCHH1_B_2"	SACCH associated with 2nd TCH/H_1 of cell B
C_SACCHH0_B_3	LOGICCH	"C_SACCHH0_B_3"	SACCH associated with 3rd TCH/H_0 of cell B
C_SACCHH1_B_3	LOGICCH	"C_SACCHH1_B_3"	SACCH associated with 3rd TCH/H_1 of cell B
C_SACCHH0_C_1	LOGICCH	"C_SACCHH0_C_1"	SACCH associated with 1st TCH/H_0 of cell C
C_SACCHH1_C_1	LOGICCH	"C_SACCHH1_C_1"	SACCH associated with 1st TCH/H_1 of cell C
C_SACCHH0_C_2	LOGICCH	"C_SACCHH0_C_2"	SACCH associated with 2nd TCH/H_0 of cell C
C_SACCHH1_C_2	LOGICCH	"C_SACCHH1_C_2"	SACCH associated with 2nd TCH/H_1 of cell C
C_SACCHH0_C_3	LOGICCH	"C_SACCHH0_C_3"	SACCH associated with 3rd TCH/H_0 of cell C
C_SACCHH1_C_3	LOGICCH	"C_SACCHH1_C_3"	SACCH associated with 3rd TCH/H_1 of cell C
C_SACCHC4_A	LOGICCH	"C_SACCHC4_A"	SACCHC4 channel of cell A
C_SACCHC4_B	LOGICCH	"C_SACCHC4_B"	SACCHC4 channel of cell B
C_SACCHC4_C	LOGICCH	"C_SACCHC4_C"	SACCHC4 channel of cell C
C_SACCHC4_D	LOGICCH	"C_SACCHC4_D"	SACCHC4 channel of cell D
C_SACCHC4_E	LOGICCH	"C_SACCHC4_E"	SACCHC4 channel of cell E
C_SACCHC4_F	LOGICCH	"C_SACCHC4_F"	SACCHC4 channel of cell F
C_SACCHC4_G	LOGICCH	"C_SACCHC4_G"	SACCHC4 channel of cell G
C_SACCHC4_H	LOGICCH	"C_SACCHC4_H"	SACCHC4 channel of cell H
C_SACCHC40_A	LOGICCH	"C_SACCHC40_A"	SACCH/C4(0) associated with SDCCH/4(0) of cell A
C_SACCHC41_A	LOGICCH	"C_SACCHC41_A"	SACCH/C4(1) associated with SDCCH/4(1) of cell A
C_SACCHC42_A	LOGICCH	"C_SACCHC42_A"	SACCH/C4(2) associated with SDCCH/4(2) of cell A
C_SACCHC43_A	LOGICCH	"C_SACCCH43_A"	SACCH/C4(3) associated with SDCCH/4(3) of cell A
C_SACCHC40_B	LOGICCH	"C_SACCHC40_B"	SACCH/C4(0) associated with SDCCH/4(0) of cell B
C_SACCHC41_B	LOGICCH	"C_SACCHC41_B"	SACCH/C4(1) associated with SDCCH/4(1) of cell B
C_SACCHC42_B	LOGICCH	"C_SACCHC42_B"	SACCH/C4(2) associated with SDCCH/4(2) of cell B
C_SACCHC43_B	LOGICCH	"C_SACCHC43_B"	SACCH/C4(3) associated with SDCCH/4(3) of cell B
C_SACCHC40_C	LOGICCH	"C_SACCHC40_C"	SACCH/C4(0) associated with SDCCH/4(0) of cell C
C_SACCHC41_C	LOGICCH	"C_SACCHC41_C"	SACCH/C4(1) associated with SDCCH/4(1) of cell C
C_SACCHC42_C	LOGICCH	"C_SACCHC42_C"	SACCH/C4(2) associated with SDCCH/4(2) of cell C
C_SACCHC43_C	LOGICCH	"C_SACCHC43_C"	SACCH/C4(3) associated with SDCCH/4(3) of cell C
C_SACCHC40_D	LOGICCH	"C_SACCHC40_D"	SACCH/C4(0) associated with SDCCH/4(0) of cell D
C_SACCHC41_D	LOGICCH	"C_SACCHC41_D"	SACCH/C4(1) associated with SDCCH/4(1) of cell D
C_SACCHC42_D	LOGICCH	"C_SACCHC42_D"	SACCH/C4(2) associated with SDCCH/4(2) of cell D
C_SACCHC43_D	LOGICCH	"C_SACCCH43_D"	SACCH/C4(3) associated with SDCCH/4(3) of cell D
C_SACCHC40_E	LOGICCH	"C_SACCHC40_E"	SACCH/C4(0) associated with SDCCH/4(0) of cell E
C_SACCHC41_E	LOGICCH	"C_SACCHC41_E"	SACCH/C4(1) associated with SDCCH/4(1) of cell E
C_SACCHC42_E	LOGICCH	"C_SACCHC42_E"	SACCH/C4(2) associated with SDCCH/4(2) of cell E
C_SACCHC43_E	LOGICCH	"C_SACCHC43_E"	SACCH/C4(3) associated with SDCCH/4(3) of cell E
C_SACCHC40_F	LOGICCH	"C_SACCHC40_F"	SACCH/C4(0) associated with SDCCH/4(0) of cell F

C_SACCHC41_F	LOGICCH	"C_SACCHC41_F"	SACCH/C4(1) associated with SDCCH/4(1) of cell F
C_SACCHC42_F	LOGICCH	"C_SACCHC42_F"	SACCH/C4(2) associated with SDCCH/4(2) of cell F
C_SACCHC43_F	LOGICCH	"C_SACCHC43_F"	SACCH/C4(3) associated with SDCCH/4(3) of cell F
C_SACCHC40_G	LOGICCH	"C_SACCHC40_G"	SACCH/C4(0) associated with SDCCH/4(0) of cell G
C_SACCHC41_G	LOGICCH	"C_SACCHC41_G"	SACCH/C4(1) associated with SDCCH/4(1) of cell G
C_SACCHC42_G	LOGICCH	"C_SACCHC42_G"	SACCH/C4(2) associated with SDCCH/4(2) of cell G
C_SACCHC43_G	LOGICCH	"C_SACCHC43_G"	SACCH/C4(3) associated with SDCCH/4(3) of cell G
C_SACCHC40_H	LOGICCH	"C_SACCHC40_C"	SACCH/C4(0) associated with SDCCH/4(0) of cell H
C_SACCHC41_H	LOGICCH	"C_SACCHC41_C"	SACCH/C4(1) associated with SDCCH/4(1) of cell H
C_SACCHC42_H	LOGICCH	"C_SACCHC42_C"	SACCH/C4(2) associated with SDCCH/4(2) of cell H
C_SACCHC43_H	LOGICCH	"C_SACCHC43_C"	SACCH/C4(3) associated with SDCCH/4(3) of cell H
C_SACCHC8_A_1	LOGICCH	"C_SACCHC8_A_1"	1st SACCH/8 channel of cell A
C_SACCHC8_A_2	LOGICCH	"C_SACCHC8_A_2"	2nd SACCH/8 channel of cell A
C_SACCHC8_A_3	LOGICCH	"C_SACCHC8_A_3"	3rd SACCH/8 channel of cell A
C_SACCHC8_B_1	LOGICCH	"C_SACCHC8_B_1"	1st SACCH/8 channel of cell B
C_SACCHC8_B_2	LOGICCH	"C_SACCHC8_B_2"	2nd SACCH/8 channel of cell B
C_SACCHC8_B_3	LOGICCH	"C_SACCHC8_B_3"	3rd SACCH/8 channel of cell B
C_SACCHC8_C_1	LOGICCH	"C_SACCHC8_C_1"	1st SACCH/8 channel of cell C
C_SACCHC8_C_2	LOGICCH	"C_SACCHC8_C_2"	2nd SACCH/8 channel of cell C
C_SACCHC8_C_3	LOGICCH	"C_SACCHC8_C_3"	3rd SACCH/8 channel of cell C
C_SACCHC80_A_1	LOGICCH	"C_SACCHC80_A_1"	SACCH/C8(0) associated with 1st SDCCH/8(0) of cell A
C_SACCHC81_A_1	LOGICCH	"C_SACCHC81_A_1"	SACCH/C8(1) associated with 1st SDCCH/8(1) of cell A
C_SACCHC82_A_1	LOGICCH	"C_SACCHC82_A_1"	SACCH/C8(2) associated with 1st SDCCH/8(2) of cell A
C_SACCHC83_A_1	LOGICCH	"C_SACCHC83_A_1"	SACCH/C8(3) associated with 1st SDCCH/8(3) of cell A
C_SACCHC84_A_1	LOGICCH	"C_SACCHC84_A_1"	SACCH/C8(4) associated with 1st SDCCH/8(4) of cell A
C_SACCHC85_A_1	LOGICCH	"C_SACCHC85_A_1"	SACCH/C8(5) associated with 1st SDCCH/8(5) of cell A
C_SACCHC86_A_1	LOGICCH	"C_SACCHC86_A_1"	SACCH/C8(6) associated with 1st SDCCH/8(6) of cell A
C_SACCHC87_A_1	LOGICCH	"C_SACCHC87_A_1"	SACCH/C8(7) associated with 1st SDCCH/8(7) of cell A
C_SACCHC80_A_2	LOGICCH	"C_SACCHC80_A_2"	SACCH/C8(0) associated with 2nd SDCCH/8(0) of cell A
C_SACCHC81_A_2	LOGICCH	"C_SACCHC81_A_2"	SACCH/C8(1) associated with 2nd SDCCH/8(1) of cell A
C_SACCHC82_A_2	LOGICCH	"C_SACCHC82_A_2"	SACCH/C8(2) associated with 2nd SDCCH/8(2) of cell A
C_SACCHC83_A_2	LOGICCH	"C_SACCHC83_A_2"	SACCH/C8(3) associated with 2nd SDCCH/8(3) of cell A
C_SACCHC84_A_2	LOGICCH	"C_SACCHC84_A_2"	SACCH/C8(4) associated with 2nd SDCCH/8(4) of cell A
C_SACCHC85_A_2	LOGICCH	"C_SACCHC85_A_2"	SACCH/C8(5) associated with 2nd SDCCH/8(5) of cell A
C_SACCHC86_A_2	LOGICCH	"C_SACCHC86_A_2"	SACCH/C8(6) associated with 2nd SDCCH/8(6) of cell A
C_SACCHC87_A_2	LOGICCH	"C_SACCHC87_A_2"	SACCH/C8(7) associated with 2nd SDCCH/8(7) of cell A
C_SACCHC80_A_3	LOGICCH	"C_SACCHC80_A_3"	SACCH/C8(0) associated with 3rd SDCCH/8(0) of cell A
C_SACCHC81_A_3	LOGICCH	"C_SACCHC81_A_3"	SACCH/C8(1) associated with 3rd SDCCH/8(1) of cell A
C_SACCHC82_A_3	LOGICCH	"C_SACCHC82_A_3"	SACCH/C8(2) associated with 3rd SDCCH/8(2) of cell A
C_SACCHC83_A_3	LOGICCH	"C_SACCHC83_A_3"	SACCH/C8(3) associated with 3rd

			SDCCH/8(3) of cell A
C_SACCHC84_A_3	LOGICCH	"C_SACCHC84_A_3"	SACCH/C8(4) associated with 3rd SDCCH/8(4) of cell A
C_SACCHC85_A_3	LOGICCH	"C_SACCHC85_A_3"	SACCH/C8(5) associated with 3rd SDCCH/8(5) of cell A
C_SACCHC86_A_3	LOGICCH	"C_SACCHC86_A_3"	SACCH/C8(6) associated with 3rd SDCCH/8(6) of cell A
C_SACCHC87_A_3	LOGICCH	"C_SACCHC87_A_3"	SACCH/C8(7) associated with 3rd SDCCH/8(7) of cell A
C_SACCHC80_B_1	LOGICCH	"C_SACCHC80_B_1"	SACCH/C8(0) associated with 1st SDCCH/8(0) of cell B
C_SACCHC81_B_1	LOGICCH	"C_SACCHC81_B_1"	SACCH/C8(1) associated with 1st SDCCH/8(1) of cell B
C_SACCHC82_B_1	LOGICCH	"C_SACCHC82_B_1"	SACCH/C8(2) associated with 1st SDCCH/8(2) of cell B
C_SACCHC83_B_1	LOGICCH	"C_SACCHC83_B_1"	SACCH/C8(3) associated with 1st SDCCH/8(3) of cell B
C_SACCHC84_B_1	LOGICCH	"C_SACCHC84_B_1"	SACCH/C8(4) associated with 1st SDCCH/8(4) of cell B
C_SACCHC85_B_1	LOGICCH	"C_SACCHC85_B_1"	SACCH/C8(5) associated with 1st SDCCH/8(5) of cell B
C_SACCHC86_B_1	LOGICCH	"C_SACCHC86_B_1"	SACCH/C8(6) associated with 1st SDCCH/8(6) of cell B
C_SACCHC87_B_1	LOGICCH	"C_SACCHC87_B_1"	SACCH/C8(7) associated with 1st SDCCH/8(7) of cell B
C_SACCHC80_B_2	LOGICCH	"C_SACCHC80_B_2"	SACCH/C8(0) associated with 2nd SDCCH/8(0) of cell B
C_SACCHC81_B_2	LOGICCH	"C_SACCHC81_B_2"	SACCH/C8(1) associated with 2nd SDCCH/8(1) of cell B
C_SACCHC82_B_2	LOGICCH	"C_SACCHC82_B_2"	SACCH/C8(2) associated with 2nd SDCCH/8(2) of cell B
C_SACCHC83_B_2	LOGICCH	"C_SACCHC83_B_2"	SACCH/C8(3) associated with 2nd SDCCH/8(3) of cell B
C_SACCHC84_B_2	LOGICCH	"C_SACCHC84_B_2"	SACCH/C8(4) associated with 2nd SDCCH/8(4) of cell B
C_SACCHC85_B_2	LOGICCH	"C_SACCHC85_B_2"	SACCH/C8(5) associated with 2nd SDCCH/8(5) of cell B
C_SACCHC86_B_2	LOGICCH	"C_SACCHC86_B_2"	SACCH/C8(6) associated with 2nd SDCCH/8(6) of cell B
C_SACCHC87_B_2	LOGICCH	"C_SACCHC87_B_2"	SACCH/C8(7) associated with 2nd SDCCH/8(7) of cell B
C_SACCHC80_B_3	LOGICCH	"C_SACCHC80_B_3"	SACCH/C8(0) associated with 3rd SDCCH/8(0) of cell B
C_SACCHC81_B_3	LOGICCH	"C_SACCHC81_B_3"	SACCH/C8(1) associated with 3rd SDCCH/8(1) of cell B
C_SACCHC82_B_3	LOGICCH	"C_SACCHC82_B_3"	SACCH/C8(2) associated with 3rd SDCCH/8(2) of cell B
C_SACCHC83_B_3	LOGICCH	"C_SACCHC83_B_3"	SACCH/C8(3) associated with 3rd SDCCH/8(3) of cell B
C_SACCHC84_B_3	LOGICCH	"C_SACCHC84_B_3"	SACCH/C8(4) associated with 3rd SDCCH/8(4) of cell B
C_SACCHC85_B_3	LOGICCH	"C_SACCHC85_B_3"	SACCH/C8(5) associated with 3rd SDCCH/8(5) of cell B
C_SACCHC86_B_3	LOGICCH	"C_SACCHC86_B_3"	SACCH/C8(6) associated with 3rd SDCCH/8(6) of cell B
C_SACCHC87_B_3	LOGICCH	"C_SACCHC87_B_3"	SACCH/C8(7) associated with 3rd SDCCH/8(7) of cell B
C_SACCHC80_C_1	LOGICCH	"C_SACCHC80_C_1"	SACCH/C8(0) associated with 1st SDCCH/8(0) of cell C
C_SACCHC81_C_1	LOGICCH	"C_SACCHC81_C_1"	SACCH/C8(1) associated with 1st SDCCH/8(1) of cell C
C_SACCHC82_C_1	LOGICCH	"C_SACCHC82_C_1"	SACCH/C8(2) associated with 1st SDCCH/8(2) of cell C
C_SACCHC83_C_1	LOGICCH	"C_SACCHC83_C_1"	SACCH/C8(3) associated with 1st SDCCH/8(3) of cell C
C_SACCHC84_C_1	LOGICCH	"C_SACCHC84_C_1"	SACCH/C8(4) associated with 1st SDCCH/8(4) of cell C
C_SACCHC85_C_1	LOGICCH	"C_SACCHC85_C_1"	SACCH/C8(5) associated with 1st SDCCH/8(5) of cell C
C_SACCHC86_C_1	LOGICCH	"C_SACCHC86_C_1"	SACCH/C8(6) associated with 1st SDCCH/8(6) of cell C

C_SACCHC87_C_1	LOGICCH	"C_SACCHC87_C_1"	SACCH/C8(7) associated with 1st SDCCH/8(7) of cell C
C_SACCHC80_C_2	LOGICCH	"C_SACCHC80_C_2"	SACCH/C8(0) associated with 2nd SDCCH/8(0) of cell C
C_SACCHC81_C_2	LOGICCH	"C_SACCHC81_C_2"	SACCH/C8(1) associated with 2nd SDCCH/8(1) of cell C
C_SACCHC82_C_2	LOGICCH	"C_SACCHC82_C_2"	SACCH/C8(2) associated with 2nd SDCCH/8(2) of cell C
C_SACCHC83_C_2	LOGICCH	"C_SACCHC83_C_2"	SACCH/C8(3) associated with 2nd SDCCH/8(3) of cell C
C_SACCHC84_C_2	LOGICCH	"C_SACCHC84_C_2"	SACCH/C8(4) associated with 2nd SDCCH/8(4) of cell C
C_SACCHC85_C_2	LOGICCH	"C_SACCHC85_C_2"	SACCH/C8(5) associated with 2nd SDCCH/8(5) of cell C
C_SACCHC86_C_2	LOGICCH	"C_SACCHC86_C_2"	SACCH/C8(6) associated with 2nd SDCCH/8(6) of cell C
C_SACCHC87_C_2	LOGICCH	"C_SACCHC87_C_2"	SACCH/C8(7) associated with 2nd SDCCH/8(7) of cell C
C_SACCHC80_C_3	LOGICCH	"C_SACCHC80_C_3"	SACCH/C8(0) associated with 3rd SDCCH/8(0) of cell C
C_SACCHC81_C_3	LOGICCH	"C_SACCHC81_C_3"	SACCH/C8(1) associated with 3rd SDCCH/8(1) of cell C
C_SACCHC82_C_3	LOGICCH	"C_SACCHC82_C_3"	SACCH/C8(2) associated with 3rd SDCCH/8(2) of cell C
C_SACCHC83_C_3	LOGICCH	"C_SACCHC83_C_3"	SACCH/C8(3) associated with 3rd SDCCH/8(3) of cell C
C_SACCHC84_C_3	LOGICCH	"C_SACCHC84_C_3"	SACCH/C8(4) associated with 3rd SDCCH/8(4) of cell C
C_SACCHC85_C_3	LOGICCH	"C_SACCHC85_C_3"	SACCH/C8(5) associated with 3rd SDCCH/8(5) of cell C
C_SACCHC86_C_3	LOGICCH	"C_SACCHC86_C_3"	SACCH/C8(6) associated with 3rd SDCCH/8(6) of cell C
C_SACCHC87_C_3	LOGICCH	"C_SACCHC87_C_3"	SACCH/C8(7) associated with 3rd SDCCH/8(7) of cell C
C_SDCCH4_A	LOGICCH	"C_SDCCH4_A"	SDCCH/4 channel of cell A
C_SDCCH4_B	LOGICCH	"C_SDCCH4_B"	SDCCH/4 channel of cell B
C_SDCCH4_C	LOGICCH	"C_SDCCH4_C"	SDCCH/4 channel of cell C
C_SDCCH4_D	LOGICCH	"C_SDCCH4_D"	SDCCH/4 channel of cell D
C_SDCCH4_E	LOGICCH	"C_SDCCH4_E"	SDCCH/4 channel of cell E
C_SDCCH4_F	LOGICCH	"C_SDCCH4_F"	SDCCH/4 channel of cell F
C_SDCCH4_G	LOGICCH	"C_SDCCH4_G"	SDCCH/4 channel of cell G
C_SDCCH4_H	LOGICCH	"C_SDCCH4_H"	SDCCH/4 channel of cell H
C_SDCCH40_A	LOGICCH	"C_SDCCH40_A"	SDCCH/4(0) of cell A
C_SDCCH41_A	LOGICCH	"C_SDCCH41_A"	SDCCH/4(1) of cell A
C_SDCCH42_A	LOGICCH	"C_SDCCH42_A"	SDCCH/4(2) of cell A
C_SDCCH43_A	LOGICCH	"C_SDCCH43_A"	SDCCH/4(3) of cell A
C_SDCCH40_B	LOGICCH	"C_SDCCH40_B"	SDCCH/4(0) of cell B
C_SDCCH41_B	LOGICCH	"C_SDCCH41_B"	SDCCH/4(1) of cell B
C_SDCCH42_B	LOGICCH	"C_SDCCH42_B"	SDCCH/4(2) of cell B
C_SDCCH43_B	LOGICCH	"C_SDCCH43_B"	SDCCH/4(3) of cell B
C_SDCCH40_C	LOGICCH	"C_SDCCH40_C"	SDCCH/4(0) of cell C
C_SDCCH41_C	LOGICCH	"C_SDCCH41_C"	SDCCH/4(1) of cell C
C_SDCCH42_C	LOGICCH	"C_SDCCH42_C"	SDCCH/4(2) of cell C
C_SDCCH43_C	LOGICCH	"C_SDCCH43_C"	SDCCH/4(3) of cell C
C_SDCCH40_D	LOGICCH	"C_SDCCH40_D"	SDCCH/4(0) of cell D
C_SDCCH41_D	LOGICCH	"C_SDCCH41_D"	SDCCH/4(1) of cell D
C_SDCCH42_D	LOGICCH	"C_SDCCH42_D"	SDCCH/4(2) of cell D
C_SDCCH43_D	LOGICCH	"C_SDCCH43_D"	SDCCH/4(3) of cell D
C_SDCCH40_E	LOGICCH	"C_SDCCH40_E"	SDCCH/4(0) of cell E
C_SDCCH41_E	LOGICCH	"C_SDCCH41_E"	SDCCH/4(1) of cell E
C_SDCCH42_E	LOGICCH	"C_SDCCH42_E"	SDCCH/4(2) of cell E
C_SDCCH43_E	LOGICCH	"C_SDCCH43_E"	SDCCH/4(3) of cell E
C_SDCCH40_F	LOGICCH	"C_SDCCH40_F"	SDCCH/4(0) of cell F
C_SDCCH41_F	LOGICCH	"C_SDCCH41_F"	SDCCH/4(1) of cell F
C_SDCCH42_F	LOGICCH	"C_SDCCH42_F"	SDCCH/4(2) of cell F
C_SDCCH43_F	LOGICCH	"C_SDCCH43_F"	SDCCH/4(3) of cell F
C_SDCCH40_G	LOGICCH	"C_SDCCH40_G"	SDCCH/4(0) of cell G
C_SDCCH41_G	LOGICCH	"C_SDCCH41_G"	SDCCH/4(1) of cell G

C_SDCCH42_G	LOGICCH	"C_SDCCH42_G"	SDCCH/4(2) of cell G
C_SDCCH43_G	LOGICCH	"C_SDCCH43_G"	SDCCH/4(3) of cell G
C_SDCCH40_H	LOGICCH	"C_SDCCH40_H"	SDCCH/4(0) of cell H
C_SDCCH41_H	LOGICCH	"C_SDCCH41_H"	SDCCH/4(1) of cell H
C_SDCCH42_H	LOGICCH	"C_SDCCH42_H"	SDCCH/4(2) of cell H
C_SDCCH43_H	LOGICCH	"C_SDCCH43_H"	SDCCH/4(3) of cell H
C_SDCCH8_A_1	LOGICCH	"C_SDCCH8_A_1"	1st SDCCH/8 channel of cell A
C_SDCCH8_A_2	LOGICCH	"C_SDCCH8_A_2"	2nd SDCCH/8 channel of cell A
C_SDCCH8_A_3	LOGICCH	"C_SDCCH8_A_3"	3rd SDCCH/8 channel of cell A
C_SDCCH8_B_1	LOGICCH	"C_SDCCH8_B_1"	1st SDCCH/8 channel of cell B
C_SDCCH8_B_2	LOGICCH	"C_SDCCH8_B_2"	2nd SDCCH/8 channel of cell B
C_SDCCH80_A_1	LOGICCH	"C_SDCCH80_A_1"	1st SDCCH/8(0) of cell A
C_SDCCH81_A_1	LOGICCH	"C_SDCCH81_A_1"	1st SDCCH/8(1) of cell A
C_SDCCH82_A_1	LOGICCH	"C_SDCCH82_A_1"	1st SDCCH/8(2) of cell A
C_SDCCH83_A_1	LOGICCH	"C_SDCCH83_A_1"	1st SDCCH/8(3) of cell A
C_SDCCH84_A_1	LOGICCH	"C_SDCCH84_A_1"	1st SDCCH/8(4) of cell A
C_SDCCH85_A_1	LOGICCH	"C_SDCCH85_A_1"	1st SDCCH/8(5) of cell A
C_SDCCH86_A_1	LOGICCH	"C_SDCCH86_A_1"	1st SDCCH/8(6) of cell A
C_SDCCH87_A_1	LOGICCH	"C_SDCCH87_A_1"	1st SDCCH/8(7) of cell A
C_SDCCH80_A_2	LOGICCH	"C_SDCCH80_A_2"	2nd SDCCH/8(0) of cell A
C_SDCCH81_A_2	LOGICCH	"C_SDCCH81_A_2"	2nd SDCCH/8(1) of cell A
C_SDCCH82_A_2	LOGICCH	"C_SDCCH82_A_2"	2nd SDCCH/8(2) of cell A
C_SDCCH83_A_2	LOGICCH	"C_SDCCH83_A_2"	2nd SDCCH/8(3) of cell A
C_SDCCH84_A_2	LOGICCH	"C_SDCCH84_A_2"	2nd SDCCH/8(4) of cell A
C_SDCCH85_A_2	LOGICCH	"C_SDCCH85_A_2"	2nd SDCCH/8(5) of cell A
C_SDCCH86_A_2	LOGICCH	"C_SDCCH86_A_2"	2nd SDCCH/8(6) of cell A
C_SDCCH87_A_2	LOGICCH	"C_SDCCH87_A_2"	2nd SDCCH/8(7) of cell A
C_SDCCH80_A_3	LOGICCH	"C_SDCCH80_A_3"	3rd SDCCH/8(0) of cell A
C_SDCCH81_A_3	LOGICCH	"C_SDCCH81_A_3"	3rd SDCCH/8(1) of cell A
C_SDCCH82_A_3	LOGICCH	"C_SDCCH82_A_3"	3rd SDCCH/8(2) of cell A
C_SDCCH83_A_3	LOGICCH	"C_SDCCH83_A_3"	3rd SDCCH/8(3) of cell A
C_SDCCH84_A_3	LOGICCH	"C_SDCCH84_A_3"	3rd SDCCH/8(4) of cell A
C_SDCCH85_A_3	LOGICCH	"C_SDCCH85_A_3"	3rd SDCCH/8(5) of cell A
C_SDCCH86_A_3	LOGICCH	"C_SDCCH86_A_3"	3rd SDCCH/8(6) of cell A
C_SDCCH87_A_3	LOGICCH	"C_SDCCH87_A_3"	3rd SDCCH/8(7) of cell A
C_SDCCH80_B_1	LOGICCH	"C_SDCCH80_B_1"	1st SDCCH/8(0) of cell B
C_SDCCH81_B_1	LOGICCH	"C_SDCCH81_B_1"	1st SDCCH/8(1) of cell B
C_SDCCH82_B_1	LOGICCH	"C_SDCCH82_B_1"	1st SDCCH/8(2) of cell B
C_SDCCH83_B_1	LOGICCH	"C_SDCCH83_B_1"	1st SDCCH/8(3) of cell B
C_SDCCH84_B_1	LOGICCH	"C_SDCCH84_B_1"	1st SDCCH/8(4) of cell B
C_SDCCH85_B_1	LOGICCH	"C_SDCCH85_B_1"	1st SDCCH/8(5) of cell B
C_SDCCH86_B_1	LOGICCH	"C_SDCCH86_B_1"	1st SDCCH/8(6) of cell B
C_SDCCH87_B_1	LOGICCH	"C_SDCCH87_B_1"	1st SDCCH/8(7) of cell B
C_SDCCH80_B_2	LOGICCH	"C_SDCCH80_B_2"	2nd SDCCH/8(0) of cell B
C_SDCCH81_B_2	LOGICCH	"C_SDCCH81_B_2"	2nd SDCCH/8(1) of cell B
C_SDCCH82_B_2	LOGICCH	"C_SDCCH82_B_2"	2nd SDCCH/8(2) of cell B
C_SDCCH83_B_2	LOGICCH	"C_SDCCH83_B_2"	2nd SDCCH/8(3) of cell B
C_SDCCH84_B_2	LOGICCH	"C_SDCCH84_B_2"	2nd SDCCH/8(4) of cell B
C_SDCCH85_B_2	LOGICCH	"C_SDCCH85_B_2"	2nd SDCCH/8(5) of cell B
C_SDCCH86_B_2	LOGICCH	"C_SDCCH86_B_2"	2nd SDCCH/8(6) of cell B
C_SDCCH87_B_2	LOGICCH	"C_SDCCH87_B_2"	2nd SDCCH/8(7) of cell B
C_SDCCH80_B_3	LOGICCH	"C_SDCCH80_B_3"	3rd SDCCH/8(0) of cell B
C_SDCCH81_B_3	LOGICCH	"C_SDCCH81_B_3"	3rd SDCCH/8(1) of cell B
C_SDCCH82_B_3	LOGICCH	"C_SDCCH82_B_3"	3rd SDCCH/8(2) of cell B
C_SDCCH83_B_3	LOGICCH	"C_SDCCH83_B_3"	3rd SDCCH/8(3) of cell B
C_SDCCH84_B_3	LOGICCH	"C_SDCCH84_B_3"	3rd SDCCH/8(4) of cell B
C_SDCCH85_B_3	LOGICCH	"C_SDCCH85_B_3"	3rd SDCCH/8(5) of cell B
C_SDCCH86_B_3	LOGICCH	"C_SDCCH86_B_3"	3rd SDCCH/8(6) of cell B
C_SDCCH87_B_3	LOGICCH	"C_SDCCH87_B_3"	3rd SDCCH/8(7) of cell B
C_SDCCH80_C_1	LOGICCH	"C_SDCCH80_C_1"	1st SDCCH/8(0) of cell C
C_SDCCH81_C_1	LOGICCH	"C_SDCCH81_C_1"	1st SDCCH/8(1) of cell C
C_SDCCH82_C_1	LOGICCH	"C_SDCCH82_C_1"	1st SDCCH/8(2) of cell C
C_SDCCH83_C_1	LOGICCH	"C_SDCCH83_C_1"	1st SDCCH/8(3) of cell C
C_SDCCH84_C_1	LOGICCH	"C_SDCCH84_C_1"	1st SDCCH/8(4) of cell C
C_SDCCH85_C_1	LOGICCH	"C_SDCCH85_C_1"	1st SDCCH/8(5) of cell C

C_SDCCH86_C_1	LOGICCH	"C_SDCCH86_C_1"	1st SDCCH/8(6) of cell C
C_SDCCH87_C_1	LOGICCH	"C_SDCCH87_C_1"	1st SDCCH/8(7) of cell C
C_SDCCH80_C_2	LOGICCH	"C_SDCCH80_C_2"	2nd SDCCH/8(0) of cell C
C_SDCCH81_C_2	LOGICCH	"C_SDCCH81_C_2"	2nd SDCCH/8(1) of cell C
C_SDCCH82_C_2	LOGICCH	"C_SDCCH82_C_2"	2nd SDCCH/8(2) of cell C
C_SDCCH83_C_2	LOGICCH	"C_SDCCH83_C_2"	2nd SDCCH/8(3) of cell C
C_SDCCH84_C_2	LOGICCH	"C_SDCCH84_C_2"	2nd SDCCH/8(4) of cell C
C_SDCCH85_C_2	LOGICCH	"C_SDCCH85_C_2"	2nd SDCCH/8(5) of cell C
C_SDCCH86_C_2	LOGICCH	"C_SDCCH86_C_2"	2nd SDCCH/8(6) of cell C
C_SDCCH87_C_2	LOGICCH	"C_SDCCH87_C_2"	2nd SDCCH/8(7) of cell C
C_SDCCH80_C_3	LOGICCH	"C_SDCCH80_C_3"	3rd SDCCH/8(0) of cell C
C_SDCCH81_C_3	LOGICCH	"C_SDCCH81_C_3"	3rd SDCCH/8(1) of cell C
C_SDCCH82_C_3	LOGICCH	"C_SDCCH82_C_3"	3rd SDCCH/8(2) of cell C
C_SDCCH83_C_3	LOGICCH	"C_SDCCH83_C_3"	3rd SDCCH/8(3) of cell C
C_SDCCH84_C_3	LOGICCH	"C_SDCCH84_C_3"	3rd SDCCH/8(4) of cell C
C_SDCCH85_C_3	LOGICCH	"C_SDCCH85_C_3"	3rd SDCCH/8(5) of cell C
C_SDCCH86_C_3	LOGICCH	"C_SDCCH86_C_3"	3rd SDCCH/8(6) of cell C
C_SDCCH87_C_3	LOGICCH	"C_SDCCH87_C_3"	3rd SDCCH/8(7) of cell C
C_CHSDCCH8_FH	OCTETSTRING	'00'O	ch.type SDCCH8 with FH
C_CHSDCCH8_NonFH	OCTETSTRING	'01'O	ch.type SDCCH8 No FH
C_CHSDCCH4_NonFH	OCTETSTRING	'03'O	ch.type SDCCH4 NoFH
C_CHTCHF_FH	OCTETSTRING	'04'O	ch.type TCHF FH
C_CHTCHF_NonFH	OCTETSTRING	'05'O	ch.type TCHF NonFH
C_CHTCHH_FH	OCTETSTRING	'06'O	ch.type TCHH FH
C_TCHF_ACCHF_1	LOGCH	"C_TCHF_ACCHF_1"	first TCH/F + ACCHs type channel
C_TCHF_ACCHF_2	LOGCH	"C_TCHF_ACCHF_2"	second TCH/F + ACCHs type channel
C_TCHH_ACCHH_1	LOGCH	"C_TCHH_ACCHH_1"	first TCH/H + ACCHs type channel
C_TCHH_ACCHH_2	LOGCH	"C_TCHH_ACCHH_2"	second TCH/H + ACCHs type channel
C_FCCH_SCH_BCCH_CCCH	LOGCH	"C_FCCH_SCH_BCCH_CCCH"	FCCH_SCH_BCCH_CCCH type channel
C_BCCH_CCCH_2	LOGCH	"C_BCCH_CCCH_2"	second BCCH_CCCH type channel
C_BCCH_CCCH_3	LOGCH	"C_BCCH_CCCH_3"	third BCCH_CCCH type channel
C_BCCH_CCCH_4	LOGCH	"C_BCCH_CCCH_4"	fourth BCCH_CCCH type channel
C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4	LOGCH	"C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4"	combined CCCH type channel
C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4	LOGCH	"C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4"	combined CCCH type channel for cell broadcast SM
C_SDCCH8_SACCHC8_1	LOGCH	"C_SDCCH8_SACCHC8_1"	first SDCCH/8 type channel
C_SDCCH8_SACCHC8_2	LOGCH	"C_SDCCH8_SACCHC8_2"	second SDCCH/8 type channel
C_SDCCH8_SACCHC8_3	LOGCH	"C_SDCCH8_SACCHC8_3"	3rd SDCCH/8 type channel
C_S0	SN	'000'B	time slot 0
C_S2	SN	'010'B	time slot 2
C_S3	SN	'011'B	time slot 3
C_S4	SN	'100'B	time slot 4
C_S6	SN	'110'B	time slot 6
C_SAVE	INTEGER	0	the OC_SaveAndProc saves input values
C_PROC	INTEGER	1	the OC_SvaAndProc processes saved values
C_RETRV	INTEGER	1	
C_U1	CCSTATE	1	CC state U1
C_U3	CCSTATE	3	CC state U3
C_U4	CCSTATE	4	CC state U4
C_U6	CCSTATE	6	CC state U6
C_U7	CCSTATE	7	CC state U7
C_U8	CCSTATE	8	CC state U8
C_U9	CCSTATE	9	CC state U9
C_U10	CCSTATE	10	CC state U10
C_U11	CCSTATE	11	CC state U11
C_U12	CCSTATE	12	CC state U12
C_U19	CCSTATE	19	CC state U19

C_U26	CCSTATE	26	CC state U26
C_NxtButOne	SENDINGMODE	1	send the second message on the next but one paging subblock
C_FmrAGB	SENDINGMODE	2	send the second message on the former access grant block
C_BfReOcc	SENDINGMODE	3	send the second message before the MS's original paging subchannel re-occurs, but later than the next paging block of that CCCH
C_NxtButOneNxt	SENDINGMODE	4	nothing is sent in the next but one paging sub block, then send the second message in the next paging subblock of the MS's paging subchannel
C_CMServiceTypeE	CMSVTYPE	'0010'B	CM Service Type for emergency call.
C_Telephony	IA5String	"C_Telephony"	telephony service (TS11)
C_EmgCall	IA5String	"C_EmgCall"	emergency call service (TS12)
C_AltSpchFax	IA5String	"C_AltSpchFax"	alternate speech and G3 fax service (TS61)
C_AutoFax	IA5String	"C_AutoFax"	automatic G3 fax service (TS62)
C_Async300	IA5String	"C_Async300"	data circuit duplex async. 300 bit/s service (BS21)
C_Async1200	IA5String	"C_Async1200"	data circuit duplex async. 1200 bit/s service (BS22)
C_Async120075	IA5String	"C_Async120075"	data circuit duplex async. 1200/75 bit/s service (BS23)
C_Async2400	IA5String	"C_Async2400"	data circuit duplex async. 2400 bit/s service (BS24)
C_Async4800	IA5String	"C_Async4800"	data circuit duplex async. 4800 bit/s service (BS25)
C_Async9600	IA5String	"C_Async9600"	data circuit duplex async 9600 bit/s service (BS26)
C_Sync1200	IA5String	"C_Sync1200"	data circuit duplex sync. 1200 bit/s service (BS31)
C_Sync2400	IA5String	"C_Sync2400"	data circuit duplex sync. 2400 bit/s service (BS32)
C_Sync4800	IA5String	"C_Sync4800"	data circuit duplex sync. 4800 bit/s service (BS33)
C_Sync9600	IA5String	"C_Sync9600"	data circuit duplex sync 9600 bit/s service (BS34)
C_PAD300	IA5String	"C_PAD300"	PAD access 300 bit/s service (BS41)
C_PAD1200	IA5String	"C_PAD1200"	PAD access 1200 bit/s service (BS42)
C_PAD120075	IA5String	"C_PAD120075"	PAD access 1200/75 bit/s service (BS43)
C_PAD2400	IA5String	"C_PAD2400"	PAD access 2400 bit/s service (BS44)
C_PAD4800	IA5String	"C_PAD4800"	PAD access 4800 bit/s service (BS45)
C_PAD9600	IA5String	"C_PAD9600"	PAD access 9600 bit/s service (BS46)
C_Packet2400	IA5String	"C_Packet2400"	packt access 2400 bit/s service (BS51)
C_Packet4800	IA5String	"C_Packet4800"	packt access 4800 bit/s service (BS52)
C_Packet9600	IA5String	"C_Packet9600"	packt access 9600 bit/s service (BS53)
C_AltSpchData	IA5String	"C_AltSpchData"	alternate speech/data service (BS61)
C_SpchData	IA5String	"C_SpchData"	speech followed data service (BS81)
C_I	INTEGER	1	I command of L 2
C_one	INTEGER	1	one frequency, no hopping
C_Immass	BITSTRING	'000'B	activation for immediate assignment
C_Ass	BITSTRING	'001'B	activation for assignment
C_Asynho	BITSTRING	'010'B	activation for non synchronous handover

C_Synho	BITSTRING	'011'B	activation for synchronous handover
C_Rcv	BITSTRING	'101'B	activation for receiving only
C_LocEndRel	RelMode	'01'O	local end release
C_MaxPwrLvlD	INTEGER	15	Max power level for MS DCS
C_MaxPwrLvlG	INTEGER	19	Max power level for MS GSM
C_rc_conditIEerror	REJCAU	'64'O	reject cause: Conditional IE error
C_rc_congestion	REJCAU	'16'O	reject cause: congestion
C_rc_illegal_ms	REJCAU	'03'O	reject cause: illegal MS
C_rc_illegal_me	REJCAU	'06'O	reject cause: illegal ME
C_rc_imsiunknownhlr	REJCAU	'02'O	reject cause: IMSI unknown in HLR
C_rc_imsiunknownvlr	REJCAU	'04'O	reject cause: IMSI unknown in VLR
C_rc_invalidmaninfo	REJCAU	'60'O	reject cause: invalid mandatory information
C_rc_LAnotallowed	REJCAU	'0C'O	reject cause: LocationArea is not allowed
C_rc_networkfailure	REJCAU	'11'O	reject cause: network failure
C_rc_notidentified	REJCAU	'26'O	reject cause: can not be identified
C_rc_plmn_not	REJCAU	'0B'O	reject cause: PLMN not allowed
C_rc_protocolerror	REJCAU	'6F'O	reject cause: Protocol error unspecified
C_rc_reqservoptnotsub	REJCAU	'21'O	reject cause: requested service option not subscribed
C_rc_roamingnot	REJCAU	'0D'O	reject cause: Roaming not allowed
C_Restablishment	B_1	'0'B	Restablishment allowed
C_noRestablishment	B_1	'1'B	No restablishment allowed
C_SlotNU	SN	'111'B	Slot Number not used
C_CellReselectHys14	INTEGER	7	Cell Reselection Hysteresis value 14db
C_CellReselectHys12	INTEGER	6	Cell Reselection Hysteresis value 12db
C_CellReselectHys10	INTEGER	5	Cell Reselection Hysteresis value 10db
C_CellReselectHys4	INTEGER	2	Cell Reselection Hysteresis value 4db
C_RI_alternate	B_8	'11010001'B	Bcap repeat indicator for alternate (circular selection)
C_RI_follow	B_8	'11010011'B	Bcap repeat indicator for successive (sequential selection)
C_SIMIn	BOOLEAN	FALSE	SIM is always inserted in a test case
C_SIMneedRmv	BOOLEAN	TRUE	Removal of SIM is needed in a test case (SIM is not always inserted.)
C_Yes	BOOLEAN	TRUE	
C_No	BOOLEAN	FALSE	
C_Start_Imsi	INTEGER	1	Test case body starts without TMSI
C_Start_Tmsi	INTEGER	2	Test case body starts with Tmsi_01
C_Start_TmsiOff	INTEGER	5	Test case body starts with Tmsi_01. MS has been deactivated.
C_Start_Rej	INTEGER	3	Test case body starts with LAI deleted. MS has been deactivated.
C_Start_PLMNnot	INTEGER	4	Test case body starts with PLMN

			not allowed
C_Test_fh	INTEGER	1	Test case with possibly frequency hopping
C_Test_nfh	INTEGER	2	Test case with no frequency hopping
C_Test_egsm	INTEGER	3	Test case for egsm
C_Test_cbms	INTEGER	4	Test case for CBMS
C_normal_updating	B_2	'00'B	normal location updating
C_periodic_updating	B_2	'01'B	periodic location updating
C_normalOrperiodic	B_2	'0?'B	periodic or normal location updating
C_imsi_attach	B_2	'10'B	location updating IMSI attach
C_norm_period_attach	B_2	'??'B	imsi attach or periodic location updating or normal location updating
C_normal_paging	B_2	'00'B	normal page mode
C_extended_paging	B_2	'01'B	extended paging
C_reorg_paging	B_2	'10'B	paging reorganisation
C_sab_paging	B_2	'11'B	same as before paging
C_non_synchronised	B_2	'00'B	non-synchronised
C_synchronised	B_2	'01'B	synchronised
C_pre_synchronised	B_2	'10'B	pre-synchronised
C_pseudo_synchronised	B_2	'11'B	pseudo-synchronised
C_not_report_otd	B_1	'0'B	not report observed time difference
C_report_otd	B_1	'1'B	report observed time difference
C_Tzone0	TZONES	0	Time zone 0 (used in TC_34_2_1, TC_34_2_5_3)
C_Tzone1	TZONES	4	Time zone 4 (used in TC_34_2_2, TC_34_2_7)
C_Tzone2	TZONES	10	Time zone 10 (used in TC_34_2_3)
C_Tzone3	TZONES	15	Time zone 15 (used in TC_34_2_4)
C_Tzone4	TZONES	25	Time zone 25 (used in TC_34_2_5_1)
C_Tzone5	TZONES	35	Time zone 35 (used in TC_34_2_5_2)
C_Tzone6	TZONES	45	Time zone 45 (used in TC_34_2_8)
C_T_AssCmd	INTEGER	900	600ms + 300ms(tolerance)
C_T_Wait1stChReq	INTEGER	35000	max time for full BCCH data decoding
C_T_mrsrp	INTEGER	1500	3 times time btw 2 msr rpt.
C_T_T3124	INTEGER	675	T3124(675ms)
C_T_T3211plus	INTEGER	20000	20s
C_T_T3212	INTEGER	360000	T3212(6min)
C_T_T3212min	INTEGER	345000	T3212(6min) - 15sec
C_T_T3212dif	INTEGER	15000	Difference between T3212 and T3212min
C_T_T3210	INTEGER	20000	timer T3210 in ms
C_T_T3211min	INTEGER	13500	90% T3211 in ms
C_T_T3230min	INTEGER	13500	90% T3230 in ms
C_T_T3240min	INTEGER	9000	90% T3240 in ms
C_T_T3240tol	INTEGER	2000	20% T3240 in ms
C_T_Wait	INTEGER	5000	5s. T3126max
C_T_Wait1	INTEGER	1000	1s
C_T_Wait_FAC	INTEGER	1500	1.5s Facility response time (1s) + tolerance (0.5s)
C_release_time	INTEGER	11	release timer value
C_RegCFNRy	INTEGER	1	registration of call forwarding service for CFNRy (speech)
C_RegCFU	INTEGER	2	registration of call forwarding service for CFU (all facsimile)
C_RegCF	INTEGER	3	registration of call forwarding service for CF (all facsimile)
C_ErsCFC	INTEGER	4	erasure of call forwarding service for CFC (all facsimile)

C_ErsCFNRc	INTEGER	5	erasure of call forwarding service for CFNRc (all basic services)
C_ErsCFNRy	INTEGER	6	erasure of call forwarding service for CFNRy (all facsimile)
C_ActCF	INTEGER	7	activation of call forwarding service for CF (all synchronous services)
C_ActCFU	INTEGER	8	activation of call forwarding service for CFU (all basic services)
C_DeactCFC	INTEGER	9	deactivation of call forwarding service for CFC (speech)
C_DeactCFNRc	INTEGER	10	deactivation of call forwarding service for CFNRc (all facsimile)
C_InterrogCFC	INTEGER	11	interrogation of call forwarding service for CFC (Speech)
C_InterrogCFB	INTEGER	12	interrogation of call forwarding service for CFB (all basic services)
C_NotifyCFB	INTEGER	13	notification of call forwarding service for CFB (incoming call is forwarded)
C_NotifyCFU	INTEGER	14	notification of call forwarding service for CFU (provisioned, registered, active)
C_NotifyCFC	INTEGER	15	notification of call forwarding service for CFC (provisioned, registered, active)
C_NotifyCFNRc	INTEGER	16	notification of call forwarding service for CFNRc (Ms not reachable)
C_RegPswd	INTEGER	17	Registration of password for all call barring services
C_ActBOAC	INTEGER	18	Activation of BOAC
C_ActBICRoam	INTEGER	19	Activation of BICRoam
C_ActBOIC	INTEGER	20	Activation of BOIC
C_ActBAIC	INTEGER	21	Activation of BAIC
C_DeactBO	INTEGER	22	Deactivation of BO
C_DeactBI	INTEGER	23	Deactivation of BI
C_DeactBOICExHC	INTEGER	24	Deactivation of BOICExHC
C_NotifyBI	INTEGER	25	Notify of BI
C_InterrogBOIC	INTEGER	26	Interrogation of BOIC
C_InterrogBOICExHC	INTEGER	27	Interrogation of BOICExHC
C_Full	IA5String	"F"	full rate channel
C_Half	IA5String	"H"	half rate channel
C_nirr_dontcare	B_1	'?'B	NIRR no meaning or requested
C_Rchr_dontcare	B_2	'??'B	radio channel requirement not checked
C_Rchr_Full	B_2	'01'B	radio channel requirement : full rate
C_Rchr_DualForDualH	B_2	'1?'B	radio channel requirement : dual rate support/half rate preferred or dual rate support/full rate preferred
C_Rchr_DualForDualH0	B_2	'10'B	radio channel requirement : dual rate support/half rate preferred or dual rate support/full rate preferred
C_StartingTimeHO	INTEGER	238	Starting time 238 (1.1s), in TC_26_6_5_1, TC_26_6_5_4_2
C_StrT_1000	INTEGER	1000	Starting time 1000, in TC_26_6_13_1
C_Norm	OCTETSTRING	'00'O	Establish mode: normal
C_Speech	B_3	'000'B	Info. Transfer Cap. speech
C_3100Hz	B_3	'010'B	Info. Transfer Cap. 3.1 kHz
C_UDI	B_3	'001'B	Info. Transfer Cap. UDI
C_FAX3	B_3	'011'B	Info. Transfer Cap. FAX3
C_SDUintegrity	B_2	'00'B	Structure SDU integrity
C_Unstructured	B_2	'11'B	Structure Unstructured
C_struc_dontchk	B_2	'??'B	Structure not checked

C_nirr_nomeaning	B_1	'0'B	NIRR no meaning
C_no_rate_adaption	B_2	'00'B	Rate adaption no
C_rate_adaption_V110	B_2	'01'B	Rate adaption V110
C_rate_adaption_X31	B_2	'10'B	Rate adaption X31 flag
C_sacp_NA	B_3	'001'B	Signaling access NA (sec.11.8.1.5.1)
C_I440_450	B_3	'001'B	Signaling access I440/450
C_X21	B_3	'010'B	Signaling access X21
C_X28_unui	B_3	'100'B	Signaling access X28 U NUI
C_X28_nond	B_3	'101'B	Signaling access X28 non dedicated
C_X32	B_3	'110'B	Signaling access X32
C_Synchronous	B_1	'0'B	Synchronous
C_Asynchronous	B_1	'1'B	Asynchronous
C_stopbit_NA	B_1	'0'B	Number of Stop bits NA
C_stopbit_dontcare	B_1	'?'B	Number of Stop bits not care
C_databit_NA	B_1	'1'B	Number of Data bits not care
C_databit_dontcare	B_1	'?'B	Number of Data bits NA
C_300bs	B_4	'0001'B	User rate 300 bs
C_1200bs	B_4	'0010'B	User rate 1200 bs
C_120075bs	B_4	'0111'B	User rate 1200/75 bs
C_2400bs	B_4	'0011'B	User rate 2400 bs
C_4800bs	B_4	'0100'B	User rate 4800 bs
C_9600bs	B_4	'0101'B	User rate 9600 bs
C_parity_NA	B_3	'011'B	Parity NA
C_parity_dontcare	B_3	'???'B	Parity not care
C_ir_8kbs	B_2	'10'B	Intermediate rate 8 Kbs
C_ir_16kbs	B_2	'11'B	Intermediate rate 16 Kbs
C_ir_8or16kbs	B_2	'1?'B	Intermediate rate 8 or 16 Kbs
C_transparent	B_2	'00'B	Connection element T
C_notransparent	B_2	'01'B	Connection element NT
C_BothT	B_2	'10'B	Connection element both T
C_BothNT	B_2	'11'B	Connection element both NT
C_modemt_none	B_5	'00000'B	no modem
C_modemt_V21	B_5	'00001'B	modem V21
C_modemt_V22	B_5	'00010'B	modem V22
C_modemt_V22bis	B_5	'00011'B	modem V22bis
C_modemt_V23	B_5	'00100'B	modem V23
C_modemt_V32	B_5	'00110'B	modem V32
C_ISO6429	B_5	'01000'B	Inband flow control
C_COPnoFLCT	B_5	'01100'B	No flow control
C_X25_flct	B_5	'00110'B	X25 flow control
C_Outband	FLWCNTL	0	outband flow control
C_Inband	FLWCNTL	1	outband flow control
C_Nocontrol	FLWCNTL	2	outband flow control
C_uil2p_dontcare	B_5	'?????'B	not checked
C_HSN_0	INTEGER	0	cyclic hopping
maxAddressLength	Asn1Integer	20	
maxISDN_AddressLength	Asn1Integer	9	
maxISDN_SubaddressLength	Asn1Integer	21	
maxNumOfBasicServiceGroups	Asn1Integer	13	
maxNumOfCUG	Asn1Integer	10	
maxNumberOfSegmentsPerDataInterval	Asn1Integer	8191	
maxSignalInfoLength	Asn1Integer	200	
maxUSSD_StringLength	Asn1Integer	160	
max10TimesIncrement	Asn1Integer	8191	
max10TimesIncrementPerDataInterval	Asn1Integer	8191	
max10TimesInitialTime	Asn1Integer	8191	
max100TimesScalingFactor	Asn1Integer	8191	
max10TimesTimeInterval	Asn1Integer	8191	
max10TimesUnitsPerTime	Asn1Integer	8191	
Detailed Comments:			

Test case variable declarations

Test Case Variable Declarations			
Variable Name	**Type**	**Value**	**Comments**
TCV_CC	BOOLEAN	TSPC_Serv_TS11 OR TSPC_Serv_TS12 OR TSPC_Serv_TS61 OR TSPC_Serv_TS62 OR TSPC_Serv_BS21 OR TSPC_Serv_BS22 OR TSPC_Serv_BS23 OR TSPC_Serv_BS24 OR TSPC_Serv_BS25 OR TSPC_Serv_BS26 OR TSPC_Serv_BS31 OR TSPC_Serv_BS32 OR TSPC_Serv_BS33 OR TSPC_Serv_BS34 OR TSPC_Serv_BS41 OR TSPC_Serv_BS42 OR TSPC_Serv_BS43 OR TSPC_Serv_BS44 OR TSPC_Serv_BS45 OR TSPC_Serv_BS46 OR TSPC_Serv_BS51 OR TSPC_Serv_BS52 OR TSPC_Serv_BS53 OR TSPC_Serv_BS61 OR TSPC_Serv_BS81	
TCV_AssCmd	ASS_CMD_PDU		to hold ASSIGNMENT COMMAND PDU
TCV_Bcap1	BCAP		to hold bearer capability
TCV_BcapMT1	BCAP		to hold BC for MT call
TCV_BcapMO1	BCAP		to hold BC for MO call
TCV_Bcap2	BCAP		to hold bearer capability
TCV_BcapMT2	BCAP		to hold BC for MT call
TCV_bs_ag_blks_res	INTEGER		
TCV_bs_cc_chans	INTEGER		
TCV_bs_pa_mfrms	INTEGER		
TCV_CalledNum	CDPN		to hold called party number
TCV_CallProc	CALL_PROC_PDU		to hold CALL PROCEEDING PDU
TCV_CallCfm	CALL_CO_PDU		to hold CALL COMFIRM PDU
TCV_Cau	CAU		to hold cause
TCV_Cau0	CAU		to hold cause
TCV_CAgsm	CCHD		To compute Cell Channel Descriptor for gsm
TCV_CAdcs	CCHD		To compute Cell Channel Descriptor for dcs
TCV_CB1	OCTETSTRING		store SMSCB block data
TCV_CB2	OCTETSTRING		store SMSCB block data
TCV_CB3	OCTETSTRING		store SMSCB block data
TCV_CB4	OCTETSTRING		store SMSCB block data
TCV_CE	CE		
TCV_Ccchg	INTEGER		CCCH_GROUP
TCV_Ccd0A	CCD		a copy of control channel description in use for cell A
TCV_Ccd0B	CCD		a copy of control channel description in use for cell B
TCV_Ccd0C	CCD		a copy of control channel description in use for cell C
TCV_Ccd0D	CCD		a copy of control channel description in use for cell D
TCV_Ccd0E	CCD		a copy of control channel description in use for cell E
TCV_Ccd0F	CCD		a copy of control channel description in use for cell F
TCV_Ccd0G	CCD		a copy of control channel description in use for cell G
TCV_Ccd0H	CCD		a copy of control channel description in use for cell H

TCV_cchdescr	OCTETSTRING		Frequency list of Cell Channel Descr, used in EGSM
TCV_CCst	CST		to hold call status
TCV_ch	LOGICCH	"dummy"	to hold logic channel
TCV_ch1	LOGICCH	"dummy"	to hold logic channel
TCV_sacch	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_B	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_C	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_D	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_E	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_F	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_G	LOGICCH	"dummy"	to hold logic channel
TCV_sacch_H	LOGICCH	"dummy"	to hold logic channel
TCV_sacch8	LOGICCH	"dummy"	to hold logic channel
TCV_sacchTch	LOGICCH	"dummy"	to hold logic channel
TCV_sacchTch1	LOGICCH	"dummy"	to hold logic channel
TCV_sacchTch_H	LOGICCH	"dummy"	to hold logic channel
TCV_chTch	LOGICCH	"dummy"	to hold logic channel
TCV_chTch1	LOGICCH	"dummy"	to hold logic channel
TCV_chSms	LOGICCH	"dummy"	to hold logic channel
TCV_agch	LOGICCH	"dummy"	downlink access grant channel
TCV_WorkingCh	LOGICCH	"dummy"	to hold logic channel
TCV_cellid	CellID		to hold the cell identifier
TCV_Char	IA5String		to hold the IA5 char. corresponding to a DTMF tone
TCV_chd1	CHD		to hold channel description IE
TCV_cchd1	CCHD		to hold cell channel description IE
TCV_ChMod	CHMOD		to hold channel mode IE
TCV_ChModb	CHMOD		to hold channel mode IE for second mode in dual mode services
TCV_Tchtype	CH_TDMA	'00001'B	Traffic channel type and TDMA offset in channel description IE
TCV_chtype	CH_TDMA		to channel type and TDMA offset in channel description IE
TCV_cksn	BITSTRING		to hold the CKSN.
TCV_Cnt	INTEGER		general loop counter
TCV_Cnt1	INTEGER		general loop counter
TCV_Cnt2	INTEGER		general loop counter
TCV_Cntstart	BOOLEAN	FALSE	general loop counter start decision
TCV_Cntref	INTEGER		Reference for general loop counter
TCV_CphAlg	BITSTRING		ciphering algorithm
TCV_CphKey	BITSTRING		ciphering key
TCV_CphMd	CPHMS		ciphering mode setting
TCV_CPDataRetx	INTEGER		Number of CP Data retransmissions for SMS
TCV_ChRate	IA5String	C_Full	to hold channel rate
TCV_ChRate1	IA5String	C_Full	to hold channel rate
TCV_counter_c	INTEGER		loop counter c
TCV_counter_k	INTEGER		loop counter k
TCV_Fk	INTEGER		to hold an interval between events
TCV_flist	OCTETSTRING		Frequency List, used in EGSM for IE "frequency list" or "frequency short list"
TCV_flistl	LENGTH		Length of Frequency List, used in EGSM for IE "frequency list" or "frequency short list"
TCV_Fn	FN		to hold the frame number (T1' , T2, T3)
TCV_Fn1	FN		to hold the frame number (T1' , T2, T3)
TCV_FnAss	FN		to hold the frame number in AssCmd statement
TCV_FollowingOctets	OCTETSTRING		Contains the octets following the invoke id
TCV_Horf	HORF		Variable for HO reference in HO-

			Messages of HO cases.
TCV_HoCMD	HO_CMD_PDU		to hold handover PDU
TCV_Hrf	HORF		to hold handover reference
TCV_iel	LENGTH		Length of IE
TCV_kcnt	INTEGER		loop counter
TCV_K	INTEGER		
TCV_L1Head0	L1HD		to hold L 1 header
TCV_L1Head	L1HD		to hold L 1 header
TCV_lac	OCTETSTRING		to hold the lac. 2 octets
TCV_lacunknown	OCTETSTRING	'??'O	to hold a wild card lac when the lac is unknown
TCV_n	INTEGER		
TCV_Neci	BITSTRING		
TCV_M	INTEGER		counter
TCV_maio	MAIO		to hold MAIO
TCV_mae1	BITSTRING		Mobile Allocation in EGSM. It includes the 1.octet of MA
TCV_mae2	BITSTRING		Mobile Allocation in EGSM. It includes the 2.octet of MA
TCV_Max	BITSTRING		to hold Max-retrans bitstring
TCV_MemCapExcd	BOOLEAN		RP error Memory Capacity Exceeded was sent
TCV_MsrRes	MSRR		to hold measurement results
TCV_Mt	BITSTRING		to hold message type
TCV_Mt1	BITSTRING		to hold message type
TCV_Modify	MODIFY_PDU		To hold Modify message
TCV_Null	BOOLEAN		to collect useless result from some test suite operations
TCV_PgCh	LOGICCH	"dummy"	logical channel for paging
TCV_Pgg	PGG		paging group
TCV_PO	B_2		power offset
TCV_PreviousOctets	OCTETSTRING		Contains the octets previous to the invoke id
TCV_ProtErrorUnspec	BOOLEAN		RP error Protocol Error Unspecified was sent
TCV_Pwrlvl	BITSTRING		to hold the power level
TCV_Pwrlvl_ho	BITSTRING		power level for HO_CMD, used in HO cases.
TCV_RaCh	LOGICCH		random access channel parameter.
TCV_Rchr	B_2		to hold Radio channel requirement
TCV_ReceivedRACHs	INTEGER		to hold the number of RACHs currently received
TCV_Res	BOOLEAN		to hold the result of a test suite operation
TCV_Rpmr	MR		RP message reference for SMS
TCV_Rqr	RQR		to hold the request reference
TCV_Rqr9	RQR		to hold the request reference differing from TCV_Rqr
TCV_Rqr10	RQR		to hold the request reference differing from TCV_Rqr, TCV_Rqr9
TCV_Rqr11	RQR		to hold the request reference differing from TCV_Rqr, TCV_Rqr9, TCV_Rqr10
TCV_Rr	BITSTRING		to hold the random request reference
TCV_Rr1	BITSTRING		to hold the random reference
TCV_Service	SERVICES		to hold basicservice selected
TCV_Service1	SERVICES		to hold basicservice selected
TCV_Setup_mo	SETUP_MO_PDU		to hold the SETUP PDU (MO)
TCV_Setup_mo1	SETUP_MO_PDU		to hold the SETUP PDU (MO)
TCV_Setup_mt	SETUP_MT_PDU		to hold the SETUP PDU (MT)
TCV_Setup_mt1	SETUP_MT_PDU		to hold the 2nd SETUP PDU (MT)
TCV_2ndtest	BOOLEAN		2nd test required in TC_11_1_1
TCV_Esetup	ESETUP_PDU		to hold the emergency SETUP PDU

TCV_Esetup1	ESETUP_PDU		to hold the emergency SETUP PDU
TCV_Ecall	BOOLEAN	FALSE	it is set to TRUE, if the service under test is emergency call
TCV_slot	SN		To hold the default slot used during the entire TC.
TCV_slt2	SN		To hold a second working slot number.
TCV_Sres	OCTETSTRING		to hold the SRES returned
TCV_sysinf5	SYSINFO5_PDU		to hold the SysInfo5 PDU
TCV_sysinfo5	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell A
TCV_sysinfo5bis	SYSINFO5bis_PDU		to hold the SysInfo5bis PDU for cell A
TCV_sysinfo5_B	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell B
TCV_sysinfo5_C	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell c
TCV_sysinfo5_D	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell D
TCV_sysinfo5_E	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell E
TCV_sysinfo5_F	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell F
TCV_sysinfo5_G	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell G
TCV_sysinfo5_H	SYSINFO5_PDU		to hold the SysInfo5 PDU for cell H
TCV_sysinfo5bis_A	SYSINFO5bis_PDU		to hold the SysInfo5bis PDU for cell A(TC_26_10_2_4_1)
TCV_sysinfo5bis_B	SYSINFO5bis_PDU		to hold the SysInfo5bis PDU for cell B
TCV_sysinf6	SYSINFO6_PDU		to hold the SysInfo6 PDU
TCV_sysinfo6	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell A
TCV_sysinfo6_B	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell B
TCV_sysinfo6_C	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell C
TCV_sysinfo6_D	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell D
TCV_sysinfo6_E	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell E
TCV_sysinfo6_F	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell F
TCV_sysinfo6_G	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell G
TCV_sysinfo6_H	SYSINFO6_PDU		to hold the SysInfo6 PDU for cell H
TCV_temp	INTEGER		to hold results of expressions that are to be passed as parameters
TCV_ti_v	TI_V		to hold the transaction id. value
TCV_ti_v_2	TI_V		to hold the transaction id. value for another transaction
TCV_T	INTEGER		to hold Tx-integer chosen randomly
TCV_T_TwiceTC1M	INTEGER	TSPX_TC1M * 2000	Twice TC1M in ms
TCV_chdescr_arfcn	INTEGER		ARFCN carrier of actual cell
TCV_tch_arfcn	INTEGER		ARFCN carrier for traffic channel
TCV_asscmd_ts	BITSTRING		Time slot for channel description in assignment cmd.
TCV_ts	BITSTRING		Time slot for channel description in handover cmd.
TCV_ia_ts	BITSTRING		Time slot for channel description in immediate assignment
TCV_tsc	TSC		To hold the default training sequence used during the entire TC.
TCV_Td	OCTETSTRING		to hold timing difference
TCV_Time	INTEGER		to hold the measured value of

			T310 or T305 or T308
TCV_TI	TI		to hold the transaction ID for MO calls in sending evets
TCV_TI0	TI		to hold the transaction ID for MO calls in receiving evets
TCV_TI1	TI		to hold the transaction ID
TCV_TI2	TI		to hold the transaction ID
TCV_TI3	TI		to hold the transaction ID
TCV_tmp	INTEGER		temporary integer variable
TCV_Tx	BITSTRING		to hold Tx bitstring
TCV_S	INTEGER		to hold the S parameter
TCV_Upd	BOOLEAN		to hold the information whether the MS is updated or not,.
TCV_UssdString	IA5String		String for USS Data
TCV_UssdString1	IA5String		String for USS Data
TCV_UssdString2	IA5String		String for USS Data
TCV_InvkId	OCTETSTRING		to hold SS transaction ID
TCV_InvkId0	OCTETSTRING		to hold SS transaction ID
TCV_InvkId1	OCTETSTRING		to hold SS transaction ID
TCV_Comp	Components		to hold SS Components
TCV_Strt	STRT		to hold starting time
TCV_CBch	LOGICCH	"dummy"	Cell Broadcast channel
TCV_Tpmr	MR		TP message reference
TCV_TPOA1	BCDN		TP originating address digits
TCV_TPOA2	BCDN		TP originating address digits
TCV_TPDA	BCDN		TP destination address digits
TCV_RPOA_MT	BCDN		RP originating address digits for MT short messages
TCV_RPOA_MO	BCDN		RP originating address digits for MO short messages
TCV_RPOA1	BCDN		RP originating address digits for MT short messages
TCV_RPOA2	BCDN		RP originating address digits for MT short messages
TCV_RPDA_MT	BCDN		RP destination address digits for MT short messages
TCV_RPDA_MO	BCDN		RP destination address digits for MO short messages
TCV_SMTypeM	INTEGER		Replace short message type
TCV_SMTypeN	INTEGER		Replace short message type
TCV_SMcntns	OCTETSTRING		short message contents
TCV_SMSCBpack	SMSCBpack		CB message contents
TCV_freq	FRQPARA		to hold Freq type constraints
TCV_srv	SERVICES		to hold MT service
TCV_ur	B_4		to hold user rate
TCV_itc	B_3		to hold information transfer capability
TCV_sacp	B_3		to hold service access protocol
TCV_ce	B_2		to hold connection element
TCV_ce1	B_2		to hold connection element
TCV_ce3	B_2		to hold connection element
TCV_sa	B_1		to hold synchronous/asynchronous
TCV_ir	B_2		to hold intermediate rate
TCV_uil1p	B_2		to hold user information layer 1 protocol
TCV_modemt	B_5		to hold modem type
TCV_MODEM	MODEMTYPE		
TCV_ur2	B_4		to hold user rate for second configuration
TCV_itc2	B_3		to hold information transfer capability for second configuration
TCV_sacp2	B_3		to hold service access protocol for second configuration
TCV_strc	B_2		to hold structure
TCV_sa2	B_1		to hold synchronous/asynchronous for second configuration

TCV_supported	BOOLEAN	FALSE	used for indicating a selected service is declared supported or not
TCV_SpecialCase	BOOLEAN	FALSE	used for indicating whether a test case tests non-supported service
Detailed Comments:			

PCO declarations

PCO Declarations			
PCO Name	**Type**	**Role**	**Comments**
L	SAP0_3	LT	
Detailed Comments:	1. The PCO consists of multiple SAPs: the SAP0 and SAP3. 2. The lower tester (LT) is the user of the data link layer service. 3. The SAP0 at the lower tester controlling and observing the exchange of CC, MM, RR and SS PDUs (messages) on the DCCH, SACCH and/or RACH, BCCH, CCCH channels. 4. The SAP3 at the lower tester controlling and observing the exchange of SMS PDUs (messages) on the DCCH or SACCH channels.		

Timer declarations

Timer Declarations			
Timer Name	**Duration**	**Unit**	**Comments**
T_dly		ms	general purpose delay timer
T_dly1		ms	general purpose delay timer
T_dly2		ms	general purpose delay timer
T_dlyAss	5000	ms	AssCmd timer, 1)
T_guard	300	s	guard timer
T_release	C_release_time	s	release timer
T_pause		ms	pause timer
Detailed Comments:	1) In order to give to the lower tester enough time to dequeue and to send out the call SETUP and the other call establishment related messages in the sending buffer before starting sending of ASSINGMENT COMMAND message, the default value 5s is used in the test step AssCh_complete. In the test step AssCh_failure the timer value is set to 4s. For the TC_26_6_4_1, 600ms is used.		

ASP type definitions

TTCN ASP Type definitions

ASP Type Definition		
ASP Name:	DL_EstRq (DL_ESTABLISH_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the establishment of multiple frame operation (L3 -> L2). The normal establishment procedure is initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstIn (DL_ESTABLISH_INDICATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation (L2 -> L3). The normal establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInCmsRq (DL_ESTABLISH_INDICATION_CM_SERVICE_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the MM CM SERVICE REQUEST message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	CMS_RQ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInLupRq (DL_ESTABLISH_INDICATION_LOCATION_UPDATING_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the MM LOCATION UPDATING REQUEST message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	LUP_RQ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInImsidIn (DL_ESTABLISH_INDICATION_IMSI_DETACH_INDICATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the MM IMSI DETACH INDICATION message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	IMSID_IN_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInPgRes (DL_ESTABLISH_INDICATION_PAGING_RESPONSE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the RR PAGING RESPONSE message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	PG_RES_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstInCmreRq (DL_ESTABLISH_INDICATION_CM_REESTABLISHMENT_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the establishment of multiple frame operation due to the receipt of the MM CM REESTABLISHMENT REQUEST message (L2 -> L3). The contention resolution establishment procedure has been initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
msg	CMRE_RQ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_EstCo (DL_ESTABLISH_CONFIRM)	
PCO Type:	SAP0_3	
Comments:	The ASP is used by the L2 to inform the L3 about the establishment of multiple frame link (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
establish_mode	EstMode	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_RelIn (DL_RELEASE_INDICATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the termination of an established multiple frame operation or to report an unsuccessful establishment attempt (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
release_mode	RelMode	
outstanding_indicator	BOOLEAN	
fn	FN	
Detailed Comments:	The outstanding indicator indicates whether or not there are outstanding acknowledgements or unsolved DL_DATA_REQUEST primitives.	

ASP Type Definition		
ASP Name:	DL_RacInChRq (DL_RANDOM_ACCESS_INDICATION_CHANNEL_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the arrival of an RR CHANNEL REQUEST message (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
fn	FN	
msg	CH_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_RacInHoacc (DL_RADOM_ACCESS_INDICATION_HANDOVER_ACCESS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the arrival of an RR HANDOVER ACCESS message (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HOACC_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmass (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	IMMASS_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmass_sp (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT message using unacknowledged operation (L3 -> L2) on the MS paging channel.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	IMMASS_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmassx (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT_EXTENDED)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT EXTENDED message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	IMMASSX_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmassx_sp (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT_EXTENDED)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT EXTENDED message using unacknowledged operation (L3 -> L2) on the MS paging channel.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	IMMASSX_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmassRej (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT REJECT message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	IMMASS_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqImmassRej_sp (DL_UNIT_DATA_REQUEST_IMMEDIATE_ASSIGNMENT_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR IMMEDIATE ASSIGNMENT REJECT message using unacknowledged operation (L3 -> L2) on the MS paging channel.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	IMMASS_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatInMsrRpt (DL_DATA_INDICATION_MEASUREMENT_REPORT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR MEASUREMENT REPORT message using unacknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MSR_RPT_PDU	
fn	FN	
Detailed Comments:	The ASPs are continuously received during the testing.	

ASP Type Definition		
ASP Name:	DL_UdatRqSMSCBData (DL_UNIT_DATA_REQUEST_SMSCB_DATA)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the SMSCB data using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SMSCB_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqPg1Rq (DL_UNIT_DATA_REQUEST_PAGING_REQUEST_TYPE1)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR PAGING REQUEST TYPE 1 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	PG1_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqPg2Rq (DL_UNIT_DATA_REQUEST_PAGING_REQUEST_TYPE2)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR PAGING REQUEST TYPE 2 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	PG2_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqPg3Rq (DL_UNIT_DATA_REQUEST_PAGING_REQUEST_TYPE3)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR PAGING REQUEST TYPE 3 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
pgg	PGG	
msg	PG3_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSchinfo (DL_UNIT_DATA_REQUEST_SYNCHRONIZATION_CHANNEL_INFORMATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYNCHRONIZATION CHANNEL INFORMATION message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SCHINFO_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo1 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE1)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 1 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO1_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo1_nh (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE1)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 3 message in case of non hopping	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO3_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo2 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE2)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 2 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO2_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo2bis (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE2bis)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 2bis message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO2bis_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo2ter (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE2ter)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 2ter message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO2ter_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo3 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE3)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 3 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO3_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo4 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE4)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 4 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO4_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo5 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE5)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 5 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO5_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo5bis (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE5bis)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 5bis message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO5bis_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo5ter (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE5ter)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 5ter message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO5ter_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo6 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE6)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 6 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO6_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo7 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE7)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 7 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO7_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatRqSysinfo8 (DL_UNIT_DATA_REQUEST_SYSTEM_INFORMATION_TYPE8)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR SYSTEM INFORMATION TYPE 8 message using unacknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SYSINFO8_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_UdatInClmChn (DL_UdatINDICATION_CLASSMARK_CHANGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR CLASSMARK CHANGE message using unacknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CLM_CHN_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAssCmd (DL_DATA_REQUEST_ASSIGNMENT_COMMAND)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR ASSIGNMENT COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ASS_CMD_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInAssCom (DL_DATA_INDICATION_ASSIGNMENT_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR ASSIGNMENT COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ASS_COM_PDU	
fn	FN	
Detailed Comments:	The ASP is a result of the ASP DL_RESUME_REQUEST of the SUT sending the ASSIGNMENT COMPLETE message.	

ASP Type Definition		
ASP Name:	DL_DatInAssfl (DL_DATA_INDICATION_ASSIGNMENT_FAILURE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR ASSIGNMENT FAILURE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ASSFL_PDU	
fn	FN	
Detailed Comments:	The ASP is a result of the ASP DL_RECONNECT_REQUEST of the SUT sending the ASSIGNMEN FAILURE message.	

ASP Type Definition		
ASP Name:	DL_DatInCmsRq (DL_DATA_INDICATION_CM_SERVICE_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM CM SERVICE REQUEST message (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CMS_RQ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqChmmo (DL_DATA_REQUEST_CHANNEL_MODE_MODIFY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CHANNEL MODE MODIFY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CHMMO_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInChmmoAck (DL_DATA_INDICATION_CHANNEL_MODE_MODIFY_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR CHANNEL MODE MODIFY ACKNOWLEDGE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CHMMO_ACK_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqChRel (DL_DATA_REQUEST_CHANNEL_RELEASE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CHANNEL_RELEASE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CH_REL_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqChRelErr (DL_DATA_REQUEST_CHANNEL_RELEASE_ERROR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CHANNEL_RELEASE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CH_REL_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCphmCmd (DL_DATA_REQUEST_CIPHERING_MODE_COMMAND)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CIPHERING_MODE_COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPHM_CMD_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCphmCmdErr (DL_DATA_REQUEST_CIPHERING_MODE_COMMAND_ERROR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the invalid RR CIPHERING_MODE_COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPHM_CMD_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCphmCom (DL_DATA_INDICATION_CIPHERING_MODE_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR CIPHERING MODE COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPHM_COM_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInClmChn (DL_DAT_INDICATION_CLASSMARK_CHANGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR CLASSMARK CHANGE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CLM_CHN_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqClmEnq (DL_DATA_REQUEST_CLASSMARK_ENQUIRY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR CLASSMARK ENQUIRY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CLM_ENQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqFrqre (DL_DATA_REQUEST_FREQUENCY_REDEFINITION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR FREQUENCY REDEFINITION message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	FRQRE_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqHoCmd (DL_DATA_REQUEST_HANDOVER_COMMAND)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR HANDOVER COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HO_CMD_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInHoCom (DL_DATA_INDICATION_HANDOVER_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR HANDOVER COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HO_COM_PDU	
fn	FN	
Detailed Comments:	The ASP is a result of the ASP DL_RESUME_REQUEST of the SUT sending the HANDOVER COMPLETE message.	

ASP Type Definition		
ASP Name:	DL_DatInHofl (DL_DATA_INDICATION_HANDOVER_FAILURE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR HANDOVER FAILURE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HOFL_PDU	
fn	FN	
Detailed Comments:	The ASP is a result of the ASP DL_RECONNECT_REQUEST of the SUT sending the HANDOVER FAILURE message.	

ASP Type Definition		
ASP Name:	DL_DatRqPhyinfo (DL_DATA_REQUEST_PHYSICAL_INFORMATION)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the RR PHYSICAL INFORMATION message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	PHYINFO_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInRrst (DL_DATA_INDICATION_RR_STATUS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the RR STATUS message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	RRST_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAuthRej (DL_DATA_REQUEST_AUTHENTICATION_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM AUTHENTICATION REJECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	AUTH_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAuthRq (DL_DATA_REQUEST_AUTHENTICATION_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM AUTHENTICATION REQUEST message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	AUTH_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInAuthRes (DL_DATA_INDICATION_AUTHENTICATION_RESPONSE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM AUTHENTICATION RESPONSE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	AUTH_RES_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCmsAcp (DL_DATA_REQUEST_CM_SERVICE_ACCEPT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM CM SERVICE ACCEPT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CMS_ACP_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCmsRej (DL_DATA_REQUEST_CM_SERVICE_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM CM SERVICE REJECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CMS_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAbrt (DL_DATA_REQUEST_ABORT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM ABORT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ABRT_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqIdRq (DL_DATA_REQUEST_IDENTITY_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM IDENTITY REQUEST message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ID_RQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInIdRes (DL_DATA_INDICATION_IDENTIFY_RESPONSE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM IDENTITY RESPONSE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ID_RES_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqLupAcp (DL_DATA_REQUEST_LOCATION_UPDATING_ACCEPT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM LOCATION UPDATING ACCEPT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	LUP_ACP_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqLupAcpErr (DL_DATA_REQUEST_LOCATION_UPDATING_ACCEPT_ERR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM LOCATION UPDATING ACCEPT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	LUP_ACP_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqLupRej (DL_DATA_REQUEST_LOCATION_UPDATING_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM LOCATION UPDATING REJECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	LUP_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInMmst (DL_DATA_INDICATION_MM_STATUS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM STATUS message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MMST_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqTmsireCmd (DL_DATA_REQUEST_TMSI_REALLOCATION_COMMAND)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the MM TMSI REALLOCATION COMMAND message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	TMSIRE_CMD_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInTmsireCom (DL_DATA_INDICATION_TMSI_REALLOCATION_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the MM TMSI REALLOCATION COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	TMSIRE_COM_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqAlert (DL_DATA_REQUEST_ALERTING)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC ALERTING message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ALERT_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInAlert (DL_DATA_INDICATION_ALERTING)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC ALERTING message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ALERT_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCallCo (DL_DATA_INDICATION_CALL_CONFIRMED)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC CALL CONFIRMED message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CALL_CO_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCallProc (DL_DATA_REQUEST_CALL_PROCEEDING)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC CALL PROCEEDING message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CALL_PROC_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqConn (DL_DATA_REQUEST_CONNECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC CONNECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqConnErr (DL_DATA_REQUEST_CONNECT_ERR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC CONNECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInConn (DL_DATA_INDICATION_CONNECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC CONNECT message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqConnAck (DL_DATA_REQUEST_CONNECT_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC CONNECT ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInConnAck (DL_DATA_INDICATION_CONNECT_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the reception of the CC CONNECT ACKNOWLEDGE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_ACK_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqDisc (DL_DATA_REQUEST_DISCONNECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC DISCONNECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	DISC_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqDiscErr (DL_DATA_REQUEST_DISCONNECT_ERR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC DISCONNECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	DISC_PDU_ERR	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInDisc (DL_DATA_INDICATION_DISCONNECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC DISCONNECT message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	DISC_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInESetup (DL_DATA_INDICATION_EMERGENCY_SETUP)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC EMERGENCY_SETUP message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ESETUP_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqFac (DL_DATA_REQUEST_FACILITY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC FACILITY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	FAC_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInFac (DL_DATA_INDICATION_FACILITY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC FACILITY message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	FAC_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInHold (DL_DATA_INDICATION_HOLD)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC HOLD message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HOLD_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqHoldAck (DL_DATA_REQUEST_HOLD_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC HOLD ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	HOLD_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqModify (DL_DATA_REQUEST_MODIFY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC MODIFY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInModify (DL_DATA_INDICATION_MODIFY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC MODIFY message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqModifyCom (DL_DATA_REQUEST_MODIFY_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC MODIFY COMPLETE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_COM_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqModifyRej (DL_DATA_REQUEST_MODIFY_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC MODIFY REJECT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInModifyRej (DL_DATA_INDICATION_MODIFY_REJECT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC MODIFY REJECT message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	MODIFY_REJ_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqNotify (DL_DATA_REQUEST_NOTIFY)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC NOTIFY message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	NOTIFY_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqProg (DL_DATA_REQUEST_PROGRESS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC PROGRESS message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	PROG_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqRegister (DL_DATA_REQUEST_REGISTER)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the SS REGISTER message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REGISTER_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInRegister (DL_DATA_INDICATION_REGISTER)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to receive the transmission of the SS REGISTER message using acknowledged operation.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REGISTER_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqRel (DL_DATA_REQUEST_RELEASE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC RELEASE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REL_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInRel (DL_DATA_INDICATION_RELEASE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC RELEASE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REL_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqRelCmp (DL_DATA_REQUEST_RELEASE_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC RELEASE COMPLETE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REL_COM_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInRelCmp (DL_DATA_INDICATION_RELEASE_COMPLETE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC RELEASE COMPLETE message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	REL_COM_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqSetup (DL_DATA_REQUEST_SETUP)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC SETUP message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SETUP_MT_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInSetup (DL_DATA_INDICATION_SETUP)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC SETUP message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	SETUP_MO_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInStartDtmf (DL_DATA_INDICATION_START_DTMF)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC START_DTMF message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	START_DTMF_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqStartDtmfAck (DL_DATA_REQUEST_START_DTMF_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC START DTMF ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	START_DTMF_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqStartDtmfRej (DL_DATA_REQUEST_START_DTMF_REJCT)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC START DTMF REJCT message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	START_DTMF_REJ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInStopDtmf (DL_DATA_INDICATION_STOP_DTMF)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC STOP_DTMF message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	STOP_DTMF_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqStopDtmfAck (DL_DATA_REQUEST_STOP_DTMF_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC STOP DTMF ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	STOP_DTMF_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCcst (DL_DATA_REQUEST_CC_STATUS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC STATUS message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CCST_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCcst (DL_DATA_INDICATION_CC_STATUS)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the CC STATUS message using acknowledged operation (L2 -> L3).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CCST_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCcstEnq (DL_DATA_REQUEST_CC_STATUS_ENQ)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CC STATUS_ENQ message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CCST_ENQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqUndefCC (DL_DATA_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to transmit any message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CONN_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqUndefMM (DL_DATA_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to transmit any message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	ID_RES_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqUndefRR (DL_DATA_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to transmit any message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	PART_REL_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqUnknown (DL_DATA_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to transmit any message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CCST_ENQ_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCpData (DL_DATA_INDICATION_CP_DATA)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the SMS CP DATA message using acknowledged operation (L2 -> L3) for MT.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CP_DATA_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCpData (DL_DATA_REQUEST_CP_DATA)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the SMS CP DATA message using acknowledged operation (L3 -> L2) for MO.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CP_DATA_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatInCpDataAck (DL_DATA_INDICATION_CPDATA_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to indicate the receipt of the SMS CP DATA message using acknowledged operation (L2 -> L3) .	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPDATA_ACK_PDU	
fn	FN	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCpDataAck (DL_DATA_REQUEST_CPDATA_ACKNOWLEDGE)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the CP DATA ACKNOWLEDGE message using acknowledged operation (L3 -> L2).	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPDATA_ACK_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	DL_DatRqCpError (DL_DATA_REQUEST_CP_ERROR)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the transmission of the SMS CP ERROR message using acknowledged operation (L3 -> L2) for MO.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
msg	CPERR_PDU	
Detailed Comments:		

ASP Type Definition		
ASP Name:	MDL_RelRq (MDL_RELEASE_REQUEST)	
PCO Type:	SAP0_3	
Comments:	The ASP is used to request the local end termiination of a previous established acknowledged mode operation (L3 -> L2). The local end termination procedure is initiated.	
Parameter Name	**Parameter Type**	**Comments**
sapi	SAPID	
logic_ch	LOGICCH	
release_mode	RelMode	
Detailed Comments:		

PDU type definitions

TTCN PDU Type definitions

PDU Type Definition		
PDU Name:	ASS_CMD_PDU	
PCO Type:	SAP0_3	
Comments:	RR ASSSIGNMENT COMMAND n -> ms GSM 04.08, 9.1.2	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
ch1d_at	CHD	
pcmd	PCMD	
frql_at	FRQL	
cchd	CCHD	
ch1mod	CHMOD	
ch2d_at	CHD	
ch2mod	CHMOD	
ma_at	MA	
strt	STRT	
frql_bt	FRQL	
ch1d_bt	CHD	
ch2d_bt	CHD	
frqchs_bt	FRQCHS	
ma_bt	MA	
cphms	CPHMS	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ASS_COM_PDU	
PCO Type:	SAP0_3	
Comments:	RR ASSSIGNMENT COMPLETE ms -> n GSM 04.08, 9.1.3	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ASSFL_PDU	
PCO Type:	SAP0_3	
Comments:	RR ASSSIGNMENT FAILURE ms -> n GSM 04.08, 9.1.4	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CHMMO_PDU	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL MODE MODIFY n -> ms GSM 04.08, 9.1.5	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
chd	CHD	
chmod	CHMOD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CHMMO_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL MODE MODIFY ACKNOWLEDGE ms -> n GSM 04.08, 9.1.6	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
chd	CHD	
chmod	CHMOD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CH_REL_PDU	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL RELEASE n -> ms GSM 04.08, 9.1.7	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CH_REL_PDU_ERR	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL RELEASE containing additional unknown IE n -> ms GSM 04.08, 9.1.7	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
add	OCTETSTRING	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CH_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR CHANNEL REQUEST ms -> n GSM 04.08, 9.1.8	
Field Name	**Field Type**	**Comments**
ecau_rrf	BITSTRING [8]	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CPHM_CMD_PDU	
PCO Type:	SAP0_3	
Comments:	RR CIPHERING MODE COMMAND n -> ms GSM 04.08, 9.1.9	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
cph_res	CPH_RES	
cphms	CPHMS	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CPHM_CMD_PDU_ERR	
PCO Type:	SAP0_3	
Comments:	RR CIPHERING MODE COMMAND with additional unknown IE GSM 04.08, 9.1.9	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
cph_res	CPH_RES	
cphms	CPHMS	
add	OCTETSTRING	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CPHM_COM_PDU	
PCO Type:	SAP0_3	
Comments:	RR CIPHERING MODE COMPLETE ms -> n GSM 04.08, 9.1.10	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
mei	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CLM_CHN_PDU	
PCO Type:	SAP0_3	
Comments:	RR CLASSMARK CHANGE ms -> n GSM 04.08, 9.1.11	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
msclm	MSCLM2	
msclm_adi	MSCLM3	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CLM_ENQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR CLASSMARK ENQUIRY n -> ms GSM 04.08, 9.1.12	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	FRQRE_PDU	
PCO Type:	SAP0_3	
Comments:	RR FREQUENCY REDEFINITIONY n -> ms GSM 04.08, 9.1.13	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
chd	CHD	
ma	MA	
strt	STRT	
cchd	CCHD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HOACC_PDU	
PCO Type:	SAP0_3	
Comments:	RR HANDOVER ACCESS ms -> n GSM 04.08, 9.1.14	
Field Name	**Field Type**	**Comments**
horf	HORF	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HO_CMD_PDU	
PCO Type:	SAP0_3	
Comments:	RR HANDOVER COMMAND n -> ms GSM 04.08, 9.1.15	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
cd	CD	
ch1d_at	CHD	
horf	HORF	
pcmd	PCMD	
synchi	SYNCHI	
frqsl_at	FRQL	
frql_at	FRQL	
cchd	CCHD	
ch1mod	CHMOD	
ch2d_at	CHD	
ch2mod	CHMOD	
frqchs_at	FRQCHS	
ma_at	MA	
strt	STRT	
rtdif	TDIF	
ta	TA	
frqsl_bt	FRQL	
frql_bt	FRQL	
ch1d_bt	CHD	
ch2d_bt	CHD	
frqchs_bt	FRQCHS	
ma_bt	MA	
cphms	CPHMS	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HO_COM_PDU	
PCO Type:	SAP0_3	
Comments:	RR HANDOVER COMPLETE ms -> n GSM 04.08, 9.1.16	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
motdif	MTDIF	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HOFL_PDU	
PCO Type:	SAP0_3	
Comments:	RR HANDOVER FAILURE ms -> n GSM 04.08, 9.1.17	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	IMMASS_PDU	
PCO Type:	SAP0_3	
Comments:	RR IMMEDIATE ASSIGNMENT n -> ms GSM 04.08, 9.1.18	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
shoct	SHOCT	
pm	PM	
chd	CHD	
rqr	RQR	
ta	TA	
ma	MA	
strt	STRT	
iaroct	IARESTOCT	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	IMMASSX_PDU	
PCO Type:	SAP0_3	
Comments:	RR IMMEDIATE ASSIGNMENT n -> ms GSM 04.08, 9.1.19	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
shoct	SHOCT	
pm	PM	
chd1	CHD	
rqr1	RQR	
ta1	TA	
chd2	CHD	
rqr2	RQR	
ta2	TA	
ma	MA	
strt	STRT	
iaxroct	OCTETSTRING [1..4]	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	IMMASS_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	RR IMMEDIATE REJECT n -> ms GSM 04.08, 9.1.20	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
shoct	SHOCT	
pm	PM	
rqr1	RQR	
wi1	WI	
rqr2	RQR	
wi2	WI	
rqr3	RQR	
wi3	WI	
rqr4	RQR	
wi4	WI	
iarroct	OCTETSTRING [3]	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	MSR_RPT_PDU	
PCO Type:	SAP0_3	
Comments:	RR MEASUREMENT REPORT ms -> n GSM 04.08, 9.1.21	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
msrr	MSRR	
Detailed Comments:		

PDU Type Definition		
PDU Name:	PG1_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR PAGING REQUEST_TYPE1 n -> ms GSM 04.08, 9.1.22	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
chn_m1_2	CHNEED	
pm	PM	
mi1	MI	
mi2	MI	
p1roct	OCTETSTRING [0..17]	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	PG2_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR PAGING REQUEST TYPE2 n -> ms GSM 04.08, 9.1.23	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
chn_m1_2	CHNEED	
pm	PM	
mi1	TMSI	
mi2	TMSI	
mi3	MI	
p2roct	OCTETSTRING [1..11]	
Detailed Comments:	The message has a fixed length of 23 octets.	

PDU Type Definition		
PDU Name:	PG3_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	RR PAGING REQUEST TYPE3 n -> ms GSM 04.08, 9.1.24	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
chn_m1_2	CHNEED	
pm	PM	
mi1	TMSI	
mi2	TMSI	
mi3	TMSI	
mi4	TMSI	
p3roct	OCTETSTRING [3]	
Detailed Comments:	The message has an L2 pseudo length of 19 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	PG_RES_PDU	
PCO Type:	SAP0_3	
Comments:	RR PAGING RESPONSE ms -> n GSM 04.08, 9.1.25	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
shoct	SHOCT	
cphksn	CPHKSN	
msclm	MSCLM2	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	PART_REL_PDU	
PCO Type:	SAP0_3	
Comments:	RR PARTIAL RELEASE n -> ms GSM 04.08, 9.1.26	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
chd	MAC	
Detailed Comments:		

PDU Type Definition		
PDU Name:	PHYINFO_PDU	
PCO Type:	SAP0_3	
Comments:	RR PHYSICAL INFORMATION n -> ms GSM 04.08 clause 9.1.28	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
ta	TA	
Detailed Comments:		

PDU Type Definition		
PDU Name:	RRST_PDU	
PCO Type:	SAP0_3	
Comments:	RR STATUS n <-> ms GSM 04.08, 9.1.29	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
rrcau	RRCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SCHINFO_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYNCHRONIZATION CHANNEL INFORMATION n -> ms GSM 04.08, 9.1.30	
Field Name	**Field Type**	**Comments**
ncc	NCC	
bcc	BCC	
t1	T1	
t2	T2	
t3_	T3_	
Detailed Comments:	SCHINFO_PDU has a total lengh of 25 bits (GSM 04.04).	

PDU Type Definition		
PDU Name:	SYSINFO1_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE1 n -> ms GSM 04.08, 9.1.31	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
cchd	CCHD	
rachcp	RACHCP	
si1roct	OCTETSTRING [1]	
Detailed Comments:	The message has an L2 pseudo length of 21 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO2_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE2 n -> ms GSM 04.08, 9.1.32	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
bcchfl	NCD	
nccp	NCCP	
rachcp	RACHCP	
Detailed Comments:	The message has an L2 pseudo length of 22 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO2bis_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE2bis or 2ter n -> ms GSM 04.08, 9.1.33, 9.1.33a	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
xbcchfl	NCD	
rachcp	RACHCP	
si2bisroct	OCTETSTRING [1..4]	
Detailed Comments:	This PDU type is used both for SYSINFO2bis, as well as for 2ter. The 2bis message has an L2 pseudo length of 21 octets and a total length of 23 octets. The 2ter message has an L2 pseudo length of 18 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO2ter_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION 2ter n -> ms GSM 04.08, 9.1.34	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
xbcchfl	NCD2	
si2terroct	OCTETSTRING[4]	
Detailed Comments:	The 2ter message has an L2 pseudo length of 18 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO3_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE3 n -> ms GSM 04.08, 9.1.34	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
ci	CI	
lai	LAI	
ccd	CCD	
co	CO	
csp	CSP	
rachcp	RACHCP	
si3roct	SI3RO	
Detailed Comments:	The message has an L2 pseudo length of 18 octets and a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO4_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE4 n -> ms GSM 04.08, 9.1.35	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
lai	LAI	
csp	CSP	
rachcp	RACHCP	
cbchd	CHD	
cbchma	MA	
si4roct	SI4RO	
Detailed Comments:	The message has a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO5_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE5 n -> ms GSM 04.08, 9.1.36	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
bcchfl	NCD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SYSINFO5bis_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE5bis n -> ms GSM 04.08, 9.1.38	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
xbcchfl	NCD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SYSINFO5ter_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE5ter n -> ms GSM 04.08, 9.1.39	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
xbcchfl	NCD2	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SYSINFO6_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE6 n -> ms GSM 04.08, 9.1.38	
Field Name	**Field Type**	**Comments**
ski	SKI	
rrpd	PD	
mt	MT	
ci	CI	
lai	LAI	
co	CO	
nccp	NCCP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SYSINFO7_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE7 n -> ms GSM 04.08, 9.1.41	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
si7roct	SI7RO	
Detailed Comments:	The message has a total length of 23 octets.	

PDU Type Definition		
PDU Name:	SYSINFO8_PDU	
PCO Type:	SAP0_3	
Comments:	RR SYSTEM INFORMATION TYPE8 n -> ms GSM 04.08, 9.1.42	
Field Name	**Field Type**	**Comments**
l2_pl	LENGTH	
ski	SKI	
rrpd	PD	
mt	MT	
si8roct	SI7RO	
Detailed Comments:	The message has a total length of 23 octets.	

PDU Type Definition		
PDU Name:	ABRT_PDU	
PCO Type:	SAP0_3	
Comments:	MM ABORT n -> ms GSM 04.08, 9.2.8	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
rejcau	REJCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	AUTH_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	MM AUTHENTICATION REJECT n -> ms GSM 04.08, 9.2.1	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	AUTH_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM AUTHENTICATION REQUEST n -> ms GSM 04.08, 9.2.2	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
shoct	SHOCT	
cphksn	CPHKSN	
rand	RAND	
Detailed Comments:		

PDU Type Definition		
PDU Name:	AUTH_RES_PDU	
PCO Type:	SAP0_3	
Comments:	MM AUTHENTICATION RESPONSE ms -> n GSM 04.08, 9.2.3	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
sres	SRES	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CMRE_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM CM REESTABLISHMENT REQUEST ms -> n GSM 04.08, 9.2.4	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
shoct	SHOCT	
cphksn	CPHKSN	
msclm	MSCLM2	
mi	MI	
lai	LAI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CMS_ACP_PDU	
PCO Type:	SAP0_3	
Comments:	MM CM SERVICE ACCEPT n -> ms GSM 04.08, 9.2.5	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CMS_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	MM CM SERVICE REJECT n -> ms GSM 04.08, 9.2.6	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
mmcau	REJCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CMS_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM CM SERVICE REQUEST ms -> n GSM 04.08, 9.2.9	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
cphksn	CPHKSN	
svtype	CMSVTYPE	
msclm	MSCLM2	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ID_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM IDENTITY REQUEST n -> ms GSM 04.08, 9.2.10	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
shoct	SHOCT	
idtype	IDTYPE	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ID_RES_PDU	
PCO Type:	SAP0_3	
Comments:	MM IDENTITY RESPONSE ms -> n GSM 04.08, 9.2.11	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	IMSID_IN_PDU	
PCO Type:	SAP0_3	
Comments:	MM IMSI DETACH INDICATION ms -> n GSM 04.08, 9.2.12	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
msclm	MSCLM1	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	LUP_ACP_PDU	
PCO Type:	SAP0_3	
Comments:	MM LOCATION UPDATING ACCEPT n -> ms GSM 04.08, 9.2.13	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
lai	LAI	
mi	MI	
fop	IEI_8	
Detailed Comments:		

PDU Type Definition		
PDU Name:	LUP_ACP_PDU_ERR	
PCO Type:	SAP0_3	
Comments:	MM LOCATION UPDATING ACCEPT n -> ms GSM 04.08, 9.2.13	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
lai	LAI	
mi	MI	
dupmi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	LUP_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	MM LOCATION UPDATING REJECT n -> ms GSM 04.08, 9.2.14	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
rejcau	REJCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	LUP_RQ_PDU	
PCO Type:	SAP0_3	
Comments:	MM LOCATION UPDATING REQUEST ms -> n GSM 04.08, 9.2.15	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
cphksn	CPHKSN	
lutype	LUT	
lai	LAI	
msclm	MSCLM1	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	MMST_PDU	
PCO Type:	SAP0_3	
Comments:	MM STATUS n <-> ms GSM 04.08, 9.2.16	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
rejcau	REJCAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	TMSIRE_CMD_PDU	
PCO Type:	SAP0_3	
Comments:	MM TMSI REALLOCATION COMMAND n -> ms GSM 04.08, 9.2.17	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
lai	LAI	
mi	MI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	TMSIRE_COM_PDU	
PCO Type:	SAP0_3	
Comments:	MM TMSI REALLOCATION COMPLETE ms -> n GSM 04.08, 9.2.18	
Field Name	**Field Type**	**Comments**
ski	SKI	
mmpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ALERT_PDU	
PCO Type:	SAP0_3	
Comments:	CC ALERTING ms <-> n GSM 04.08, 9.3.1	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
fie	FIE	
pi	PI	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CALL_CO_PDU	
PCO Type:	SAP0_3	
Comments:	CC CALL CONFIRMED ms -> n GSM 04.08, 9.3.2	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcri	RPI	
bcap1	BCAP	
bcap2	BCAP	
cau	CAU	
cccap	CCCAP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CALL_PROC_PDU	
PCO Type:	SAP0_3	
Comments:	CC CALL PROCEEDING n -> ms GSM 04.08, 9.3.3	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcri	RPI	
bcap1	BCAP	
bcap2	BCAP	
fie	FIE	
pi	PI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CONN_PDU	
PCO Type:	SAP0_3	
Comments:	CC CONNECT n <-> ms GSM 04.08, 9.3.5	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
fie	FIE	
pi	PI	
cnn	CNN	
cns	CNS	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CONN_PDU_ERR	
PCO Type:	SAP0_3	
Comments:	CC CONNECT n <-> ms Used as invalid message	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
unknown	UNKWN	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CONN_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CC CONNECT ACKNOWLEDGE ms <-> n GSM 04.08, 9.3.6	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	DISC_PDU	
PCO Type:	SAP0_3	
Comments:	CC DISCONNECT ms <-> n (both directions) GSM 04.08, 9.3.7	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
fie	FIE	
pi	PI	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	DISC_PDU_ERR	
PCO Type:	SAP0_3	
Comments:		
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
unknown	UNKWN	
Detailed Comments:		

PDU Type Definition		
PDU Name:	ESETUP_PDU	
PCO Type:	SAP0_3	
Comments:	CC EMERGENCY SETUP ms -> n GSM 04.08, 9.3.8	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcap	BCAP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	FAC_PDU	
PCO Type:	SAP0_3	
Comments:	CC FACILITY n <-> ms GSM 04.08, 9.3.9	
Field Name	**Field Type**	**Comments**
ti	TI	
cc_sspd	PD	
mt	MT	
fie	FIE	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HOLD_PDU	
PCO Type:	SAP0_3	
Comments:	CC HOLD ms -> n GSM 04.08, 9.3.10	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	HOLD_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CC HOLD ACKNOWLEDGE n -> ms GSM 04.08, 9.3.11	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	MODIFY_PDU	
PCO Type:	SAP0_3	
Comments:	CC MODIFY ms <-> n GSM 04.08, 9.3.13	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcap	BCAP	
llcmp	LLCMP	
hlcmp	HLCMP	
rcsd	RCSD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	MODIFY_COM_PDU	
PCO Type:	SAP0_3	
Comments:	CC MODIFY COMPLETE ms <-> n GSM 04.08, 9.3.14	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcap	BCAP	
llcmp	LLCMP	
hlcmp	HLCMP	
rcsd	RCSD	
Detailed Comments:		

PDU Type Definition		
PDU Name:	MODIFY_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	CC MODIFY REJECT ms <-> n GSM 04.08, 9.3.15	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcap	BCAP	
cau	CAU	
llcmp	LLCMP	
hlcmp	HLCMP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	NOTIFY_PDU	
PCO Type:	SAP0_3	
Comments:	CC NOTIFY ms <-> n GSM 04.08, 9.3.16	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
nti	NTI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	REGISTER_PDU	
PCO Type:	SAP0_3	
Comments:	SS REGISTER ms <-> n GSM 04.80, 2.4	
Field Name	**Field Type**	**Comments**
ti	TI	
sspd	PD	
mt	MT	
fie	FIE	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	PROG_PDU	
PCO Type:	SAP0_3	
Comments:	CC PROGRESS n -> ms GSM 04.08, 9.3.17	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
pi	PI	
uu	UU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	REL_PDU	
PCO Type:	SAP0_3	
Comments:	CC RELEASE n <-> ms GSM 04.08, 9.3.18	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
cau2	CAU	
fie	FIE	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	REL_COM_PDU	
PCO Type:	SAP0_3	
Comments:	CC or SS RELEASE COMPLETE n <-> ms GSM 04.08, 9.3.19; GSM 04.80, 2.5	
Field Name	**Field Type**	**Comments**
ti	TI	
cc_sspd	PD	
mt	MT	
cau	CAU	
fie	FIE	
uu	UU	
ssvi	SSVI	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SETUP_MO_PDU	
PCO Type:	SAP0_3	
Comments:	CC SETUP n <- ms GSM 04.08, 9.3.23.2	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcri	RPI	
bcap1	BCAP	
bcap2	BCAP	
fie	FIE	
cgps	CGPS	
cdpn	CDPN	
cdps	CDPS	
llcri	RPI	
llcmp1	LLCMP	
llcmp2	LLCMP	
hlcri	RPI	
hlcmp1	HLCMP	
hlcmp2	HLCMP	
uu	UU	
ssvi	SSVI	
clirsup	CLRSUP	
clirinv	CLRINV	
cccap	CCCAP	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SETUP_MT_PDU	
PCO Type:	SAP0_3	
Comments:	CC SETUP n -> ms GSM 04.08, 9.3.23.1	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
bcri	RPI	
bcap1	BCAP	
bcap2	BCAP	
fie	FIE	
pi	PI	
sig	SIGNAL	
cgpn	CGPN	
cgps	CGPS	
cdpn	CDPN	
cdps	CDPS	
llcri	RPI	
llcmp1	LLCMP	
llcmp2	LLCMP	
hlcri	RPI	
hlcmp1	HLCMP	
hlcmp2	HLCMP	
uu	UU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	SMSCB_PDU	
PCO Type:	SAP0_3	
Comments:	SMSCB message GSM 03.41, 9.3	
Field Name	**Field Type**	**Comments**
blocktype	BLOCKTYPE	
serial_number	SERIAL_NUMBER	
message_id	OCTETSTRING[2]	
dcs	TPDCS	
page_param	BITSTRING[8]	
message_contents	OCTETSTRING	
Detailed Comments:		

PDU Type Definition		
PDU Name:	START_DTMF_PDU	
PCO Type:	SAP0_3	
Comments:	CC START DTMF ms -> n GSM 04.08, 9.3.24	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
kpf	KPF	
Detailed Comments:		

PDU Type Definition		
PDU Name:	START_DTMF_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CC START DTMF ACKNOWLEDGE n -> ms GSM 04.08, 9.3.25	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
kpf	KPF	
Detailed Comments:		

PDU Type Definition		
PDU Name:	START_DTMF_REJ_PDU	
PCO Type:	SAP0_3	
Comments:	CC START DTMF REJECT n -> ms GSM 04.08, 9.3.26	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CCST_PDU	
PCO Type:	SAP0_3	
Comments:	CC STATUS n <-> ms GSM 04.08, 9.3.27	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
cau	CAU	
cst	CST	
acst	ACST	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CCST_ENQ_PDU	
PCO Type:	SAP0_3	
Comments:	CC STATUS ENQUIRY ms <-> n GSM 04.08, 9.3.28	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	STOP_DTMF_PDU	
PCO Type:	SAP0_3	
Comments:	CC STOP DTMF ms -> n GSM 04.08, 9.3.29	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	STOP_DTMF_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CC STOP DTMF acknowledge n -> ms GSM 04.08, 9.3.30	
Field Name	**Field Type**	**Comments**
ti	TI	
ccpd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CP_DATA_PDU	
PCO Type:	SAP0_3	
Comments:	SMS CP DATA ms <-> n GSM 04.11, 7.2.1	
Field Name	**Field Type**	**Comments**
ti	TI	
smspd	PD	
mt	MT	
CPdata	CPDATA	
Detailed Comments:	CPDATA contains RPDU - RP_ACK or RP_ERROR, either returning to the MS in case of MO, or reporting the outcome of a MT messaging attempt in case of MT .	

PDU Type Definition		
PDU Name:	CPDATA_ACK_PDU	
PCO Type:	SAP0_3	
Comments:	CP DATA ACKNOWLEDGE ms <-> n GSM 04.11, 7.2.2	
Field Name	**Field Type**	**Comments**
ti	TI	
smspd	PD	
mt	MT	
Detailed Comments:		

PDU Type Definition		
PDU Name:	CPERR_PDU	
PCO Type:	SAP0_3	
Comments:	CP_ERROR n <-> ms GSM 04.11, 7.2.3	
Field Name	**Field Type**	**Comments**
ti	TI	
smspd	PD	
mt	MT	
cp_cause	CP_CAU	
Detailed Comments:	CP_UDAT contains RPDU, RP_ACK or RP_ERROR, either returning to the MS in case of MO, or reporting the outcome of a MT messaging attempt in case of MT .	

Constraints Part

Test suite type constraint declarations

Structured type constraint declarations

Structured Type Constraint		
Constraint Name:	Bcap_MO(rchr1:B_2; itc1:B_3; strc1:B_2; nirr:B_1; ra1 :B_2; sacp:B_3; sb, nsb1, ndb1:B_1; ur1:B_4; ir1:B_2; parity:B_3; ce1:B_2; modemt1:B_5)	
Structured Type:	BCAP	
Derivation Path:		
Comments:	BS bearer capability without OCTET 7 for direction ms -> n	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	'07'O	
extb3	'1'B	
rchr	rchr1	
cs	'0'B	
tm	'0'B	
itc	itc1	
oct_3a_etc_1	-	
oct_3a_etc_2	-	
oct_3a_etc_3	-	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	'1'B	
spb	'0'B	
strc	strc1	
dplxm	'1'B	
config	'0'B	
nirr	nirr	
est	'0'B	
extb5	'1'B	
accid	'00'B	
ra	ra1	
sacp	sacp	
extb6	'0'B	
l1id	'01'B	
uil1	'0000'B	
sb	sb	
extb6a	'0'B	
nsb	nsb1	
nb	'0'B	
ndb	ndb1	
ur	ur1	
extb6b	'0'B	
ir	ir1	
nictx	'0'B	
nicrx	'0'B	
pi	parity	
extb6c	'1'B	
ce	ce1	
modemt	modemt1	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcapX_MO(rchr1:B_2; itc1:B_3; strc1:B_2; nirr:B_1; ra1 :B_2; sacp:B_3; sb, nsb1, ndb1:B_1; ur1:B_4; ir1:B_2; parity:B_3; ce1:B_2; modemt1:B_5; fctl:B_5)	
Structured Type:	BCAP	
Derivation Path:		
Comments:	BS bearer capability with OCTET 7 for direction ms -> n	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	('07'O, '08'O)	
extb3	'1'B	
rchr	rchr1	
cs	'0'B	
tm	'0'B	
itc	itc1	
oct_3a_etc_1	-	
oct_3a_etc_2	-	
oct_3a_etc_3	-	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	'1'B	
spb	'0'B	
strc	strc1	
dplxm	'1'B	
config	'0'B	
nirr	nirr	
est	'0'B	
extb5	'1'B	
accid	'00'B	
ra	ra1	
sacp	sacp	
extb6	'0'B	
l1id	'01'B	
uil1	'0000'B	
sb	sb	
extb6a	'0'B	
nsb	nsb1	
nb	'0'B	
ndb	ndb1	
ur	ur1	
extb6b	'0'B	
ir	ir1	
nictx	'0'B	
nicrx	'0'B	
pi	parity	
extb6c	'1'B	
ce	ce1	
modemt	modemt1	
extb7	'1'B IF_PRESENT	
l2id	'10'B IF_PRESENT	
uil2	fctl IF_PRESENT	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Bcap_Speech_Efr_MO(rcr:B_2)	
Structured Type:	BCAP	
Derivation Path:		
Comments:	EFR speech bearer capability for direction ms->n	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	('03'O, '04'O, '05'O, '06'O, '07'O)	
extb3	'0'B	
rchr	rcr	
cs	'0'B	
tm	'0'B	
itc	'000'B	
oct_3a_etc_1	?	
oct_3a_etc_2	?	
oct_3a_etc_3	*	
oct_3a_etc_4	*	
oct_3a_etc_5	*	
oct_3a_etc_6	*	
extb4	-	
spb	-	
strc	-	
dplxm	-	
config	-	
nirr	-	
est	-	
extb5	-	
accid	-	
ra	-	
sacp	-	
extb6	-	
l1id	-	
uil1	-	
sb	-	
extb6a	-	
nsb	-	
nb	-	
ndb	-	
ur	-	
extb6b	-	
ir	-	
nictx	-	
nicrx	-	
pi	-	
extb6c	-	
ce	-	
modemt	-	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Bcap_Speech_MO(rcr:B_2)	
Structured Type:	BCAP	
Derivation Path:		
Comments:	speech bearer capability for direction ms->n.	
Element Name	**Element Value**	**Comments**
iei	'00000100'B	
iel	'01'O	
extb3	'1'B	
rchr	rcr	
cs	'0'B	
tm	'0'B	
itc	'000'B	
oct_3a_etc_1	-	
oct_3a_etc_2	-	
oct_3a_etc_3	-	
oct_3a_etc_4	-	
oct_3a_etc_5	-	
oct_3a_etc_6	-	
extb4	-	
spb	-	
strc	-	
dplxm	-	
config	-	
nirr	-	
est	-	
extb5	-	
accid	-	
ra	-	
sacp	-	
extb6	-	
l1id	-	
uil1	-	
sb	-	
extb6a	-	
nsb	-	
nb	-	
ndb	-	
ur	-	
extb6b	-	
ir	-	
nictx	-	
nicrx	-	
pi	-	
extb6c	-	
ce	-	
modemt	-	
extb7	-	
l2id	-	
uil2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_Omit	
Structured Type:	NCD	
Derivation Path:		
Comments:	Omit neighbour cells description	
Element Name	**Element Value**	**Comments**
rfl2	-	
extind	-	
baind	-	
rfl4	-	
rfl	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_01	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under GSM900 with the ARFCN list = {10, 20, 40, 80, 90, 100, 110, 120}.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'0'B	
baind	'0'B	
rfl4	'0000'B	
rfl	'80200802008000000008000080200'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_47	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2bis and 5bis in cell B under EGSMwith the ARFCN list = {1005, 1010, 1015}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'0'B	
rfl4	'1101'B	
rfl	'F68AEC00000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_48	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 for DCS1800 for cell B in RR, MM and SMS test cases with the ARFCN list = {520, 590, 600, 700, 780, 810, 870}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'04411307BB00C80000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_50	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under GSM900 with the ARFCN list = {3, 9, 18, 25, 41, 43, 49, 50, 54, 58, 62, 66, 70, 80, 92, 124}.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'0'B	
baind	'0'B	
rfl4	'1000'B	
rfl	'000000080080222223050001020104'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_51	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under DCS1800 with the ARFCN list = {512, 543, 568, 589, 602, 641, 662, 683, 696, 711, 732, 754, 794, 851, 870, 871, 872, 884}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'0031EE85660D545ABFEB88D30DA0FF'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_52	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2bis under EGSMwith the ARFCN list = {985, 989, 995, 1010, 1014}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'EC867C413D00000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_53	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under GSM900 with the ARFCN list = {3, 9, 18, 25}.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'0'B	
baind	'0'B	
rfl4	'1000'B	
rfl	'000000000000000000000001020104'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_54	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under DCS1800 with the ARFCN list = {512, 543, 568, 589}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'000E39C540000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_55	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2bis under EGSMwith the ARFCN list = {985}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'EC800000000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_56	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2bis under EGSMwith the ARFCN list = {9, 43, 70, 500, 550, 990, 995}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'1'B	
baind	'0'B	
rfl4	'0000'B	
rfl	'09ED88BEF29086C000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_57	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under GSM900 with the ARFCN list = {3, 18, 25, 41, 43, 49, 50, 54, 58, 62, 66, 70, 80, 92, 124}.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'0'B	
baind	'0'B	
rfl4	'1000'B	
rfl	'00000008008022222305000102004'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_58	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under DCS1800 with the ARFCN list = {512, 568, 589, 602, 641, 662, 683, 696, 711, 732, 754, 794, 851, 870, 871, 872, 884}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'00372EA2A20C440B38B3FE7A0F3160'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_59	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under GSM900 with the ARFCN list = {3, 9, 18, 25, 41}.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'0'B	
baind	'0'B	
rfl4	'1000'B	
rfl	'00000000000000000000010001020104'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_60	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under GSM900 with the ARFCN list = {3, 18, 25, 41}.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'0'B	
baind	'0'B	
rfl4	'1000'B	
rfl	'00000000000000000000010001020004'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_61	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under DCS1800 with the ARFCN list = {512, 543, 568, 589, 602}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'00137AC37380000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_62	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under GSM900 with the ARFCN list = {3, 9, 18, 25, 41, 43, 49}.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'0'B	
baind	'0'B	
rfl4	'1000'B	
rfl	'00000000000000000010500010201O4'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_63	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under GSM900 with the ARFCN list = {3, 18, 25, 41, 43, 49}.	
Element Name	**Element Value**	**Comments**
rfl2	'00'B	
extind	'0'B	
baind	'0'B	
rfl4	'1000'B	
rfl	'00000000000000000010500010200O4'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst_64	
Structured Type:	NCD	
Derivation Path:		
Comments:	Default neighbour cells description for SYSTEM INFORMATION 2 and 5 under DCS1800 with the ARFCN list = {512, 543, 568, 589, 602, 641, 662}.	
Element Name	**Element Value**	**Comments**
rfl2	'10'B	
extind	'0'B	
baind	'0'B	
rfl4	'1001'B	
rfl	'0016B78F33EB2A0000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	BcchFreqLst2_65	
Structured Type:	NCD2	
Derivation Path:		
Comments:	Default neighbour cells description type 2 for SYSTEM INFORMATION 2 and 5 for DCS1800 with the ARFCN list = {45, 76, 891, 905, 985}.	
Element Name	**Element Value**	**Comments**
b128	'1'B	
mbrpt	'00'B	
baind	'0'B	
b122_124	'001'B	
rfl1	'1'B	
rfl	'D9D81CBCB8400000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CallCntrlCap	
Structured Type:	CCCAP	
Derivation Path:		
Comments:	CC state 'st'	
Element Name	**Element Value**	**Comments**
iei	'00010101'B	
iel	'01'O	
spr	'0000000'B	
dtmf	TSPX_DTMF	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_02	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in RR testing of GSM 900 with the ARFCN_list={20,30,50,70}.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'00000000000000200002000020080000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_03	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell A in RR testing of DCS1800 with the ARFCN list = {590, 650, 750, 850}..	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'89272827190000000000000000000000' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_04	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell B in RR testing of GSM 900.	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'00000000000000000000000000000200' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_24	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell selection testing : 3, 9, 18, 25, 41, 43, 49, 50, 54, 58, 62, 66, 70, 80, 92, 124	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'0800000008008022222305000102010 4' O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellChDes_24d	
Structured Type:	CCHD	
Derivation Path:		
Comments:	for cell selection testing : 512, 543, 568, 589, 602, 641, 662, 683, 696, 711, 732, 754, 794, 851, 870, 871, 872, 884	
Element Name	**Element Value**	**Comments**
iei	-	
rfl	'890031EE85660D545ABFEB88D30DA0FF'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellOpt_01	
Structured Type:	CO	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
sprb	'0'B	
pwrc	'0'B	
dtx	'10'B	
rlt	'0001'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellSelPara(crh, mtmc:INTEGER; neci:B_1)	
Structured Type:	CSP	
Derivation Path:		
Comments:	Default value.	
Element Name	**Element Value**	**Comments**
crh	INT_TO_BIT(crh, 3)	
mtmc	INT_TO_BIT(mtmc, 5)	
acs	'0'B	
neci	neci	
ram	'000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellSelPara_bis(crh, mtmc:INTEGER; neci:B_1; ram:B_6)	
Structured Type:	CSP	
Derivation Path:		
Comments:	Default value.	
Element Name	**Element Value**	**Comments**
crh	INT_TO_BIT(crh, 3)	
mtmc	INT_TO_BIT(mtmc, 5)	
acs	'1'B	
neci	neci	
ram	ram	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CellSelPara_ram(crh, mtmc:INTEGER; neci:B_1; ram:B_6)	
Structured Type:	CSP	
Derivation Path:		
Comments:	Default value.	
Element Name	**Element Value**	**Comments**
crh	INT_TO_BIT(crh, 3)	
mtmc	INT_TO_BIT(mtmc, 5)	
acs	'0'B	
neci	neci	
ram	ram	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChDescrp_nfh(type:CH_TDMA; slot:SN; tsc:TSC; par_arfcn: INTEGER)	
Structured Type:	CHD	
Derivation Path:		
Comments:	Channel description for CC testing, non hopping, no starting time.	
Element Name	**Element Value**	**Comments**
iei	-	
cht_schn	type	
tn	slot	
tsc	tsc	
hch	'0'B	
maio	-	
hsn	-	
spr	'00'B	
arfcn	INT_TO_BIT(par_arfcn, 10)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_r01	
Structured Type:	CHMOD	
Derivation Path:		
Comments:	for TC_26_6_13_1	
Element Name	**Element Value**	**Comments**
iei	'01100011'B	
mode	TSPX_ChMod1	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ChMod_sign	
Structured Type:	CHMOD	
Derivation Path:		
Comments:	signalling only	
Element Name	**Element Value**	**Comments**
iei	-	
mode	C_ChMod_s	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Chneed_01	
Structured Type:	CHNEED	
Derivation Path:		
Comments:	any channel	
Element Name	**Element Value**	**Comments**
ch2	'00'B	
ch1	'00'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ClassMark1	
Structured Type:	MSCLM1	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
spr1	'0'B	
rl	TSPX_RevLevel	
spr2	'0'B	
a5_1	TSPX_CiphAlgA5_1	
rfpc	TSPX_RfPwrCap	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CntrlChDscrp(att: INTEGER; babr, cch_con, bpm:B_3; timer:OCTETSTRING)	
Structured Type:	CCD	
Derivation Path:		
Comments:	Generic Control channel descriptor.	
Element Name	**Element Value**	**Comments**
spr1	'0'B	
att	INT_TO_BIT(att,1)	
babr	babr	
ccch_con	cch_con	
spr2	'00000'B	
bpm	bpm	
t3212	timer	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	CphKeySN_07(par: BITSTRING)	
Structured Type:	CPHKSN	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
sprb	'0'B	
ks	par	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Fn_01	
Structured Type:	FN	
Derivation Path:		
Comments:	not pertaining to the MS under test	
Element Name	**Element Value**	**Comments**
t1_	'00000'B	
t3	'000000'B	
t2	'00000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqBCCH(arfcn:INTEGER)	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Broadcast channel.	
Element Name	**Element Value**	**Comments**
hch	'0'B	
maio	-	
hsn	-	
spr	'00'B	
arfcn	INT_TO_BIT(arfcn, 10)	
maclength	'00'O	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FreqTCHa1	
Structured Type:	FRQPARA	
Derivation Path:		
Comments:	Traffic channel for cell A, ARFCN = 124 hopping, for GSM900.	
Element Name	**Element Value**	**Comments**
hch	'1'B	
maio	'000000'B	
hsn	'111111'B	
spr	-	
arfcn	-	
maclength	'01'O	
mac_8n	'00000001'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
flst	Frql_04	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Frql_04	
Structured Type:	FRQL	
Derivation Path:		
Comments:	frequency list for GSM900 hopping channel with only one frequency, ARFCN = 124	
Element Name	**Element Value**	**Comments**
iei	'00000101'B	
iel	'10'O	
fl	'08000000000000000000000000000000'O	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Hlcmp_AnyOrOmit	
Structured Type:	HLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE with parameter "any"' or "omit".	
Element Name	**Element Value**	**Comments**
iei	'01111101'B IF_PRESENT	
iel	*	
extb3	*	
cs	*	
in	*	
pmpp	*	
extb4	*	
hlci	*	
extb4a	*	
ehlci	*	
Detailed Comments:	used in test cases where the HLC is not checked	

Structured Type Constraint		
Constraint Name:	Hlcmp_NotApplicable	
Structured Type:	HLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE which is not applicable for a specific service (empty IE).	
Element Name	**Element Value**	**Comments**
iei	'01111101'B	
iel	'00'O	
extb3	-	
cs	-	
in	-	
pmpp	-	
extb4	-	
hlci	-	
extb4a	-	
ehlci	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Hlcmp_TS61	
Structured Type:	HLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE for teleservice 61.	
Element Name	**Element Value**	**Comments**
iei	'01111101'B	
iel	'02'O	
extb3	'1'B	
cs	'00'B	
in	'100'B	
pmpp	'01'B	
extb4	'?'B	
hlci	'0000100'B	
extb4a	*	
ehlci	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Hlcmp_TS62	
Structured Type:	HLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE for teleservice 62.	
Element Name	**Element Value**	**Comments**
iei	'01111101'B	
iel	'02'O	
extb3	'1'B	
cs	'00'B	
in	'100'B	
pmpp	'01'B	
extb4	'?'B	
hlci	'0000100'B	
extb4a	*	
ehlci	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct(iel:LENGTH)	
Structured Type:	IARESTOCT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iaroct1	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O,(11 - OC_OctToInt(iel)))	
iaroct2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	IaRestOct_01	
Structured Type:	IARESTOCT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iaroct1	'2B2B2B2B2B2B2B2B2B2B'O	
iaroct2	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Llcmp_AnyOrOmit	
Structured Type:	LLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE which has any or omit value in all fields.	
Element Name	**Element Value**	**Comments**
iei	'01111100'B IF_PRESENT	
iel	*	
extb3	*	
cs	*	
itc	*	
extb3a	*	
negind	*	
spb3a	*	
extb4	*	
tm	*	
itr	*	
extb4_1	*	
rateml	*	
extb5	*	
l1id	*	
uil1	*	
extb5a	*	
sb	*	
neg	*	
ur	*	
extb5b1	*	
ir	*	
nictx	*	
nicrx	*	
fctx	*	
fcrx	*	
spb5b1	*	
extb5b2	*	
hdrb	*	
mfs	*	
mode	*	
llineg	*	
ass	*	
ibob	*	
spb5b2	*	
extb5c	*	
nsb	*	
ndb	*	
pi	*	
extb5d	*	
dplxm	*	
modemt	*	
extb6	*	
l2id	*	
uil2	*	
extb6a1	*	
modl2	*	
spb6a1	*	
q933	*	
extb6a2	*	
usl2pi	*	
extb6b	*	
ws	*	
extb7	*	
l3id	*	
uil3	*	
extb7a1	*	
modl3	*	

spb7a1	*	
extb7a2	*	
usl3pi	*	
extb7b	*	
spb7b	*	
dps	*	
extb7c	*	
pws	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Llcmp_NotApplicable	
Structured Type:	LLCMP	
Derivation Path:		
Comments:	High Layer Compatibility IE which is not applicable for a specific service (empty IE).	
Element Name	**Element Value**	**Comments**
iei	'01111100'B	
iel	'00'O	
extb3	-	
cs	-	
itc	-	
extb3a	-	
negind	-	
spb3a	-	
extb4	-	
tm	-	
itr	-	
extb4_1	-	
rateml	-	
extb5	-	
l1id	-	
uil1	-	
extb5a	-	
sb	-	
neg	-	
ur	-	
extb5b1	-	
ir	-	
nictx	-	
nicrx	-	
fctx	-	
fcrx	-	
spb5b1	-	
extb5b2	-	
hdrb	-	
mfs	-	
mode	-	
llineg	-	
ass	-	
ibob	-	
spb5b2	-	
extb5c	-	
nsb	-	
ndb	-	
pi	-	
extb5d	-	
dplxm	-	
modemt	-	
extb6	-	
l2id	-	
uil2	-	
extb6a1	-	
modl2	-	
spb6a1	-	
q933	-	
extb6a2	-	
usl2pi	-	
extb6b	-	
ws	-	
extb7	-	
l3id	-	
uil3	-	
extb7a1	-	
modl3	-	

spb7a1	-	
extb7a2	-	
usl3pi	-	
extb7b	-	
spb7b	-	
dps	-	
extb7c	-	
pws	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Llcmp_BS2xOrBS4x_UDI	
Structured Type:	LLCMP	
Derivation Path:		
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.2.1 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iei	'01111100'B	
iel	('06'O, '07'O, '08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'01000'B	
extb3a	'1'B IF_PRESENT	
negind	*	
spb3a	'000000'B IF_PRESENT	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb4_1	-	
rateml	-	
extb5	'0'B	
l1id	'01'B	
uil1	'00001'B	
extb5a	'0'B	
sb	'1'B	
neg	'0'B	
ur	('00010'B, '00011'B, '00101'B, '01000'B, '10111'B, '11110'B)	
extb5b1	'0'B	
ir	('01'B, '10'B)	
nictx	'?'B	
nicrx	'?'B	
fctx	'0'B	
fcrx	'0'B	
spb5b1	'0'B	
extb5b2	-	
hdrb	-	
mfs	-	
mode	-	
llineg	-	
ass	-	
ibob	-	
spb5b2	-	
extb5c	'?'B	
nsb	('01'B, '11'B)	
ndb	('10'B, '11'B)	
pi	('000'B,'010'B, '011'B, '100'B, '101'B)	
extb5d	'1'B IF_PRESENT	
dplxm	*	
modemt	*	
extb6	*	
l2id	*	
uil2	*	
extb6a1	*	
modl2	*	
spb6a1	*	
q933	*	
extb6a2	*	
usl2pi	*	
extb6b	*	
ws	*	
extb7	*	
l3id	*	
uil3	*	

extb7a1	*	
modl3	*	
spb7a1	*	
extb7a2	*	
usl3pi	*	
extb7b	*	
spb7b	*	
dps	*	
extb7c	*	
pws	*	
Detailed Comments:	Corresponding to the Bearer Services 21 ... 26, 41 ... 46 Unrestrict digital information transfer capability.	

Structured Type Constraint		
Constraint Name:	Llcmp_BS2xOrBS61aOrBS81a_3100	
Structured Type:	LLCMP	
Derivation Path:	Llcmp_BS2xOrBS4x_UDI.	
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.2.2 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iel	('07'O,'08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'10000'B	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb5	'0'B	
l1id	'01'B	
uil1	'00011'B	
extb5a	'0'B	
sb	?	
neg	?	
ur	?	
extb5b1	'0'B	
ir	?	
nictx	?	
nicrx	?	
fctx	?	
fcrx	?	
spb5b1	'0'B	
extb5c	'0'B	
nsb	?	
ndb	?	
pi	?	
extb5d	'1'B	
dplxm	?	
modemt	('010001'B, '010010'B, '010011'B, '010100'B, '010111'B, '011100'B)	
Detailed Comments:	Corresponding to the Bearer Services 21 ... 26, 61 Asynchronous or 81 Asynchronous 3.1 kHz audio ex-PLMN information transfer capability.	

Structured Type Constraint		
Constraint Name:	Llcmp_BS3x_UDI	
Structured Type:	LLCMP	
Derivation Path:	Llcmp_BS2xOrBS4x_UDI.	
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.3.1 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iel	('05'O, '06'O, '07'O, '08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'01000'B	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb5	'0'B	
l1id	'01'B	
uil1	('00001'B, '01001'B)	
extb5a	'0'B	
sb	'0'B	
neg	'0'B	
ur	('00010'B, '00011'B, '00101'B, '01000'B)	
extb5b1	'?'B	
ir	('01'B, '10'B)	
nictx	'?'B	
nicrx	'?'B	
fctx	'?'B	
fcrx	'?'B	
spb5b1	'0'B	
extb5c	'0'B IF_PRESENT	
nsb	*	
ndb	*	
pi	*	
extb5d	'1'B IF_PRESENT	
dplxm	*	
modemt	*	
extb6	*	
l2id	'10'B IF_PRESENT	
uil2	('00110'B, '00111'B) IF_PRESENT	
extb7	*	
l3id	'11'B IF_PRESENT	
uil3	('00110'B, '00111'B) IF_PRESENT	
Detailed Comments:	Corresponding to the Bearer Services 31 ... 34 unrestricted digital information transfer capability.	

Structured Type Constraint		
Constraint Name:	Llcmp_BS3xOrBS61sOrBS81s_3100	
Structured Type:	LLCMP	
Derivation Path:	Llcmp_BS2xOrBS4x_UDI.	
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.3.2 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iel	('03'O,'04'O, '08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'10000'B	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb5	'?'B	
l1id	'01'B	
uil1	'00011'B	
extb5a	'0'B IF_PRESENT	
sb	*	
neg	*	
ur	*	
extb5b1	*	
ir	*	
nictx	*	
nicrx	*	
fctx	*	
fcrx	*	
spb5b1	*	
extb5c	'0'B IF_PRESENT	
nsb	*	
ndb	*	
pi	*	
extb5d	'1'B IF_PRESENT	
dplxm	*	
modemt	('010010'B, '010011'B, '010111'B, '011100'B) IF_PRESENT	
extb6	*	
l2id	'10'B IF_PRESENT	
uil2	('00110'B, '00111'B) IF_PRESENT	
extb7	*	
l3id	'11'B IF_PRESENT	
uil3	('00110'B, '00111'B) IF_PRESENT	
Detailed Comments:	Corresponding to the Bearer Services 31 ... 34, 61 Synchronous or 81 Synchronous 3.1kHz audio information transfer capability.	

Structured Type Constraint		
Constraint Name:	Llcmp_BS5x	
Structured Type:	LLCMP	
Derivation Path:	Llcmp_BS2xOrBS4x_UDI.	
Comments:	Lower Layer Compatibility IE for a setup PDU, B.2.5.1 of GSM 07.01.	
Element Name	**Element Value**	**Comments**
iel	('06'O,'07'O, '08'O, '09'O, '0A'O, '0B'O, '0C'O, '0D'O, '0E'O)	
extb3	'?'B	
cs	'00'B	
itc	'01000'B	
extb4	'1'B	
tm	'00'B	
itr	'10000'B	
extb5	'0'B	
l1id	'01'B	
uil1	'01001'B	
extb5a	'?'B	
sb	'0'B	
neg	'0'B	
ur	('00011'B, '00101'B, '01000'B)	
extb5b1	*	
ir	*	
nictx	*	
nicrx	*	
fctx	*	
fcrx	*	
spb5b1	*	
extb5c	*	
nsb	*	
ndb	*	
pi	*	
extb5d	*	
dplxm	*	
modemt	*	
extb6	?	
l2id	'10'B	
uil2	('00110'B, '00111'B)	
extb7	?	
l3id	'11'B	
uil3	('00110'B, '00111'B)	
Detailed Comments:	Corresponding to the Bearer Services 51 ... 53 unrestricted digital information transfer capability.	

Structured Type Constraint		
Constraint Name:	LocAreaId(mcc, mnc, lac:OCTETSTRING)	
Structured Type:	LAI	
Derivation Path:		
Comments:	Cell A default value for L3 test	
Element Name	**Element Value**	**Comments**
iei	-	
mcc	mcc	
mnc	mnc	
lac	lac	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	LocUpType(locup:B_2)	
Structured Type:	LUT	
Derivation Path:		
Comments:	location updating type	
Element Name	**Element Value**	**Comments**
foreq	'?'B	
sprb	'0'B	
lut	locup	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiImsi_01	
Structured Type:	MI	
Derivation Path:		
Comments:	the TSPX_IMSI is the IMSI of the MS under test	
Element Name	**Element Value**	**Comments**
iei	-	
iel	OC_IntToOct(((LENGTH_OF(TSPX_IMSI))/2 + 1),1)	
idigit_1	OC_FirstDigi(TSPX_IMSI)	
oei	OC_OeBit(TSPX_IMSI)	
toi	'001'B	
idigits_other	OC_OtherDigi(TSPX_IMSI)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiImsi_01iei	
Structured Type:	MI	
Derivation Path:		
Comments:	the TSPX_IMSI is the IMSI of the MS under test	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	OC_IntToOct(((LENGTH_OF(TSPX_IMSI))/2 + 1),1)	
idigit_1	OC_FirstDigi(TSPX_IMSI)	
oei	OC_OeBit(TSPX_IMSI)	
toi	'001'B	
idigits_other	OC_OtherDigi(TSPX_IMSI)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiImsi_02	
Structured Type:	MI	
Derivation Path:		
Comments:	This constraint is used in paging filling message. No mobile is paged (Type of identity is set to '000'B, i.e. no identity).	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'01'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'000'B	
idigits_other	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiImsi_r01	
Structured Type:	MI	
Derivation Path:		
Comments:	An another IMSI differing from MiImsi_01 for RR testing.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'07'O	
idigit_1	'0000'B	
oei	'1'B	
toi	'001'B	
idigits_other	'981032547698'O	
Detailed Comments:	IMSI has 13 digits: '08 90 12 34 56 78 9'	

Structured Type Constraint		
Constraint Name:	MiImsi_r01iei	
Structured Type:	MI	
Derivation Path:		
Comments:	An another IMSI differing from MiImsi_01 for RR testing.	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	'07'O	
idigit_1	'0000'B	
oei	'1'B	
toi	'001'B	
idigits_other	'981032547698'O	
Detailed Comments:	IMSI has 13 digits: '08 90 12 34 56 78 9'	

Structured Type Constraint		
Constraint Name:	MiTmsi_01	
Structured Type:	MI	
Derivation Path:		
Comments:	default TMSI for testing.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'100'B	
idigits_other	TSPX_TMSI	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_01iei	
Structured Type:	MI	
Derivation Path:		
Comments:	default TMSI for testing.	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'100'B	
idigits_other	TSPX_TMSI	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_02	
Structured Type:	MI	
Derivation Path:		
Comments:	A new TMSI	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'100'B	
idigits_other	TSPX_TMSI1	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_02iei	
Structured Type:	MI	
Derivation Path:		
Comments:	A new TMSI	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'100'B	
idigits_other	TSPX_TMSI1	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_r01	
Structured Type:	MI	
Derivation Path:		
Comments:	A TMSI differs from MiTmsi_01	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'100'B	
idigits_other	OC_IncTmsi(TSPX_TMSI, '01'O)	
Detailed Comments:	Note: the TSPX_TMSI + '01'O shall not be identical to TSPX_TMSI1	

Structured Type Constraint		
Constraint Name:	MiTmsi_r02	
Structured Type:	MI	
Derivation Path:		
Comments:	default TMSI for testing with toi setting to no id.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'000'B	
idigits_other	TSPX_TMSI	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MiTmsi_r02iei	
Structured Type:	MI	
Derivation Path:		
Comments:	default TMSI for testing with toi setting to no id.	
Element Name	**Element Value**	**Comments**
iei	'00010111'B	
iel	'05'O	
idigit_1	'1111'B	
oei	'0'B	
toi	'000'B	
idigits_other	TSPX_TMSI	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_01	
Structured Type:	MA	
Derivation Path:		
Comments:	non hopping	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'00'O	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_01iei	
Structured Type:	MA	
Derivation Path:		
Comments:	non hopping	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'00'O	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_02	
Structured Type:	MA	
Derivation Path:	MoblAllc_01.	
Comments:	hopping	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'01'O	
mac_8n	'00000010'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_04	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping used in TC_26_6_4_2_2.	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'01'O	
mac_8n	'00001110'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_A0	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with 0001 1111 1111 1111 1111 indicates all of the frequencies in CA of cell A with ARFCN_list={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114}	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	'11111111'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_A1	
Structured Type:	MA	
Derivation Path:		
Comments:	indicates all of the frequencies in CA of cell A with ARFCN_list={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114} except for the BCCH frequency 20.	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	'11111011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_A3d	
Structured Type:	MA	
Derivation Path:		
Comments:	indicates all of the frequencies in CA of cell A ARFCN_list={734,741,747,754,759,762,766,767,773,775,779,782,791,798,829,832,844} except 747 and 767.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	'01111011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_252	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with 0001 1100 indicates ARFCN_list={40,66,73,74,75,76,108,114} to the ARFCN_list={73,74,75}.	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'01'O	
mac_8n	'00011100'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_252d	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with 0111 0000 0 List_ARFCN={761,764,771,779,782,791,798,829,832} to the ARFCN_list={791,798,829}.	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'02'O	
mac_8n	'00000000'B	
mac_7n	'11100000'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_22	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with all of f's from cell allocation of cell B with except of {764,832,844}	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'02'O	
mac_8n	'00111111'B	
mac_7n	'01111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_281	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with indicates to the ARFCN_list={73,74,75} of cell allocation A.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000000'B	
mac_7n	'00111000'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_281d	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with indicates to the ARFCN_list={773,775,779} of cell allocation A.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000000'B	
mac_7n	'00000111'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_281e	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping sequence with EGSM frequencies. The frequencies which are used in the mobile hopping sequence are {0, 80, 1005, 1010} .	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'01'O	
mac_8n	'10111000'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_281e2	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with indicates to the ARFCN_list={20,66,78} of E-GSM cell allocation with ARFCN_List={20,40,66,73,74,75,76,77,78,79,108,114}	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	'00000001'B	
mac_7n	'00000101'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_282	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with indicates to the Complete Cell Allocation of cell A with except for BCCH-f	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111111'B	
mac_6n	'11111011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_B1	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with 1111 1111 1111 1111 ARFCN_list={14,18,22,24,30,31,38,40,60,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	'11111111'B	
mac_7n	'11111111'B	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_B1iei	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with ARFCN_list={14,18,22,24,30,31,38,40,60,66,73,74,75,76,108,114}.	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'02'O	
mac_8n	'10100111'B	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_Be1(par_ma: BITSTRING)	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with par_ma	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'01'O	
mac_8n	par_ma	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_Be2(par_ma1: BITSTRING; par_ma2: BITSTRING)	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with par_ma1 and par_ma2	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
mac_8n	par_ma1	
mac_7n	par_ma2	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_Be2iei(par_ma1: BITSTRING; par_ma2: BITSTRING)	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with par_ma1 and par_ma2	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'02'O	
mac_8n	par_ma1	
mac_7n	par_ma2	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_A2	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with default Cell Allocation of HO cases and only one frequency in hopping	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_A2iei	
Structured Type:	MA	
Derivation Path:		
Comments:	hopping with 1000 0000 0000 0000 indicates from default Cell Allocation of HO cases with the ARFCN_List={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114} the ARFCN_list={114}	
Element Name	**Element Value**	**Comments**
iei	'01110010'B	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'00000000'B	
mac_6n	'00000000'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_20_A3	
Structured Type:	MA	
Derivation Path:		
Comments:	indicates all of the frequencies in CA of cell A ARFCN_list={10,17,20,26,34,42,45,46,52,59,66,73,74,75,76,108,114} except 20 and 52.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'03'O	
mac_8n	'00000001'B	
mac_7n	'11111110'B	
mac_6n	'11111011'B	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MoblAllc_omit	
Structured Type:	MA	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
mac_8n	-	
mac_7n	-	
mac_6n	-	
mac_5n	-	
mac_4n	-	
mac_3n	-	
mac_2n	-	
mac_1n	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_01	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing no measurement results.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	('000'B, '111'B)	
rxlev_nc1	'000000'B	
bcchfrq_nc1	'00000'B	
bsic_nc1	'000000'B	
rxlev_nc2	'000000'B	
bcchfrq_nc2	'00000'B	
bsic_nc2	'000000'B	
rxlev_nc3	'000000'B	
bcchfrq_nc3	'00000'B	
bsic_nc3	'000000'B	
rxlev_nc4	'000000'B	
bcchfrq_nc4	'00000'B	
bsic_nc4	'000000'B	
rxlev_nc5	'000000'B	
bcchfrq_nc5	'00000'B	
bsic_nc5	'000000'B	
rxlev_nc6	'000000'B	
bcchfrq_nc6	'00000'B	
bsic_nc6	'000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_02	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE matching any value.	
Element Name	**Element Value**	**Comments**
ba_used	?	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	?	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	?	
rxlev_nc1	*	
bcchfrq_nc1	*	
bsic_nc1	*	
rxlev_nc2	*	
bcchfrq_nc2	*	
bsic_nc2	*	
rxlev_nc3	*	
bcchfrq_nc3	*	
bsic_nc3	*	
rxlev_nc4	*	
bcchfrq_nc4	*	
bsic_nc4	*	
rxlev_nc5	*	
bcchfrq_nc5	*	
bsic_nc5	*	
rxlev_nc6	*	
bcchfrq_nc6	*	
bsic_nc6	*	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_03	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing 6 measurement results.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'110'B	
rxlev_nc1	?	
bcchfrq_nc1	?	
bsic_nc1	?	
rxlev_nc2	?	
bcchfrq_nc2	?	
bsic_nc2	?	
rxlev_nc3	?	
bcchfrq_nc3	?	
bsic_nc3	?	
rxlev_nc4	?	
bcchfrq_nc4	?	
bsic_nc4	?	
rxlev_nc5	?	
bcchfrq_nc5	?	
bsic_nc5	?	
rxlev_nc6	?	
bcchfrq_nc6	?	
bsic_nc6	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_03e1	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing 6 measurement results for EGSM.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'110'B	
rxlev_nc1	?	
bcchfrq_nc1	'00001'B	
bsic_nc1	'001011'B	
rxlev_nc2	?	
bcchfrq_nc2	'00010'B	
bsic_nc2	'001111'B	
rxlev_nc3	?	
bcchfrq_nc3	'00100'B	
bsic_nc3	'001101'B	
rxlev_nc4	?	
bcchfrq_nc4	'00101'B	
bsic_nc4	'001001'B	
rxlev_nc5	?	
bcchfrq_nc5	'00110'B	
bsic_nc5	'001111'B	
rxlev_nc6	?	
bcchfrq_nc6	'00111'B	
bsic_nc6	'001101'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_03e2	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing 6 measurement results for EGSM.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'110'B	
rxlev_nc1	?	
bcchfrq_nc1	'00010'B	
bsic_nc1	'001011'B	
rxlev_nc2	?	
bcchfrq_nc2	'00011'B	
bsic_nc2	'001111'B	
rxlev_nc3	?	
bcchfrq_nc3	'00100'B	
bsic_nc3	'001101'B	
rxlev_nc4	?	
bcchfrq_nc4	'00101'B	
bsic_nc4	'001001'B	
rxlev_nc5	?	
bcchfrq_nc5	'00110'B	
bsic_nc5	'001111'B	
rxlev_nc6	?	
bcchfrq_nc6	'00001'B	
bsic_nc6	'001100'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_03e3	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing 6 measurement results for EGSM.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'110'B	
rxlev_nc1	?	
bcchfrq_nc1	'00010'B	
bsic_nc1	'001011'B	
rxlev_nc2	?	
bcchfrq_nc2	'00011'B	
bsic_nc2	'001111'B	
rxlev_nc3	?	
bcchfrq_nc3	'00110'B	
bsic_nc3	'001001'B	
rxlev_nc4	?	
bcchfrq_nc4	'01000'B	
bsic_nc4	'001111'B	
rxlev_nc5	?	
bcchfrq_nc5	'01000'B	
bsic_nc5	'001101'B	
rxlev_nc6	?	
bcchfrq_nc6	'00001'B	
bsic_nc6	'001100'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_03e4	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing 6 measurement results for EGSM.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'110'B	
rxlev_nc1	?	
bcchfrq_nc1	'00010'B	
bsic_nc1	'001011'B	
rxlev_nc2	?	
bcchfrq_nc2	'00011'B	
bsic_nc2	'001111'B	
rxlev_nc3	?	
bcchfrq_nc3	'00110'B	
bsic_nc3	'001101'B	
rxlev_nc4	?	
bcchfrq_nc4	'00111'B	
bsic_nc4	'001001'B	
rxlev_nc5	?	
bcchfrq_nc5	'01000'B	
bsic_nc5	'001111'B	
rxlev_nc6	?	
bcchfrq_nc6	'00001'B	
bsic_nc6	'001100'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_03e5	
Structured Type:	MSRR	
Derivation Path:		
Comments:	A measurement results IE containing 6 measurement results for EGSM.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'110'B	
rxlev_nc1	?	
bcchfrq_nc1	'00010'B	
bsic_nc1	'001011'B	
rxlev_nc2	?	
bcchfrq_nc2	'00011'B	
bsic_nc2	'001111'B	
rxlev_nc3	?	
bcchfrq_nc3	'00101'B	
bsic_nc3	'001101'B	
rxlev_nc4	?	
bcchfrq_nc4	'00110'B	
bsic_nc4	'001001'B	
rxlev_nc5	?	
bcchfrq_nc5	'00111'B	
bsic_nc5	'001111'B	
rxlev_nc6	?	
bcchfrq_nc6	'00001'B	
bsic_nc6	'001100'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_04	
Structured Type:	MSRR	
Derivation Path:	MsrResult_03.	
Comments:	A measurement results IE containing 4 measurement results.	
Element Name	**Element Value**	**Comments**
no_nc	'100'B	
bcchfrq_nc1	('00000'B, '00001'B, '00011'B, '00100'B)	
bsic_nc1	('001011'B, '001111'B, '001001'B)	
bcchfrq_nc2	('00000'B, '00001'B, '00011'B, '00100'B)	
bsic_nc2	('001011'B, '001111'B, '001001'B)	
bcchfrq_nc3	('00000'B, '00001'B, '00011'B, '00100'B)	
bsic_nc3	('001011'B, '001111'B, '001001'B)	
bcchfrq_nc4	('00000'B, '00001'B, '00011'B, '00100'B)	
bsic_nc4	('001011'B, '001111'B, '001001'B)	
rxlev_nc5	'000000'B	
bcchfrq_nc5	'00000'B	
bsic_nc5	'000000'B	
rxlev_nc6	'000000'B	
bcchfrq_nc6	'00000'B	
bsic_nc6	'000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_04e	
Structured Type:	MSRR	
Derivation Path:	MsrResult_03.	
Comments:	A measurement results IE containing 3 measurement results for EGSM	
Element Name	**Element Value**	**Comments**
no_nc	'100'B	
rxlev_nc1	?	
bcchfrq_nc1	'00001'B	
bsic_nc1	'001011'B	
rxlev_nc2	?	
bcchfrq_nc2	'00011'B	
bsic_nc2	'001101'B	
rxlev_nc3	?	
bcchfrq_nc3	'00110'B	
bsic_nc3	'001001'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_05	
Structured Type:	MSRR	
Derivation Path:	MsrResult_03.	
Comments:	A measurement results IE containing 6 measurement results and DTX was used.	
Element Name	**Element Value**	**Comments**
dtx_used	'1'B	
bcchfrq_nc1	('00000'B, '00011'B, '00100'B, '00101'B, '00110'B, '00111'B)	
bsic_nc1	('001011'B, '001101'B, '001111'B, '001001'B)	
bcchfrq_nc2	('00000'B, '00011'B, '00100'B, '00101'B, '00110'B, '00111'B)	
bsic_nc2	('001011'B, '001101'B, '001111'B, '001001'B)	
bcchfrq_nc3	('00000'B, '00011'B, '00100'B, '00101'B, '00110'B, '00111'B)	
bsic_nc3	('001011'B, '001101'B, '001111'B, '001001'B)	
bcchfrq_nc4	('00001'B, '00100'B, '00101'B, '00110'B, '00111'B, '01000'B)	
bsic_nc4	('001011'B, '001101'B, '001111'B, '001001'B)	
bcchfrq_nc5	('00001'B, '00100'B, '00101'B, '00110'B, '00111'B, '01000'B)	
bsic_nc5	('001011'B, '001101'B, '001111'B, '001001'B)	
bcchfrq_nc6	('00001'B, '00100'B, '00101'B, '00110'B, '00111'B, '01000'B)	
bsic_nc6	('001011'B, '001101'B, '001111'B, '001001'B)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_06	
Structured Type:	MSRR	
Derivation Path:	MsrResult_03.MsrResult_05.	
Comments:	A measurement results IE containing 6 measurement results and DTX is not checked.	
Element Name	**Element Value**	**Comments**
dtx_used	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	MsrResult_07	
Structured Type:	MSRR	
Derivation Path:	MsrResult_03.	
Comments:	A measurement results IE containing 2 measurement results.	
Element Name	**Element Value**	**Comments**
ba_used	'1'B	
dtx_used	'0'B	
rxlev_fsc	?	
spr1	'0'B	
meas_valid	'0'B	
rxlev_ssc	?	
spr2	'0'B	
rxqual_fsc	?	
rxqual_ssc	?	
no_nc	'010'B	
rxlev_nc1	?	
bcchfrq_nc1	('00000'B, '00010'B)	
bsic_nc1	('001011'B, '001101'B)	
rxlev_nc2	?	
bcchfrq_nc2	('00000'B, '00010'B)	
bsic_nc2	('001011'B, '001101'B)	
rxlev_nc3	'000000'B	
bcchfrq_nc3	'00000'B	
bsic_nc3	'000000'B	
rxlev_nc4	'000000'B	
bcchfrq_nc4	'00000'B	
bsic_nc4	'000000'B	
rxlev_nc5	'000000'B	
bcchfrq_nc5	'00000'B	
bsic_nc5	'000000'B	
rxlev_nc6	'000000'B	
bcchfrq_nc6	'00000'B	
bsic_nc6	'000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Mtdif_01(ta_a, td_b:INTEGER)	
Structured Type:	MTDIF	
Derivation Path:		
Comments:	ta_a: timing advance in cell A, td_b: time difference leg in cell B.	
Element Name	**Element Value**	**Comments**
iei	'01110111'B	
iel	'03'O	
value	(OC_IntToOct((((2*td_b + ta_a) MOD 2097152)-2), 3), OC_IntToOct((((2*td_b + ta_a) MOD 2097152)-1), 3), OC_IntToOct(((2*td_b + ta_a) MOD 2097152), 3), OC_IntToOct((((2*td_b + ta_a) MOD 2097152)+1), 3), OC_IntToOct((((2*td_b + ta_a) MOD 2097152)+2), 3))	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	NT_BothT_BothNTorT_BothT_BothNT	
Structured Type:	CE	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
TorBothTorBothNT	('00'B, '10'B, '11'B)	
NTorBothTorBothNT	('01'B, '10'B, '11'B)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Pcmd_19(powerlevel:BITSTRING)	
Structured Type:	PCMD	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
sprb	'000'B	
pl	powerlevel	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Pcmd_20(powerlevel:B_5)	
Structured Type:	PCMD	
Derivation Path:		
Comments:	PowerCmd used in synchronized and non synchronized HO cases.	
Element Name	**Element Value**	**Comments**
sprb	'011'B	
pl	powerlevel	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Pm(pgm:B_2)	
Structured Type:	PM	
Derivation Path:		
Comments:	normal paging mode	
Element Name	**Element Value**	**Comments**
sprb	'00'B	
pgm	pgm	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Pm_02	
Structured Type:	PM	
Derivation Path:		
Comments:	normal paging mode with arbitrary spare bits.	
Element Name	**Element Value**	**Comments**
sprb	'11'B	
pgm	'00'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ProgInd_omit	
Structured Type:	PI	
Derivation Path:		
Comments:	Omitted Progress Indicator.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
extb3	-	
cs	-	
spb	-	
loc	-	
extb4	-	
prd	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ProgInd_01	
Structured Type:	PI	
Derivation Path:		
Comments:	Progress Indicator with progress description #8 (inband info now available)	
Element Name	**Element Value**	**Comments**
iei	'00011110'B	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'1'B	
loc	'0000'B	
extb4	'1'B	
prd	'0001000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ProgInd_02	
Structured Type:	PI	
Derivation Path:		
Comments:	Progress Indicator containing progress indicator #4.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
loc	'0001'B	
extb4	'1'B	
prd	'0000100'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ProgInd_03	
Structured Type:	PI	
Derivation Path:		
Comments:	Progress Indicator containing progress indicator #8.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
loc	'0001'B	
extb4	'1'B	
prd	'0001000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	ProgInd_03iei	
Structured Type:	PI	
Derivation Path:		
Comments:	Progress Indicator containing progress indicator #8.	
Element Name	**Element Value**	**Comments**
iei	'00011110'B	
iel	'02'O	
extb3	'1'B	
cs	'11'B	
spb	'0'B	
loc	'0001'B	
extb4	'1'B	
prd	'0001000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RachCntrlPara(maxtx:B_2; txint:B_4; re:B_1)	
Structured Type:	RACHCP	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
maxrtx	maxtx	
txint	txint	
cba	'0'B	
re	re	
acc_2	'00000'B	
ec	'0'B	
acc_1	'0000000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RachCntrlPara_cba(maxtx:B_2; txint:B_4; re:B_1; cba:B_1; acc:B_5)	
Structured Type:	RACHCP	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
maxrtx	maxtx	
txint	txint	
cba	cba	
re	re	
acc_2	acc	
ec	'0'B	
acc_1	'0000000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RachCntrlPara_01	
Structured Type:	RACHCP	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
maxrtx	'00'B	
txint	'0010'B	
cba	'0'B	
re	'0'B	
acc_2	'00000'B	
ec	'0'B	
acc_1	'0000000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	RachCntrlPara_r01	
Structured Type:	RACHCP	
Derivation Path:		
Comments:	Call reestablishment is not allowed in the cell.	
Element Name	**Element Value**	**Comments**
maxrtx	'00'B	
txint	'0010'B	
cba	'0'B	
re	'1'B	
acc_2	'00000'B	
ec	'0'B	
acc_1	'0000000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SI3RestOctetPara(pt:B_5; cro:B_6; to:B_3; po:B_2; cbq:B_1)	
Structured Type:	SI3RO	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
pi	'1'B	
cbq	cbq	
cro	cro	
to	to	
pt	pt	
poi	'1'B	
po	po	
ti	'0'B	
ecsc	'0'B	
spr1	'000'B	
spr2	'00101011'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SI3RestOctetPara_ti(pt:B_5; cro:B_6; to:B_3; po:B_2)	
Structured Type:	SI3RO	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
pi	'1'B	
cbq	'0'B	
cro	cro	
to	to	
pt	pt	
poi	'1'B	
po	po	
ti	'1'B	
ecsc	'0'B	
spr1	'000'B	
spr2	'00101011'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SI4RestOctetDef	
Structured Type:	SI4RO	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
pi	'0'B	
cbq	'0'B	
cro	'101011'B	
to	'001'B	
pt	'01011'B	
poi	'0'B	
po	'01'B	
spr1	'01011'B	
spr2	'00101011'B	
spr3	'00101011'B	
spr4	'00101011'B	
spr5	'00101011'B	
spr6	'00101011'B	
spr7	'00101011'B	
spr8	'00101011'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SI4RestOctetPara(pt:B_5; cro:B_6; to:B_3; po:B_2; cbq:B_1)	
Structured Type:	SI4RO	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
pi	'1'B	
cbq	cbq	
cro	cro	
to	to	
pt	pt	
poi	'1'B	
po	po	
spr1	'00000'B	
spr2	'00101011'B	
spr3	'00101011'B	
spr4	'00101011'B	
spr5	'00101011'B	
spr6	'00101011'B	
spr7	'00101011'B	
spr8	'00101011'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	SI7RestOctetPara(pt:B_5; cro:B_6; to:B_3; po:B_2; cbq:B_1)	
Structured Type:	SI7RO	
Derivation Path:		
Comments:	Default value for L 3 testing.	
Element Name	**Element Value**	**Comments**
pi	'1'B	
cbq	cbq	
cro	cro	
to	to	
pt	pt	
poi	'1'B	
po	po	
spr1	'00000'B	
spr2	'00101011'B	
spr3	'00101011'B	
spr4	'00101011'B	
spr5	'00101011'B	
spr6	'00101011'B	
spr7	'00101011'B	
spr8	'00101011'B	
spr9	'00101011'B	
spr10	'00101011'B	
spr11	'00101011'B	
spr12	'00101011'B	
spr13	'00101011'B	
spr14	'00101011'B	
spr15	'00101011'B	
spr16	'00101011'B	
spr17	'00101011'B	
spr18	'00101011'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_01	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1270201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11881013C8202008C830164840200FA85010086010087020258000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_02	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1250201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1168101008201008 30164840203E8850100860100870100000 00'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_03	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1290201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11A810209C48202 00A0830200C8840213888501008601008 70202580000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_04	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1240201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11581010A82010A830164840100850164860 10A87010A0000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_05	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1270201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11881017D8202012C830164840200FA85016486010A8702012C0000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_06	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1240201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1158101008201008301008401008501008601008701000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_07	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1250201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1168101008201008301648402 03E885010086010087010000 00'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_08	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1270201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11881013C8202008C830164840200FA85010086010087022580000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_09	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1240201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11581010A82010A830164840100850100860100870010A0000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_10	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1270201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A11881017D8202012C830164840200FA85010086010087020 12C0000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_11	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1260201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1178101648202011883016484016485010086010087020258 0000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_12	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1260201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1178101648202008C830164840132850100860100870202580000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_13	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1250201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1168101648202019083016484010085010086010087010000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_14	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1190201'O	
invokeId	'00'O	
comp_part2	'02017D3011800172A1808101648202019 08301640000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_15	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1250201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1168101468202019 08301648401008501008601008701000 00'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_16	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1260201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1178102008282020190830164840100850100860100870100000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_17	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1260201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A117810200BE8202019083016484010085010086010087010000000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_18	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1260201'O	
invokeId	'00'O	
comp_part2	'02017D3080800172A1178102012282020190830164840100850100860100870100000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_19	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1800201'O	
invokeId	'00'O	
comp_part2	'02017D301B800172A116810164820202268301648401648501008601008701640000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_20	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging	
Element Name	**Element Value**	**Comments**
comp_part1	'A1800201'O	
invokeId	'00'O	
comp_part2	'02017D301B800172A11681010A8202012C830164840100850100860100870100000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	FwdChAdvSS_21	
Structured Type:	Component_T	
Derivation Path:		
Comments:	AoC- Charging, charging value is 0.	
Element Name	**Element Value**	**Comments**
comp_part1	'A1800201'O	
invokeId	'00'O	
comp_part2	'02017D301A800172A115810100820100830100840100850100860100870100000'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	NotificationSS_05	
Structured Type:	Component_T	
Derivation Path:		
Comments:	BI	
Element Name	**Element Value**	**Comments**
comp_part1	'A10E0201'O	
invokeId	'00'O	
comp_part2	'02011030068101998401 07'O	
comp_part3	-	
comp_part4	-	
comp_part5	-	
comp_part6	-	
comp_part7	-	
comp_part8	-	
comp_part9	-	
comp_part10	-	
comp_part11	-	
comp_part12	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Rqr1(Rr: BITSTRING; Fn: FN)	
Structured Type:	RQR	
Derivation Path:		
Comments:	not address the MS under test	
Element Name	**Element Value**	**Comments**
ra	INT_TO_BIT((BIT_TO_INT(Rr) + 1), 8)	
fn	OC_FnInc(Fn, 2)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Rqr2(Rr: BITSTRING; Fn: FN)	
Structured Type:	RQR	
Derivation Path:		
Comments:	To address the MS under test	
Element Name	**Element Value**	**Comments**
ra	Rr	
fn	Fn	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Rqr3	
Structured Type:	RQR	
Derivation Path:		
Comments:	not pertaining to the MS under test	
Element Name	**Element Value**	**Comments**
ra	'00000000'B	
fn	Fn_01	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Signal_01	
Structured Type:	SIGNAL	
Derivation Path:		
Comments:	signal value is arbitrarily selected.	
Element Name	**Element Value**	**Comments**
iei	'00110100'B	
sigv	'00000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Signal_02	
Structured Type:	SIGNAL	
Derivation Path:		
Comments:	Signal IE with value #7 "call waiting tone on"	
Element Name	**Element Value**	**Comments**
iei	'00110100'B	
sigv	'00000111'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	StartingTm_01(fn:FN)	
Structured Type:	STRT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	'01111100'B	
fn	fn	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	StartingTm_omit	
Structured Type:	STRT	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
fn	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TimingAdv(bits:INTEGER)	
Structured Type:	TA	
Derivation Path:		
Comments:	Time Advance value is pameterized.	
Element Name	**Element Value**	**Comments**
iei	-	
sprb	'00'B	
value	INT_TO_BIT(bits, 6)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TimingAdv_iei(bits:INTEGER)	
Structured Type:	TA	
Derivation Path:		
Comments:	0 time advance.	
Element Name	**Element Value**	**Comments**
iei	'01111101'B	
sprb	'00'B	
value	INT_TO_BIT(bits, 6)	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TimingAdv_inv	
Structured Type:	TA	
Derivation Path:		
Comments:	containing arbitrary spare bits	
Element Name	**Element Value**	**Comments**
iei	-	
sprb	'11'B	
value	'000000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TimingAdv_omit	
Structured Type:	TA	
Derivation Path:		
Comments:	Time Advance value is pameterized.	
Element Name	**Element Value**	**Comments**
iei	-	
sprb	-	
value	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_01	
Structured Type:	TI	
Derivation Path:		
Comments:	used by the MS in the transaction initiated by the test system.	
Element Name	**Element Value**	**Comments**
ti_f	'1'B	
ti_v	'000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_02	
Structured Type:	TI	
Derivation Path:		
Comments:	used in the messages sent to the MS in the transaction initiated by test system.	
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	'000'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_03	
Structured Type:	TI	
Derivation Path:		
Comments:	used in the messages sent to the MS in the transaction initiated by test system.	
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	'110'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_04	
Structured Type:	TI	
Derivation Path:		
Comments:	arbitrary value	
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	'011'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_05	
Structured Type:	TI	
Derivation Path:		
Comments:	used by the MS in the transaction initiated by the test system.	
Element Name	**Element Value**	**Comments**
ti_f	'1'B	
ti_v	'110'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_06	
Structured Type:	TI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	'111'B	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_07(ti :TI_V)	
Structured Type:	TI	
Derivation Path:		
Comments:	used in the messages sent to the MS in the transaction initiated by the MS.	
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	ti	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_08(ti:TI_V)	
Structured Type:	TI	
Derivation Path:		
Comments:	used in the messages sent to the MS in the transaction initiated by the system simulator.	
Element Name	**Element Value**	**Comments**
ti_f	'1'B	
ti_v	ti	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	TI_09	
Structured Type:	TI	
Derivation Path:		
Comments:	used by the MS in the transaction initiated by the MS.	
Element Name	**Element Value**	**Comments**
ti_f	'0'B	
ti_v	?	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Tmsi_01	
Structured Type:	TMSI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
tmsi_val	TSPX_TMSI	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	Tmsi_r01	
Structured Type:	TMSI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
tmsi_val	OC_IncTmsi(TSPX_TMSI, '01'O)	
Detailed Comments:	Note: the TSPX_TMSI + '01'O shall not be identical to TSPX_TMSI1	

Structured Type Constraint		
Constraint Name:	Tmsi_r03	
Structured Type:	TMSI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
tmsi_val	OC_IncTmsi(TSPX_TMSI, '02'O)	
Detailed Comments:	Note: the TSPX_TMSI + '02'O shall not be identical to TSPX_TMSI1	

Structured Type Constraint		
Constraint Name:	Tmsi_r04	
Structured Type:	TMSI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
tmsi_val	OC_IncTmsi(TSPX_TMSI, '03'O)	
Detailed Comments:	Note: the TSPX_TMSI + '03'O shall not be identical to TSPX_TMSI1	

Structured Type Constraint		
Constraint Name:	Tmsi_r05	
Structured Type:	TMSI	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
iei	-	
tmsi_val	OC_IncTmsi(TSPX_TMSI, '04'O)	
Detailed Comments:	Note: the TSPX_TMSI + '04'O shall not be identical to TSPX_TMSI1	

Structured Type Constraint		
Constraint Name:	UuInfo_omit	
Structured Type:	UU	
Derivation Path:		
Comments:	Omitted UUInfo.	
Element Name	**Element Value**	**Comments**
iei	-	
iel	-	
uupd	-	
uui	-	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	UuInfo	
Structured Type:	UU	
Derivation Path:		
Comments:	Delete previously submitted short message	
Element Name	**Element Value**	**Comments**
iei	'01111110'B	
iel	OC_IntToOct(1+ LENGTH_OF(TSPX_UuInfo), 1)	
uupd	TSPX_Uupd	
uui	TSPX_UuInfo	
Detailed Comments:		

Structured Type Constraint		
Constraint Name:	V22bisOrV26ter	
Structured Type:	MODEMTYPE	
Derivation Path:		
Comments:		
Element Name	**Element Value**	**Comments**
V22bisOrV26ter	('00011'B, '00101'B)	
Detailed Comments:		

ASN1 type constraints

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CF all synchronous services
Constraint Value	

activateSSComponents activateSS_InvokeComp
{
invokeID
?,
localValue
12,
ss_ForBS
{
ss_Code '20'H,
basicService bearerService '68'H
}
}	

Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFU all basic services
Constraint Value	

activateSSComponents activateSS_InvokeComp
{
invokeID
?,
localValue
12,
ss_ForBS
{
ss_Code '21'H
}
}	

Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	BAOC all synchronous services

Constraint Value

```
activateSSComponents      activateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
12,
                                                                      ss_ForBS

{
ss_Code                           '92'H,
basicService          bearerService             '68'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	BIC-Roam

Constraint Value

```
activateSSComponents      activateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
12,
                                                                      ss_ForBS

{
ss_Code                           '9B'H  -- BIC-Roam
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_05
ASN.1 Type:	Component
Derivation Path:	
Comments:	BOIC

Constraint Value

```
activateSSComponents      activateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
12,
                                                                      ss_ForBS

{
ss_Code                           '93'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ActivateSS_06
ASN.1 Type:	Component
Derivation Path:	
Comments:	BAIC
Constraint Value	

activateSSComponents activateSS_InvokeComp
{
invokeID
?,
localValue
12,
ss_ForBS
{
ss_Code '9A'H
}
}	

Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	BldMptySS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	build multiparty request
Constraint Value	

buildMPTYComponents buildMPTY_InvokeComp
{
invokeID
?,
localValue
124
}	

Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	Comps_01(comp:Component)
ASN.1 Type:	Components
Derivation Path:	
Comments:	A received FIE may contain several components, but at least contains "comp".
Constraint Value	
SUPERSET({ comp})	
Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFC for speech

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '28'H,
basicService   teleservice   '10'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRc for all facsimile

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '2B'H,
basicService   teleservice   '60'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	deactivation for barring

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '90'H,
basicService   teleservice   '10'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	deactivation for barring of outging calls

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '91'H,
basicService   teleservice   '60'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_05
ASN.1 Type:	Component
Derivation Path:	
Comments:	deactivation for barring of incoming calls

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{
ss_Code       '99'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	DeactivateSS_06
ASN.1 Type:	Component
Derivation Path:	
Comments:	deactivation for BOICExHC

Constraint Value

```
deactivateSSComponents      deactivateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
13,
                                                                      ss_ForBS

{

ss_Code       '94'H

}
                                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	EraseSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFC for all facsimile

Constraint Value

```
eraseSSComponents      eraseSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
11,
                                                                      ss_ForBS

{

ss_Code       '28'H,

basicService   teleservice   '60'H

}
                                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	EraseSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRc for all basic services

Constraint Value

```
eraseSSComponents      eraseSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
11,
                                                                      ss_ForBS

{

ss_Code       '2B'H

}
                                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	EraseSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFU for speech

Constraint Value

```
eraseSSComponents      eraseSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
11,
                                                                      ss_ForBS

{
ss_Code      '21'H,
basicService   teleservice   '10'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	EraseSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRy for all facsimile

Constraint Value

```
eraseSSComponents      eraseSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
11,
                                                                      ss_ForBS

{
ss_Code      '2A'H,
basicService   teleservice   '60'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	FwdChAdvRslt_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
forwardChargeAdviceComponents     forwardChargeAdvice_ReturnResultComp
            {
                                                                      invokeID
0,
                                                                      result
*

               }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	GetPasswdRslt_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	
Constraint Value	
getPasswordComponents getPassword_ReturnResultComp { invokeID 0, result { localValue 18, currentPassword "1234" } }	
Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	GetPasswdRslt_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	
Constraint Value	
getPasswordComponents getPassword_ReturnResultComp { invokeID 0, result { localValue 18, currentPassword "9876" } }	
Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	GetPasswdRslt_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	
Constraint Value	
getPasswordComponents getPassword_ReturnResultComp { invokeID 0, result { localValue 18, currentPassword "9877" } }	
Detailed Comments:	

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFB for all basic services

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp       {
    invokeID                      ?,
    localValue                   14,
    ss_ForBS     {

ss_Code      '29'H
             }
                                                            }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRy for Speech

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
14,
                                                                      ss_ForBS

{
ss_Code      '2A'H, --CNFRy
basicService   teleservice   '10'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRc for all basic services

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
14,
                                                                      ss_ForBS

{
ss_Code      '2B'H
}
                                                                     }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFB for all facsimile

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
14,
                                                                  ss_ForBS

{
ss_Code      '29'H,
basicService   teleservice   '60'H
}
                                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_05
ASN.1 Type:	Component
Derivation Path:	
Comments:	BICRoam

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
14,
                                                                  ss_ForBS

{
ss_Code      '9B'H
}
                                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_06
ASN.1 Type:	Component
Derivation Path:	
Comments:	BOIC

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                {
                                                                  invokeID
?,
                                                                  localValue
14,
                                                                  ss_ForBS

{
ss_Code      '93'H
}
                                                                 }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_07
ASN.1 Type:	Component
Derivation Path:	
Comments:	BAIC

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
14,
                                                                      ss_ForBS

{
ss_Code      '9A'H
}
                                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	InterrogateSS_08
ASN.1 Type:	Component
Derivation Path:	
Comments:	BOICExHC

Constraint Value

```
interrogateSSComponents     interrogateSS_InvokeComp
                                                                    {
                                                                      invokeID
?,
                                                                      localValue
14,
                                                                      ss_ForBS

{
ss_Code      '94'H
}
                                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	NotificationSS_07(Invkid: OCTETSTRING)
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
unstructuredSSNotifyComponents     unstructuredSSNotify_ReturnResultComp
                                                                    {
                                                                      invokeID
OC_OctToInvokeIDType(Invkid),
                                                                      result
                                                                              {
localValue                    61
                                                                              }
                                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	NotificationSS_09(Invkid: OCTETSTRING)
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
unstructuredSSNotifyComponents     unstructuredSSNotify_ReturnErrorComp

                                                        errorCodes
                                                                   {
                                                                      invokeID
OC_OctToInvokeIDType(Invkid),
                                                                      errorCode
ussd_Busy
                                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegisterSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFNRy Speech

Constraint Value

```
registerSSComponents     registerSS_InvokeComp   {
     invokeID                    ?,
     localValue                 10,
registerSS_Arg  {

ss_Code                         '2A'H,

basicService    teleservice     '10'H,

forwardedToNumber         '8100342143'H,
                                        -- Unknown Number/ISDN telephony + '00431234'

noReplyConditionTime       5

}
                                                  }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegisterSS_02
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFU

Constraint Value

```
registerSSComponents     registerSS_InvokeComp      {
   invokeID                     ?,
   localValue                  10,
   registerSS_Arg  {

ss_Code                          '21'H,

basicService    teleservice     '60'H,

forwardedToNumber         '8100342143'H
                                        -- Unknown Number/ISDN telephony + '00431234'

                }
                                                    }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegisterSS_03
ASN.1 Type:	Component
Derivation Path:	
Comments:	CFB

Constraint Value

```
registerSSComponents     registerSS_InvokeComp     {
       invokeID                    ?,
       localValue                 10,
       registerSS_Arg     {

ss_Code                      '29'H,

basicService    bearerService      '60'H,

forwardedToNumber          '8100342143'H
                                    -- Unknown Number/ISDN telephony + '00431234'
                            }
                                                  }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegisterSS_04
ASN.1 Type:	Component
Derivation Path:	
Comments:	CF for all facsimile

Constraint Value

```
registerSSComponents     registerSS_InvokeComp     {
     invokeID                     ?,
     localValue                  10,
     registerSS_Arg      {

ss_Code                           '20'H,

basicService     teleservice      '60'H,

forwardedToNumber           '8100342143'H
                                      -- Unknown Number/ISDN telephony + '00431234'
                           }
                                                   }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	RegPasswdSS_01
ASN.1 Type:	Component
Derivation Path:	
Comments:	All call restriction services.

Constraint Value

```
registerPasswordComponents     registerPassword_InvokeComp
                                                                                {
                                                                                  invokeID
?,
                                                                                  localValue
17,
                                                                                  ss_Code
'90'H
                                                                                   }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ProcessUSSData_01(ussdString: IA5String)
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
processUnstructuredSSDataComponents     processUnstructuredSSData_InvokeComp
                                                                          {
                                                                            invokeID
?,
                                                                            localValue
19,
                                                                            ss_UserData
ussdString
                                                                          }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	ProcessUSSDReq_01(ussdString: IA5String)
ASN.1 Type:	Component
Derivation Path:	
Comments:	

Constraint Value

```
processUnstructuredSSRequestComponents     processUnstructuredSSRequest_InvokeComp
                                                                          {
                                                                            invokeID
?,
                                                                            localValue
59,
                                                                            ussd_Arg

{
ussd_DataCodingScheme      'F0'O,
ussd_String            OC_CodingOfUssdString(ussdString)
}
                                                                           }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	USSDReq_04(Invkid :OCTETSTRING; ussdString: IA5String)
ASN.1 Type:	Component
Derivation Path:	
Comments:	ReturnResult for Unstructured SS request

Constraint Value

```
unstructuredSSRequestComponents     unstructuredSSRequest_ReturnResultComp
                                                                          {
                                                                            invokeID
OC_OctToInvokeIDType(Invkid),
                                                                            result
                                                                                     {
localValue                     60,
ussd_Res
{
ussd_DataCodingScheme      'F0'O,
ussd_String OC_CodingOfUssdString(ussdString)
}
                                                                                     }
                                                                           }
```

Detailed Comments:

ASN.1 Type Constraint Declaration	
Constraint Name:	USSDReq_05(Invkid :OCTETSTRING)
ASN.1 Type:	Component
Derivation Path:	
Comments:	Return Error for UnstructuredSS-Request with the error code USSD Busy
Constraint Value	
`unstructuredSSRequestComponents  unstructuredSSRequest_ReturnErrorComp errorCodes { invokeID OC_OctToInvokeIDType(Invkid), errorCode ussd_Busy }`	
Detailed Comments:	

ASP constraint declarations

TTCN ASP constraint declarations

ASP Constraint Declaration		
Constraint Name:	AlertRcv(pdu: ALERT_PDU)	
ASP Type:	DL_DatInAlert	
Derivation Path:		
Comments:	To receive an ALERTING message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AlertSnd(ch:LOGICCH; pdu: ALERT_PDU)	
ASP Type:	DL_DatRqAlert	
Derivation Path:		
Comments:	To send an ALERTING message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AssCmd(ch:LOGICCH; pdu: ASS_CMD_PDU)	
ASP Type:	DL_DatRqAssCmd	
Derivation Path:		
Comments:	To send an ASSIGNMENT COMMAND message which is assigned in send statement.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AssCmp(ch:LOGICCH; pdu:ASS_COM_PDU)	
ASP Type:	DL_DatInAssCom	
Derivation Path:		
Comments:	To match any received ASSIGNMENT COMPLETE message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AssFl_02(ch:LOGICCH)	
ASP Type:	DL_DatInAssfl	
Derivation Path:		
Comments:	protocol error undefined	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	AssgnFl_02	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AssFl_any_cau(ch: LOGICCH)	
ASP Type:	DL_DatInAssfl	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	AssgnFl_01	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AuthReq(ch: LOGICCH; pdu: AUTH_RQ_PDU)	
ASP Type:	DL_DatRqAuthRq	
Derivation Path:		
Comments:	To send an AUTHENTICATION REQUEST message with default values.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AuthRejSnd(ch: LOGICCH; pdu:AUTH_REJ_PDU)	
ASP Type:	DL_DatRqAuthRej	
Derivation Path:		
Comments:	To send an AUTHENTICATION REJECT message with default values.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	AuthRes(pdu: AUTH_RES_PDU)	
ASP Type:	DL_DatInAuthRes	
Derivation Path:		
Comments:	To match any received AUTHENTICATION RESPONSE message which contains any SRES.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CallCfm(pdu: CALL_CO_PDU)	
ASP Type:	DL_DatInCallCo	
Derivation Path:		
Comments:	To receive a CC CALL CONFIRMED message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CallProc(ch: LOGICCH; pdu: CALL_PROC_PDU)	
ASP Type:	DL_DatRqCallProc	
Derivation Path:		
Comments:	To send a CALL PROCEEDING message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChmmoAckRcv(ch:LOGICCH; msg:CHMMO_ACK_PDU)	
ASP Type:	DL_DatInChmmoAck	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChmmoReqSnd(ch:LOGICCH; msg:CHMMO_PDU)	
ASP Type:	DL_DatRqChmmo	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChRel(ch: LOGICCH; pdu: CH_REL_PDU)	
ASP Type:	DL_DatRqChRel	
Derivation Path:		
Comments:	To send a CHANNEL RELEASE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChRel_inv(ch:LOGICCH)	
ASP Type:	DL_DatRqChRelErr	
Derivation Path:		
Comments:	To send a CHANNEL RELEASE message containing additional IE unknown in the RR protocol	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	ChRelease_inv_03	
Detailed Comments:	used in TC_26_5_6_3	

ASP Constraint Declaration		
Constraint Name:	ChReq(pdu: CH_RQ_PDU)	
ASP Type:	DL_RacInChRq	
Derivation Path:		
Comments:	To receive a primitive containing a CHANNEL REQUEST message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
fn	?	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ChReq_01(ch:LOGICCH; pdu:CH_RQ_PDU)	
ASP Type:	DL_RacInChRq	
Derivation Path:		
Comments:	To receive a CHANNEL REQUEST message in cell B	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
fn	?	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CMSerAcp(ch:LOGICCH; pdu: CMS_ACP_PDU)	
ASP Type:	DL_DatRqCmsAcp	
Derivation Path:		
Comments:	To send a CM service accept message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CMSerRej(ch:LOGICCH; pdu: CMS_REJ_PDU)	
ASP Type:	DL_DatRqCmsRej	
Derivation Path:		
Comments:	reject cause = "service or option not available, unspecified"	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CMSerReq(pdu: CMS_RQ_PDU)	
ASP Type:	DL_EstInCmsRq	
Derivation Path:		
Comments:	To receive a CM SERVICE REQUEST message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CMSerDatReq(pdu:CMS_RQ_PDU)	
ASP Type:	DL_DatInCmsRq	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ConnAckRcv(pdu: CONN_ACK_PDU)	
ASP Type:	DL_DatInConnAck	
Derivation Path:		
Comments:	To receive a CONNECT ACKNOWLEDGE message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ConnAckSnd(ch:LOGICCH; pdu: CONN_ACK_PDU)	
ASP Type:	DL_DatRqConnAck	
Derivation Path:		
Comments:	To send a CONNECT ACKNOWLEDGE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	Conn_invSnd(ch:LOGICCH; pdu:CONN_PDU_ERR)	
ASP Type:	DL_DatRqConnErr	
Derivation Path:		
Comments:	To send a CC CONNECT message containing a mandatory IE coded as comprehension required.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ConnRcv(pdu: CONN_PDU)	
ASP Type:	DL_DatInConn	
Derivation Path:		
Comments:	To receive a CC CONNECT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ConnSnd(ch: LOGICCH; pdu: CONN_PDU)	
ASP Type:	DL_DatRqConn	
Derivation Path:		
Comments:	To send a CC CONNECT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CphCmd(ch: LOGICCH; pdu: CPHM_CMD_PDU)	
ASP Type:	DL_DatRqCphmCmd	
Derivation Path:		
Comments:	To send a CIPHERING MODE COMMAND message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	CphCom(pdu: CPHM_COM_PDU)	
ASP Type:	DL_DatInCphmCom	
Derivation Path:		
Comments:	To receive a CIPHERING MODE COMPLETE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	Disc_inv(ch: LOGICCH; pdu:DISC_PDU)	
ASP Type:	DL_DatRqDisc	
Derivation Path:		
Comments:	To send a DISCONNECT message containing the transaction ID not refer to the active call.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	Disc_inv_err(ch:LOGICCH; pdu:DISC_PDU_ERR)	
ASP Type:	DL_DatRqDiscErr	
Derivation Path:		
Comments:	To send a DISCONNECT message containing unknown IEI	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DiscRcv(ch:LOGICCH; pdu: DISC_PDU)	
ASP Type:	DL_DatInDisc	
Derivation Path:		
Comments:	To match a received DISCONNECT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DiscSnd(ch:LOGICCH; pdu: DISC_PDU)	
ASP Type:	DL_DatRqDisc	
Derivation Path:		
Comments:	To send a DISCONNECT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLEstCo(ch:LOGICCH)	
ASP Type:	DL_EstCo	
Derivation Path:		
Comments:	The ASP is used by the L2 to inform the L3 about the establishment of multiple frame link (L2 -> L3).	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap3	
logic_ch	ch	
establish_mode	C_Norm	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLEstRq(ch:LOGICCH)	
ASP Type:	DL_EstRq	
Derivation Path:		
Comments:	Request of a layer 2 connection establishment	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap3	
logic_ch	ch	
establish_mode	C_Norm	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLEstInd(ch :LOGICCH)	
ASP Type:	DL_EstIn	
Derivation Path:		
Comments:	Indication of a layer 2 connection establishment	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
establish_mode	?	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLRelInd_01	
ASP Type:	DL_RelIn	
Derivation Path:		
Comments:	Layer 2 indication of the layer 2 connection has been released.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
release_mode	?	
outstanding_indicator	?	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	DLRelInd_02	
ASP Type:	DL_RelIn	
Derivation Path:		
Comments:	Layer 2 indication of the layer 2 connection has been released.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
release_mode	C_LocEndRel	
outstanding_indicator	?	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ESetupRcv(pdu: ESETUP_PDU)	
ASP Type:	DL_DatInESetup	
Derivation Path:		
Comments:	To receive an emergency call SETUP message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvSnd(ch:LOGICCH; msg:HO_CMD_PDU)	
ASP Type:	DL_DatRqHoCmd	
Derivation Path:		
Comments:	To send a HANDOVER COMMAND indicating a finely synchronised intra cell handover.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvAcc_01	
ASP Type:	DL_RacInHoacc	
Derivation Path:		
Comments:	To received any HANDOVER ACCESS message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	HandOverAcc_01	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvAccRcv(ch: LOGICCH; msg:HOACC_PDU)	
ASP Type:	DL_RacInHoacc	
Derivation Path:		
Comments:	To received any HANDOVER ACCESS message on channel `ch`.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvFlRcv(ch:LOGICCH; msg :HOFL_PDU)	
ASP Type:	DL_DatInHofl	
Derivation Path:		
Comments:	To match a received HANDOVER FAILURE message containing any RR cause.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	HndOvCmpRcv(ch:LOGICCH; msg:HO_COM_PDU)	
ASP Type:	DL_DatInHoCom	
Derivation Path:		
Comments:	To receive any HANDOVER COMPLETE message on channel ch.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	msg	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	IDReq(ch:LOGICCH; pdu: ID_RQ_PDU)	
ASP Type:	DL_DatRqIdRq	
Derivation Path:		
Comments:	To send IDENTITY REQUEST message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	IDRes(pdu: ID_RES_PDU)	
ASP Type:	DL_DatInIdRes	
Derivation Path:		
Comments:	To receive a IDENTITY RESPONSE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ImmAss(ch: LOGICCH; pdu: IMMASS_PDU)	
ASP Type:	DL_UdatRqImmass	
Derivation Path:		
Comments:	To send an IMMEDIATE ASSIGNMENT message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ImmAssSp(ch:LOGICCH; pgg:PGG; msg:IMMASS_PDU)	
ASP Type:	DL_UdatRqImmass_sp	
Derivation Path:		
Comments:	To send an IMMEDIATE ASSIGNMENT message on the MS paging channel.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
pgg	pgg	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ImmAssX(ch:LOGICCH; pdu:IMMASSX_PDU)	
ASP Type:	DL_UdatRqImmassx	
Derivation Path:		
Comments:	To send an IMMEDIATE ASSIGNMENT EXTENDED message which assigns the SDCCH/4 channel for the MS	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:	The request reference (Rr, Fn) is used for the MS1, whilst (Rr_9, Fn_9) for MS2.	

ASP Constraint Declaration		
Constraint Name:	ImmAssXSp(ch:LOGICCH; pgg:PGG; msg:IMMASSX_PDU)	
ASP Type:	DL_UdatRqImmassx_sp	
Derivation Path:		
Comments:	To send an IMMEDIATE ASSIGNMENT EXTENDED message on the MS paging channel.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
pgg	pgg	
msg	msg	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ImmAssRej(ch:LOGICCH; pdu:IMMASS_REJ_PDU)	
ASP Type:	DL_UdatRqImmassRej	
Derivation Path:		
Comments:	To send an IMMEDIATE ASSIGNMENT REJECT message containing normal paging mode and wait indication = 0 seconds.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ImmAssRejSp(ch:LOGICCH; pgg:PGG; msg:IMMASS_REJ_PDU)	
ASP Type:	DL_UdatRqImmassRej_sp	
Derivation Path:		
Comments:	To send an IMMEDIATE ASSIGNMENT REJECT message on the MS paging channel.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
pgg	pgg	
msg	msg	
Detailed Comments:	Only used in TC_26_6_2_4	

ASP Constraint Declaration		
Constraint Name:	ImsiDet(pdu:IMSID_IN_PDU)	
ASP Type:	DL_EstInImsidIn	
Derivation Path:		
Comments:	To match a received IMSI DETACH INDICATION message matching any MS classmark1 value and any mobile identity value	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	LocAcp(ch:LOGICCH; pdu:LUP_ACP_PDU)	
ASP Type:	DL_DatRqLupAcp	
Derivation Path:		
Comments:	To send a LOCATION UPDATING ACCEPT message with a new mobile identity TMSI and location area.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	LocAcp_inv(ch: LOGICCH; mi1, mi2:MI; mcc, mnc, lac:OCTETSTRING)	
ASP Type:	DL_DatRqLupAcpErr	
Derivation Path:		
Comments:	To send a LOCATION UPDATING ACCEPT message with duplicated mobile identities.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	LocUpdtAcp_inv_01(mi1, mi2, mcc, mnc, lac)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	LocUp(ch: LOGICCH; pdu:LUP_RQ_PDU)	
ASP Type:	DL_EstInLupRq	
Derivation Path:		
Comments:	To match a received LOCATION UPDATING REQUEST message containing location updating type = IMSI attach.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	LocRej(ch: LOGICCH; pdu:LUP_REJ_PDU)	
ASP Type:	DL_DatRqLupRej	
Derivation Path:		
Comments:	To send a LOCATION UPDATING REJECT message containing the reject cause IMSI unknown in HLR. Used var's: ch	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	MDLRelReq(ch :LOGICCH)	
ASP Type:	MDL_RelRq	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
release_mode	C_LocEndRel	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	MsrRept(pdu:MSR_RPT_PDU)	
ASP Type:	DL_UdatInMsrRpt	
Derivation Path:		
Comments:	To receive a measurement report without measurement results.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PgReq1(ch:LOGICCH; pgg:PGG; pg1_req_pdu: PG1_RQ_PDU)	
ASP Type:	DL_UdatRqPg1Rq	
Derivation Path:		
Comments:	To send a PAGING REQUEST TYPE 1 message to the paging group indicated by the parameter pgg which is derived from system parameters. The PAGING REQUEST TYPE1 message requests normal paging mode, any channel and for the MS identity TSPX_TMSI.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
pgg	pgg	
msg	pg1_req_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PgReq2(ch:LOGICCH; pgg:PGG; pg2_req_pdu: PG2_RQ_PDU)	
ASP Type:	DL_UdatRqPg2Rq	
Derivation Path:		
Comments:	To send a PAGING REQUEST TYPE 1 message to the paging group indicated by the parameter pgg which is derived from system parameters. The PAGING REQUEST TYPE1 message requests normal paging mode, any channel and for the MS identity TSPX_TMSI.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
pgg	pgg	
msg	pg2_req_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PgReq3(ch:LOGICCH; pgg:PGG; pg3_req_pdu: PG3_RQ_PDU)	
ASP Type:	DL_UdatRqPg3Rq	
Derivation Path:		
Comments:	To send a PAGING REQUEST TYPE 1 message to the paging group indicated by the parameter pgg which is derived from system parameters. The PAGING REQUEST TYPE1 message requests normal paging mode, any channel and for the MS identity TSPX_TMSI.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
pgg	pgg	
msg	pg3_req_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PagingRes(pdu: PG_RES_PDU)	
ASP Type:	DL_EstInPgRes	
Derivation Path:		
Comments:	To receive a PAGING RESPONSE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PgRes(ch:LOGICCH; pdu:PG_RES_PDU)	
ASP Type:	DL_EstInPgRes	
Derivation Path:		
Comments:	To match any received PAGING RESPONSE message on the channel `ch`.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
establish_mode	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	PhyInfo_01(ch: LOGICCH; ta:TA)	
ASP Type:	DL_DatRqPhyinfo	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	Phyinfor_01(ta)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ReleaseRcv(pdu: REL_PDU)	
ASP Type:	DL_DatInRel	
Derivation Path:		
Comments:	To receive a CC RELEASE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	ReleaseSnd(ch: LOGICCH; pdu: REL_PDU)	
ASP Type:	DL_DatRqRel	
Derivation Path:		
Comments:	To send a RELEASE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	RelComRcv(pdu: REL_COM_PDU)	
ASP Type:	DL_DatInRelCmp	
Derivation Path:		
Comments:	To receive a RELEASE COMPLETE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	RelComSnd(ch:LOGICCH; pdu: REL_COM_PDU)	
ASP Type:	DL_DatRqRelCmp	
Derivation Path:		
Comments:	To send a RELEASE COMPLETE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SetupRcv(pdu: SETUP_MO_PDU)	
ASP Type:	DL_DatInSetup	
Derivation Path:		
Comments:	To receive a SETUP message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	?	
msg	pdu	
fn	?	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SetupSnd(ch: LOGICCH; setup:SETUP_MT_PDU)	
ASP Type:	DL_DatRqSetup	
Derivation Path:		
Comments:	To send a SETUP message which is the input parameter of this ASP.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	setup	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SyncInfo(ch:LOGICCH; bcc:BCC; ncc:NCC)	
ASP Type:	DL_UdatRqSchinfo	
Derivation Path:		
Comments:	To send SYNCHRONIZATION INFORMATION message with default parameters.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SyncInfor(bcc, ncc)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo1(ch:LOGICCH; cchd:CCHD; maxtx:B_2; txint:B_4; re:B_1)	
ASP Type:	DL_UdatRqSysinfo1	
Derivation Path:		
Comments:	To send SYSTEM INFORMATION TYPE 1 in cell A for RR testing of GSM900.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf1(cchd, maxtx, txint, re)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo1_cba(ch:LOGICCH; cchd:CCHD; maxtx:B_2; txint:B_4; re:B_1; cba:B_1; acc:B_5)	
ASP Type:	DL_UdatRqSysinfo1	
Derivation Path:		
Comments:	To send SYSTEM INFORMATION TYPE 1 in cell A for RR testing of GSM900.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf1_cba(cchd, maxtx, txint, re, cba, acc)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo1_nh(ch:LOGICCH; ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
ASP Type:	DL_UdatRqSysinfo1_nh	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf3(ci, mcc, mnc, lac, ccd, co, crh, mtmc, neci, maxtx, txint, re)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo2(ch: LOGICCH; bcchfl: NCD; maxtx:B_2; txint:B_4; re:B_1; nccp:NCCP)	
ASP Type:	DL_UdatRqSysinfo2	
Derivation Path:		
Comments:	To send the default SYSTEM INFORMATION TYPE 2 message containing default neighbour cells description in cell B for GSM900.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf2(bcchfl, maxtx, txint, re, nccp)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo2_cba(ch: LOGICCH; bcchfl: NCD; maxtx:B_2; txint:B_4; re:B_1; nccp:NCCP; cba:B_1; acc:B_5)	
ASP Type:	DL_UdatRqSysinfo2	
Derivation Path:		
Comments:	To send the default SYSTEM INFORMATION TYPE 2 message containing default neighbour cells description in cell B for GSM900.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf2_cba(bcchfl, maxtx, txint, re, nccp, cba, acc)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo2bis(ch: LOGICCH; bcchflegsm: NCD)	
ASP Type:	DL_UdatRqSysinfo2bis	
Derivation Path:		
Comments:	SYSTEM INFORMATION 2bis in cell A under EGSM with the ARFCN list = {988, 990, 1003}.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf2bis(bcchflegsm)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo2bis_egsm(ch: LOGICCH; freqlst:NCD)	
ASP Type:	DL_UdatRqSysinfo2bis	
Derivation Path:		
Comments:	SYSTEM INFORMATION 2bis under EGSM with the ARFCN list = freqlst.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf2bis_egsm(freqlst)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo2ter(ch: LOGICCH; fl: NCD2)	
ASP Type:	DL_UdatRqSysinfo2ter	
Derivation Path:		
Comments:	SYSTEM INFORMATION 2ter with ther ARFCN list = fl.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf2ter(fl)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo3(ch:LOGICCH; ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
ASP Type:	DL_UdatRqSysinfo3	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf3(ci, mcc, mnc, lac, ccd, co, crh, mtmc, neci, maxtx, txint, re)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo3_ram_cba(ch:LOGICCH; ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; cba, cbq:B_1; acc:B_5)	
ASP Type:	DL_UdatRqSysinfo3	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf3_ram_cba(ci, mcc, mnc, lac, ccd, co, crh, mtmc, neci, maxtx, txint, re, ram, cba, cbq, acc)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo3_rpct(ch:LOGICCH; ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; pt:B_5; cro:B_6; to:B_3)	
ASP Type:	DL_UdatRqSysinfo3	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf3_rpct(ci, mcc, mnc, lac, ccd, co, crh, mtmc, neci, maxtx, txint, re, ram, pt, cro, to)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo3_ti(ch:LOGICCH; ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; pt:B_5; cro:B_6; to:B_3)	
ASP Type:	DL_UdatRqSysinfo3	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf3_ti(ci, mcc, mnc, lac, ccd, co, crh, mtmc, neci, maxtx, txint, re, ram, pt, cro, to)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo3_rpctp(ch:LOGICCH; ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; pt:B_5; cro:B_6; to:B_3; po:B_2)	
ASP Type:	DL_UdatRqSysinfo3	
Derivation Path:		
Comments:		
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf3_rpctp(ci, mcc, mnc, lac, ccd, co, crh, mtmc, neci, maxtx, txint, re, ram, pt, cro, to, po)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo4(ch:LOGICCH; mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
ASP Type:	DL_UdatRqSysinfo4	
Derivation Path:		
Comments:	To send a default SYSTEM INFORMATION TYPE 4 message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf4(mcc, mnc, lac, crh, mtmc, neci, maxtx, txint, re)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo4_bis(ch:LOGICCH; mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6)	
ASP Type:	DL_UdatRqSysinfo4	
Derivation Path:		
Comments:	To send a default SYSTEM INFORMATION TYPE 4 message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf4_bis(mcc, mnc, lac, crh, mtmc, neci, maxtx, txint, re, ram)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo4_CBMS(ch:LOGICCH; mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; cbchd:CHD)	
ASP Type:	DL_UdatRqSysinfo4	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 4 message for SMSCB with the values or GSM 11.10, 34.3	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf4_CBMS(mcc, mnc, lac, crh, mtmc, neci, maxtx,txint, re, cbchd)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo4_ram_cba(ch:LOGICCH; mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; cba, cbq:B_1; acc:B_5)	
ASP Type:	DL_UdatRqSysinfo4	
Derivation Path:		
Comments:	To send a default SYSTEM INFORMATION TYPE 4 message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf4_ram_cba(mcc, mnc, lac, crh, mtmc, neci, maxtx, txint, re, ram, cba, cbq, acc)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo4_rpct(ch:LOGICCH; mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; pt:B_5; cro:B_6; to:B_3)	
ASP Type:	DL_UdatRqSysinfo4	
Derivation Path:		
Comments:	To send a default SYSTEM INFORMATION TYPE 4 message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf4_rpct(mcc, mnc, lac, crh, mtmc, neci, maxtx, txint, re, ram, pt, cro, to)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo4_rpctp(ch:LOGICCH; mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; pt:B_5; cro:B_6; to:B_3; po:B_2)	
ASP Type:	DL_UdatRqSysinfo4	
Derivation Path:		
Comments:	To send a default SYSTEM INFORMATION TYPE 4 message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf4_rpctp(mcc, mnc, lac, crh, mtmc, neci, maxtx, txint, re, ram, pt, cro, to, po)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo5(sacch: LOGICCH; sysinfo5_pdu:SYSINFO5_PDU)	
ASP Type:	DL_UdatRqSysinfo5	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 5 message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	sacch	
msg	sysinfo5_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo5bis(sacch: LOGICCH; sysinfo5bis_pdu:SYSINFO5bis_PDU)	
ASP Type:	DL_UdatRqSysinfo5bis	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 5bis message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	sacch	
msg	sysinfo5bis_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo5ter(sacch: LOGICCH; sysinfo5ter_pdu:SYSINFO5ter_PDU)	
ASP Type:	DL_UdatRqSysinfo5ter	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 5ter message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	sacch	
msg	sysinfo5ter_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo6(sacch: LOGICCH; sysinfo6_pdu:SYSINFO6_PDU)	
ASP Type:	DL_UdatRqSysinfo6	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 6 message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	sacch	
msg	sysinfo6_pdu	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo7(ch:LOGICCH; pt:B_5; cro:B_6; to:B_3)	
ASP Type:	DL_UdatRqSysinfo7	
Derivation Path:		
Comments:	To send a default SYSTEM INFORMATION TYPE 7 message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf7(pt, cro, to)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	SysInfo8(ch:LOGICCH; pt:B_5; cro:B_6; to:B_3)	
ASP Type:	DL_UdatRqSysinfo8	
Derivation Path:		
Comments:	To send a default SYSTEM INFORMATION TYPE 8 message	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	SysInf8(pt, cro, to)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	TmsiReallocSnd(par:MI; ch:LOGICCH; mcc, mnc, lac:OCTETSTRING)	
ASP Type:	DL_DatRqTmsireCmd	
Derivation Path:		
Comments:	To send a TMSI REALLOCATION COMMAND message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	TmsiReallocCmd(par, mcc, mnc, lac)	
Detailed Comments:		

ASP Constraint Declaration		
Constraint Name:	TmsiReallocCmp(ch:LOGICCH)	
ASP Type:	DL_DatInTmsireCom	
Derivation Path:		
Comments:	To receive a TMSI REALLOCATION COMPLETE message.	
Parameter Name	**Parameter Value**	**Comments**
sapi	C_Sap0	
logic_ch	ch	
msg	TmsiReallocComp	
fn	?	
Detailed Comments:		

PDU constraint declarations

TTCN PDU constraint declarations

PDU Constraint Declaration		
Constraint Name:	Alerting_01(Ti :TI)	
PDU Type:	ALERT_PDU	
Derivation Path:		
Comments:	An ALERTING message containing mandatory IE's only to be sent to the MS.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000001'B	
fie	-	
pi	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Alerting_04(Ti :TI; fie:FIE)	
PDU Type:	ALERT_PDU	
Derivation Path:		
Comments:	An ALERTING message containing facility IE to be sent to the MS.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000001'B	
fie	fie	
pi	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AlertingInd_01(ti:TI)	
PDU Type:	ALERT_PDU	
Derivation Path:		
Comments:	To match an received ALERTING message	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?000001'B	
fie	*	
pi	-	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AlertingInd_02(Ti:TI)	
PDU Type:	ALERT_PDU	
Derivation Path:		
Comments:	To match an received ALERTING message	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?000001'B	
fie	*	
pi	-	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_nfh(type:CH_TDMA; slot:SN; tsc:TSC; par_pwlvl, arfcn:INTEGER; cchd:CCHD; chmod:CHMOD; strt:STRT; cphms:CPHMS)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message. The channel mode and type are specified as parameters.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_nfh(type, slot, tsc, arfcn)	
pcmd	Pcmd_19(INT_TO_BIT(par_pwlvl, 5))	
frql_at	-	
cchd	cchd	
ch1mod	chmod	
ch2d_at	-	
ch2mod	-	
ma_at	-	
strt	strt	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	cphms	
Detailed Comments:	used iin CC testing	

PDU Constraint Declaration		
Constraint Name:	AsgnCmd_inv_01(slot:SN; tsc:TSC)	
PDU Type:	ASS_CMD_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMMAND message containing invalid skip identifier	
Field Name	**Field Value**	**Comments**
ski	'0100'B	
rrpd	'0110'B	
mt	'00101110'B	
ch1d_at	ChDescrp_nfh('00001'B, slot, tsc, C_arfcnA)	
pcmd	Pcmd_19('01001'B)	
frql_at	-	
cchd	-	
ch1mod	-	
ch2d_at	-	
ch2mod	-	
ma_at	-	
strt	-	
frql_bt	-	
ch1d_bt	-	
ch2d_bt	-	
frqchs_bt	-	
ma_bt	-	
cphms	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AsgnCmp_02	
PDU Type:	ASS_COM_PDU	
Derivation Path:		
Comments:	An ASSIGNMENT COMPLETE message containing any cause.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101001'B	
rrcau	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AssgnFl_01	
PDU Type:	ASSFL_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101111'B	
rrcau	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AssgnFl_02	
PDU Type:	ASSFL_PDU	
Derivation Path:	AssgnFl_01.	
Comments:	#6F	
Field Name	**Field Value**	**Comments**
rrcau	'01101111'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Abortmsg_01(par: REJCAU)	
PDU Type:	ABRT_PDU	
Derivation Path:		
Comments:	An ABORT message with reject cause which should be given as parameter.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00101001'B	
rejcau	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AuthRequest(cksn: B_3; rand:RAND)	
PDU Type:	AUTH_RQ_PDU	
Derivation Path:		
Comments:	An AUTHENTICATION REQUEST message containing default ciphering key sequence number and default challenge RAND from PIXIT.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00010010'B	
shoct	'0000'B	
cphksn	CphKeySN_07(cksn)	
rand	rand	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	AuthResponse	
PDU Type:	AUTH_RES_PDU	
Derivation Path:		
Comments:	An AUTHENTICATION RESPONSE message matching any SRES.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?010100'B	
sres	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallProced_01(Ti:TI)	
PDU Type:	CALL_PROC_PDU	
Derivation Path:		
Comments:	An CALL PROCEEDING message with mandatory IE's only.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000010'B	
bcri	-	
bcap1	-	
bcap2	-	
fie	-	
pi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CallProced_03	
PDU Type:	CALL_PROC_PDU	
Derivation Path:		
Comments:	An CALL PROCEEDING message templet used for CallProcGen	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'00000010'B	
bcri	-	
bcap1	-	
bcap2	-	
fie	-	
pi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CCStatus_01(Ti:TI)	
PDU Type:	CCST_PDU	
Derivation Path:		
Comments:	A CC STATUS message to match any received CC STATUS.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?111101'B	
cau	?	
cst	?	
acst	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CCStatus_inv_01(ti:TI)	
PDU Type:	CCST_PDU	
Derivation Path:		
Comments:	A CC STATUS message without mandatory cause IE and call state IE.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00111101'B	
cau	-	
cst	-	
acst	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CCStatusEq_01(Ti:TI)	
PDU Type:	CCST_ENQ_PDU	
Derivation Path:		
Comments:	A STATUS ENQUIRY message	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00110100'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChmomoAck_01(chmd:CHMOD; chd: CHD)	
PDU Type:	CHMMO_ACK_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010111'B	
chd	chd	
chmod	chmd	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChmomoAck_02(chmd:CHMOD; chd:CHD)	
PDU Type:	CHMMO_ACK_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010111'B	
chd	chd	
chmod	chmd	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChmomoReq_01(chmd:CHMOD; chd:CHD)	
PDU Type:	CHMMO_PDU	
Derivation Path:		
Comments:	the channel being modified is default full rate traffic channel.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010000'B	
chd	chd	
chmod	chmd	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChmomoReq_02(chmd:CHMOD; chd:CHD)	
PDU Type:	CHMMO_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010000'B	
chd	chd	
chmod	chmd	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRelease_01	
PDU Type:	CH_REL_PDU	
Derivation Path:		
Comments:	A CHANNEL RELEASE message with RR cause = normal event	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00001101'B	
rrcau	'00000000'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRelease_inv_01	
PDU Type:	CH_REL_PDU	
Derivation Path:	ChRelease_01.	
Comments:	An invalid CHANNEL RELEASE message without mandatory IE RR cause	
Field Name	**Field Value**	**Comments**
rrcau	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRelease_inv_02	
PDU Type:	CH_REL_PDU	
Derivation Path:	ChRelease_01.	
Comments:	A CHANNEL RELEASE message containing incorrect skip indicator 6.	
Field Name	**Field Value**	**Comments**
ski	'0110'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRelease_inv_03	
PDU Type:	CH_REL_PDU_ERR	
Derivation Path:		
Comments:	A CHANNEL RELEASE message containing additional IE unknown in the RR protocol	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00001101'B	
rrcau	'00000000'B	
add	'6205AA55EF6701'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_01	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	A CHANNEL REQUEST message containing establishment cause = answer to paging.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'100????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_02	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match any received CHANNEL REQUEST message	
Field Name	**Field Value**	**Comments**
ecau_rrf	'????????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_03	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message containing establishment cause = '0001'B, "other procedures which can be completed with an SDCCH".	
Field Name	**Field Value**	**Comments**
ecau_rrf	'0001????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_04	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message which originates a call (establishment cause = '111'B), "Originating call and TCH/F is needed, or IMSI detach, SMS or SS procedures that can be completed with an SDCCH and NECI set to 0".	
Field Name	**Field Value**	**Comments**
ecau_rrf	'111?????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_05	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message containing establishment cause = '0100'B, "Originating speech call from dual-rate mobile station when TCH/H is sufficient and the network sets NECI bit to 1"	
Field Name	**Field Value**	**Comments**
ecau_rrf	'0100????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_06	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message which originates a half rate data call (establishment cause = '0101'B).	
Field Name	**Field Value**	**Comments**
ecau_rrf	'0101????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_07	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '0010'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'0010????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_08	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '0011'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'0011????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_09	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '000'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'000?????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_10	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	A CHANNEL REQUEST message with establishment cause = '110'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'110?????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_11	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '011010'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'011010??'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_12	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '100'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'100?????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_13	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '100'B or '0010'B or '0001'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	('100?????'B, '0010????'B, '0001????'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_14	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '100'B or '0011'B or '0001'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	('100?????'B, '0011????'B, '0001????'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_15	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '111'B or '0100'B or '0101'B --- initiate outgoing call or SDCCH other procedures.	
Field Name	**Field Value**	**Comments**
ecau_rrf	('111?????'B, '0100????'B, '0101????'B, '101?????'B, '0001????'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_16	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '101'B for emergency call.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'101?????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_17	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	A CHANNEL REQUEST message containing establishment cause = 100, 0010, 0011, 0001	
Field Name	**Field Value**	**Comments**
ecau_rrf	('100?????'B, '0010????'B, '0011????'B, '0001????'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_18	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '0000'B.	
Field Name	**Field Value**	**Comments**
ecau_rrf	'0000????'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ChRequest_19	
PDU Type:	CH_RQ_PDU	
Derivation Path:		
Comments:	To match the received CHANNEL REQUEST message with establishment cause = '111'B or '101'B --- initiate outgoing call	
Field Name	**Field Value**	**Comments**
ecau_rrf	('111?????'B, '101?????'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ClassMarkEnq_01	
PDU Type:	CLM_ENQ_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010011'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceAcp_01	
PDU Type:	CMS_ACP_PDU	
Derivation Path:		
Comments:	a CM SERVICE ACCEPT message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00100001'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceRej_01	
PDU Type:	CMS_REJ_PDU	
Derivation Path:		
Comments:	reject cause = 'service or option not available, unspecified"	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00100010'B	
mmcau	'20'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceRej_02	
PDU Type:	CMS_REJ_PDU	
Derivation Path:		
Comments:	reject cause = "IMEI not accepted"	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00100010'B	
mmcau	'05'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceRej_03	
PDU Type:	CMS_REJ_PDU	
Derivation Path:		
Comments:	reject cause = "Service Option not supported"	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00100010'B	
mmcau	'20'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceRej_04	
PDU Type:	CMS_REJ_PDU	
Derivation Path:		
Comments:	reject cause = 'network failure"	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00100010'B	
mmcau	'11'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceRej_30(par: REJCAU)	
PDU Type:	CMS_REJ_PDU	
Derivation Path:		
Comments:	reject cause = 'service or option not available, unsepcified"	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00100010'B	
mmcau	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_01	
PDU Type:	CMS_RQ_PDU	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?100100'B	
cphksn	?	
svtype	?	
msclm	?	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_04	
PDU Type:	CMS_RQ_PDU	
Derivation Path:	CMServiceReq_01.	
Comments:	To match a received CM SERVICE REQUEST message containing CM service type = "Mobile originating call establishment or packet mode connection establishment" or " emergency call establishment".	
Field Name	**Field Value**	**Comments**
svtype	('0001'B, '0010'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_05(mi:MI)	
PDU Type:	CMS_RQ_PDU	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message for emergency call with IMEI.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?100100'B	
cphksn	?	
svtype	C_CMServiceTypeE	
msclm	?	
mi	mi	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_08	
PDU Type:	CMS_RQ_PDU	
Derivation Path:	CMServiceReq_01.	
Comments:	To match the received CM SERVICE REQUEST message indicating " supplementary service activation"	
Field Name	**Field Value**	**Comments**
svtype	'1000'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_09	
PDU Type:	CMS_RQ_PDU	
Derivation Path:	CMServiceReq_01.	
Comments:	To match the received CM SERVICE REQUEST message indicating " short message transfer"	
Field Name	**Field Value**	**Comments**
svtype	'0100'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_30(parexpected_mi: MI)	
PDU Type:	CMS_RQ_PDU	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?100100'B	
cphksn	?	
svtype	?	
msclm	?	
mi	parexpected_mi	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_31(parexpected_mi: MI; cksn: BITSTRING)	
PDU Type:	CMS_RQ_PDU	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?100100'B	
cphksn	CphKeySN_07(cksn)	
svtype	C_CMServiceTypeE	
msclm	?	
mi	parexpected_mi	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CMServiceReq_32(parexpected_mi: MI; cksn: BITSTRING)	
PDU Type:	CMS_RQ_PDU	
Derivation Path:		
Comments:	To match any received CM SERVICE REQUEST message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?100100'B	
cphksn	CphKeySN_07(cksn)	
svtype	?	
msclm	?	
mi	parexpected_mi	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Connect_01(ti:TI)	
PDU Type:	CONN_PDU	
Derivation Path:		
Comments:	CC CONNECT message matching any received value	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'0?000111'B	
fie	*	
pi	-	
cnn	-	
cns	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Connect_02(Ti:TI)	
PDU Type:	CONN_PDU	
Derivation Path:		
Comments:	CC CONNECT message containing mandatory IE's only.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000111'B	
fie	-	
pi	-	
cnn	-	
cns	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Connect_05(Ti:TI; fie:FIE)	
PDU Type:	CONN_PDU	
Derivation Path:		
Comments:	CC CONNECT message containing faciliy IE.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00000111'B	
fie	fie	
pi	-	
cnn	-	
cns	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ConnectAck_01(ti:TI)	
PDU Type:	CONN_ACK_PDU	
Derivation Path:		
Comments:	A CONNECT ACKNOWLEDGE message n -> ms	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00001111'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ConnectAck_02(Ti:TI)	
PDU Type:	CONN_ACK_PDU	
Derivation Path:		
Comments:	A CONNECT ACKNOWLEDGE message	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?001111'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CphModeCmp_01	
PDU Type:	CPHM_COM_PDU	
Derivation Path:		
Comments:	To match any received CIPHERING MODE COMPLETE message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00110010'B	
mei	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	CphModeCmp_02	
PDU Type:	CPHM_COM_PDU	
Derivation Path:	CphModeCmp_01.	
Comments:	To match a received CIPHERING MODE COMPLETE message without IMEI.	
Field Name	**Field Value**	**Comments**
mei	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	DisconnS(Ti:TI; cau:CAU; progind:PI; uuinf:UU)	
PDU Type:	DISC_PDU	
Derivation Path:		
Comments:	A DISCONNECT message that contains cause = normal clearing and progress indicator = #8.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00100101'B	
cau	cau	
fie	-	
pi	progind	
uu	uuinf	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	DisconnR(Ti:TI; cau:CAU)	
PDU Type:	DISC_PDU	
Derivation Path:		
Comments:	A default DISCONNECT message matching parametrised value ms -> n.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?100101'B	
cau	cau	
fie	*	
pi	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Disconn_inv_02(ti:TI)	
PDU Type:	DISC_PDU	
Derivation Path:		
Comments:	A DISCONNECT message in which the mandatory IE cause is missing. (N -> MS)	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00100101'B	
cau	-	
fie	-	
pi	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FacilityPdu_05(Ti:TI)	
PDU Type:	FAC_PDU	
Derivation Path:		
Comments:	FACILITY message, call dependent, used in TC_11_3, containing any facility iformation element ms->n.	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'0?111010'B	
fie	?	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FacilityPdu_25(Ti:TI; fie:FIE)	
PDU Type:	FAC_PDU	
Derivation Path:		
Comments:	FACILITY message, call dependent, n -> ms	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'00111010'B	
fie	fie	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FacilityPdu_25_ci(Ti:TI; fie:FIE)	
PDU Type:	FAC_PDU	
Derivation Path:		
Comments:	FACILITY message, call independent, n -> ms	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'1011'B	
mt	'00111010'B	
fie	fie	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FacilityPdu_26(Ti:TI; fie:FIE)	
PDU Type:	FAC_PDU	
Derivation Path:		
Comments:	FACILITY message, call dependent ms -> n	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'0?111010'B	
fie	fie	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FacilityPdu_26_ci(Ti:TI; fie:FIE)	
PDU Type:	FAC_PDU	
Derivation Path:		
Comments:	FACILITY message, call independent ms -> n	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'1011'B	
mt	'0?111010'B	
fie	fie	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	FreqRedef_01(chd:CHD; ma:MA; strt:STRT; cchd:CCHD)	
PDU Type:	FRQRE_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010100'B	
chd	chd	
ma	ma	
strt	strt	
cchd	cchd	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverAcc_01	
PDU Type:	HOACC_PDU	
Derivation Path:		
Comments:	To match any received HANDOVER ACCESS message	
Field Name	**Field Value**	**Comments**
horf	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverAcc_02(horef:HORF)	
PDU Type:	HOACC_PDU	
Derivation Path:		
Comments:	To match a received HANDOVER ACCESS message with handover reference ` horef`.	
Field Name	**Field Value**	**Comments**
horf	horef	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverCmp_01	
PDU Type:	HO_COM_PDU	
Derivation Path:		
Comments:	To match any received HANDOVER COMPLETE message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101100'B	
rrcau	?	
motdif	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverCmp_03(ta_a, td_b:INTEGER)	
PDU Type:	HO_COM_PDU	
Derivation Path:		
Comments:	To match a received HANDOVER COMPLETE message with mobile time difference =((2*td_b + ta_a)) mod 2 097 152) +- 2	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101100'B	
rrcau	?	
motdif	Mtdif_01(ta_a, td_b)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOverCmp_20	
PDU Type:	HO_COM_PDU	
Derivation Path:		
Comments:	A basic received constraint for HANDOVER COMPLETE message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101100'B	
rrcau	'00000000'B	
motdif	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOvFail_01	
PDU Type:	HOFL_PDU	
Derivation Path:		
Comments:	A HANDOVER FAILURE message matching any RR cause.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101000'B	
rrcau	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HandOvFail_02	
PDU Type:	HOFL_PDU	
Derivation Path:		
Comments:	A HANDOVER FAILURE message matching any abnormal release RR cause.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101000'B	
rrcau	('00000001'B, '00000010'B, '00000011'B, '00000100'B, '01101111'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Holdpdu_01(Ti:TI)	
PDU Type:	HOLD_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?011000'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	HoldAckpdu_01(Ti:TI)	
PDU Type:	HOLD_ACK_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00011001'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDResponse_01	
PDU Type:	ID_RES_PDU	
Derivation Path:		
Comments:	An IDENTITY RESPONSE message which matches any mobile identity	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?011001'B	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDResponse_02	
PDU Type:	ID_RES_PDU	
Derivation Path:	IDResponse_01.	
Comments:	An IDENTITY RESPONSE message which matches TMSI of the MS under test	
Field Name	**Field Value**	**Comments**
mi	MiTmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDResponse_30(par:MI)	
PDU Type:	ID_RES_PDU	
Derivation Path:		
Comments:	An IDENTITY RESPONSE message which matches the given MI of the MS under test.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?011001'B	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDRequest_01(type:B_4)	
PDU Type:	ID_RQ_PDU	
Derivation Path:		
Comments:	An IDENTITY REQUEST message with specified identity type.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00011000'B	
shoct	'0000'B	
idtype	type	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDRequest_inv_01(skip :INTEGER)	
PDU Type:	ID_RQ_PDU	
Derivation Path:		
Comments:	An invalid IDENTITY REQUEST message with incorrect skip indicator.	
Field Name	**Field Value**	**Comments**
ski	INT_TO_BIT(skip, 4)	
mmpd	'0101'B	
mt	'00011000'B	
shoct	'0000'B	
idtype	'0001'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDRequest_inv_02	
PDU Type:	ID_RQ_PDU	
Derivation Path:		
Comments:	An invalid IDENTITY REQUEST message of which the identity type is coded as reserved value	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00011000'B	
shoct	'0000'B	
idtype	'1111'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	IDRequest_inv_03	
PDU Type:	ID_RQ_PDU	
Derivation Path:		
Comments:	An invalid IDENTITY REQUEST message containing arbitrary spare bits	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00011000'B	
shoct	'1010'B	
idtype	'1100'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_01Def(Rr: BITSTRING; Fn: FN; sub:B_2; slot:SN; tsc:TSC; par_arfcn:INTEGER; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/4 channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(sub)), 5), slot, tsc, par_arfcn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_nfh(Rr: BITSTRING; Fn: FN; type: CH_TDMA; slot:SN; tsc:TSC; ta:TA; arfcn: INTEGER; pgm:B_2)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign TCH/F or TCH/H channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(pgm)	
chd	ChDescrp_nfh(type, slot, tsc, arfcn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_inv_01(Rr :BITSTRING; Fn:FN; slot:SN; tsc:TSC; arfcn:INTEGER; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An invalid IMMEDIATE ASSIGNMENT message with unknown skip indicator ('0001'B)	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0001'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubDef)), 5), slot, tsc, arfcn)	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r(chd:CHD; Rr:BITSTRING; Fn:FN; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	chd	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r13(Rr: BITSTRING; Fn: FN; chd:CHD; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign TCH/F channel with a time advance of 30 bit periods.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	chd	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgn_r14(Rr: BITSTRING; Fn: FN; chd:CHD; ta:TA)	
PDU Type:	IMMASS_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign TCH/H channel with a time advance of 30 bit periods.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111111'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd	chd	
rqr	Rqr2(Rr, Fn)	
ta	ta	
ma	MoblAllc_01	
strt	-	
iaroct	IaRestOct_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnX_r01(Rr: BITSTRING; Fn: FN; Rr_9: BITSTRING; Fn_9: FN; slot:SN; tsc:TSC; ta:TA; arfcn:INTEGER)	
PDU Type:	IMMASSX_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/4 channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111001'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
chd1	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubA)), 5), slot, tsc, arfcn)	
rqr1	Rqr2(Rr, Fn)	
ta1	ta	
chd2	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(TSPX_SDCCH4SubB)), 5), slot, tsc, arfcn)	
rqr2	Rqr2(Rr_9, Fn_9)	
ta2	ta	
ma	MoblAllc_01	
strt	-	
iaxroct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnX_r03(Rr: BITSTRING; Fn: FN; sub1, sub2:B_2; slot:SN; tsc:TSC; ta:TA; arfcn:INTEGER)	
PDU Type:	IMMASSX_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT message to assign SDCCH/4 channel.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111001'B	
shoct	'0000'B	
pm	Pm(C_extended_paging)	
chd1	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(sub1)), 5), slot, tsc, arfcn)	
rqr1	Rqr2(Rr, Fn)	
ta1	ta	
chd2	ChDescrp_nfh(INT_TO_BIT((4 + BIT_TO_INT(sub2)), 5), slot, tsc, arfcn)	
rqr2	Rqr2(Rr, Fn)	
ta2	ta	
ma	MoblAllc_01	
strt	-	
iaxroct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_01(Rr: BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT REJECT message containing normal paging mode.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111010'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
rqr1	Rqr2(Rr, Fn)	
wi1	'00'H	
rqr2	Rqr2(Rr, Fn)	
wi2	'00'H	
rqr3	Rqr2(Rr, Fn)	
wi3	'00'H	
rqr4	Rqr2(Rr, Fn)	
wi4	'00'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_02(Rr:BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT REJECT message in which only the third request reference addresses the MS under test.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111010'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
rqr1	Rqr1(Rr, Fn)	
wi1	'02'H	
rqr2	Rqr1(Rr, Fn)	
wi2	'02'H	
rqr3	Rqr2(Rr, Fn)	
wi3	'00'H	
rqr4	Rqr1(Rr, Fn)	
wi4	'02'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_03(Rr: BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:	ImmAsgnRej_01.	
Comments:	An IMMEDIATE ASSIGNMENT REJECT message containing normal paging mode and wait indication = 5 seconds.	
Field Name	**Field Value**	**Comments**
rqr1	Rqr2(Rr, Fn)	
wi1	'05'H	
rqr2	Rqr2(Rr, Fn)	
wi2	'05'H	
rqr3	Rqr2(Rr, Fn)	
wi3	'05'H	
rqr4	Rqr2(Rr, Fn)	
wi4	'05'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_04(Rr: BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:	ImmAsgnRej_01.	
Comments:	An IMMEDIATE ASSIGNMENT REJECT message containing normal paging mode and wait indication = 6 seconds.	
Field Name	**Field Value**	**Comments**
rqr1	Rqr2(Rr, Fn)	
wi1	'06'H	
rqr2	Rqr2(Rr, Fn)	
wi2	'06'H	
rqr3	Rqr2(Rr, Fn)	
wi3	'06'H	
rqr4	Rqr2(Rr, Fn)	
wi4	'06'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_inv_01(Rr: BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:	ImmAsgnRej_01.	
Comments:	An invalid IMMEDIATE ASSIGNMENT REJECT message with skip indicator = 2, reject time = 255 s.	
Field Name	**Field Value**	**Comments**
ski	'0010'B	
rqr1	Rqr2(Rr, Fn)	
wi1	'FF'H	
rqr2	Rqr2(Rr, Fn)	
wi2	'FF'H	
rqr3	Rqr2(Rr, Fn)	
wi3	'FF'H	
rqr4	Rqr2(Rr, Fn)	
wi4	'FF'H	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_inv_02(Rr: BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:	ImmAsgnRej_01.	
Comments:	An invalid IMMEDIATE ASSIGNMENT REJECT message containing arbitrary spare bits	
Field Name	**Field Value**	**Comments**
shoct	'1010'B	
pm	Pm(C_normal_paging)	
rqr1	Rqr2(Rr, Fn)	
rqr2	Rqr2(Rr, Fn)	
wi2	TSPX_T3122	
rqr3	Rqr2(Rr, Fn)	
wi3	TSPX_T3122	
rqr4	Rqr2(Rr, Fn)	
wi4	TSPX_T3122	
iarroct	'010101'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_r01(rqr1, rqr2, rqr3, rqr4: RQR; t1, t2: INTEGER)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT REJECT message in which only the second and the third request reference addresses the MS under test.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111010'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
rqr1	rqr1	
wi1	'00'H	
rqr2	rqr2	
wi2	INT_TO_HEX(t1, 2)	
rqr3	rqr3	
wi3	INT_TO_HEX(t2, 2)	
rqr4	rqr4	
wi4	'00'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_r02(Rr: BITSTRING; Fn: FN)	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT REJECT message containing normal paging mode.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111010'B	
shoct	'0000'B	
pm	Pm(C_normal_paging)	
rqr1	Rqr2(Rr, Fn)	
wi1	'00'H	
rqr2	Rqr1(Rr, Fn)	
wi2	'00'H	
rqr3	Rqr1(Rr, Fn)	
wi3	'00'H	
rqr4	Rqr1(Rr, Fn)	
wi4	'00'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImmAsgnRej_r04	
PDU Type:	IMMASS_REJ_PDU	
Derivation Path:		
Comments:	An IMMEDIATE ASSIGNMENT REJECT message containing paging mode = "extended paging"and wait indication = 0 seconds. The Request References do not pertain to MS under test.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00111010'B	
shoct	'0000'B	
pm	Pm(C_extended_paging)	
rqr1	Rqr3	
wi1	'00'H	
rqr2	Rqr3	
wi2	'00'H	
rqr3	Rqr3	
wi3	'00'H	
rqr4	Rqr3	
wi4	'00'H	
iarroct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImsiDetach_01	
PDU Type:	IMSID_IN_PDU	
Derivation Path:		
Comments:	IMSI DETACH INDICATION message matching any MS classmark1 value and any mobile identity value	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?000001'B	
msclm	?	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ImsiDetach_30(par:MI)	
PDU Type:	IMSID_IN_PDU	
Derivation Path:		
Comments:	IMSI DETACH INDICATION message matching any MS classmark1 value and given IMSI.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?000001'B	
msclm	?	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtAcp(mcc, mnc, lac:OCTETSTRING)	
PDU Type:	LUP_ACP_PDU	
Derivation Path:		
Comments:	A LOCATION UPDATING ACCEPT message without mobile identity and LAI of PLMN2. LAC set in TCV_lac.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00000010'B	
lai	LocAreaId(mcc, mnc, lac)	
mi	-	
fop	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtAcp_01(newmi: MI; mcc, mnc, lac:OCTETSTRING)	
PDU Type:	LUP_ACP_PDU	
Derivation Path:		
Comments:	To assign a new TMSI with or without mobile identity.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00000010'B	
lai	LocAreaId(mcc, mnc, lac)	
mi	newmi	
fop	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtAcp_02(mcc, mnc, lac:OCTETSTRING)	
PDU Type:	LUP_ACP_PDU	
Derivation Path:		
Comments:	To assign a new TMSI.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00000010'B	
lai	LocAreaId(mcc, mnc, lac)	
mi	-	
fop	'10100001'B	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtAcp_inv_01(mi1, mi2:MI; mcc, mnc, lac:OCTETSTRING)	
PDU Type:	LUP_ACP_PDU_ERR	
Derivation Path:		
Comments:	Invalid LOCATION UPDATING ACCEPT message containing duplicated mobile identifier.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00000010'B	
lai	LocAreaId(mcc, mnc, lac)	
mi	mi1	
dupmi	mi2	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtReq_01(locup:B_2)	
PDU Type:	LUP_RQ_PDU	
Derivation Path:		
Comments:	A LOCATION UPDATING REQUEST message containing location updating type.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?001000'B	
cphksn	?	
lutype	LocUpType(locup)	
lai	?	
msclm	?	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtReq_04(locup:B_2; par:MI)	
PDU Type:	LUP_RQ_PDU	
Derivation Path:		
Comments:	LOCATION UPDATING REQUEST message containing TMSI.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?001000'B	
cphksn	?	
lutype	LocUpType(locup)	
lai	?	
msclm	?	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtReq_05	
PDU Type:	LUP_RQ_PDU	
Derivation Path:		
Comments:	to match any LOCATION UPDATING REQUEST message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?001000'B	
cphksn	?	
lutype	?	
lai	?	
msclm	?	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtReq_31(par:MI; mcc,mnc,lac:OCTETSTRING; locup:B_2; cksn: BITSTRING)	
PDU Type:	LUP_RQ_PDU	
Derivation Path:		
Comments:	A paramterised LOCATION UPDATING REQUEST message	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?001000'B	
cphksn	CphKeySN_07(cksn)	
lutype	LocUpType(locup)	
lai	LocAreaId(mcc, mnc, lac)	
msclm	ClassMark1	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	LocUpdtRej_01(par:REJCAU)	
PDU Type:	LUP_REJ_PDU	
Derivation Path:		
Comments:	A LOCATION UPDATING REJECT message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00000100'B	
rejcau	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_01	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report without measurement results	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_02	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report match any received MSR_RPT_PDU.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_02	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_03	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing results for 6 strongest carriers.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_03	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_03e(par_measres: MSRR)	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing results for 6 strongest carriers.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	par_measres	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_04	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing 4 strongest carriers.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_04	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_04e	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing 4 strongest carriers.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_04e	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_05	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing 6 strongest carriers and DTX was used.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_05	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_06	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing 6 strongest carriers and DTX is not checked.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_06	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	MsrReport_07	
PDU Type:	MSR_RPT_PDU	
Derivation Path:		
Comments:	A measurement report containing 2 strongest carriers and DTX is not used.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00010101'B	
msrr	MsrResult_07	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PagingRes_01	
PDU Type:	PG_RES_PDU	
Derivation Path:		
Comments:	To match any received PAGING RESPONSE message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00100111'B	
shoct	'0000'B	
cphksn	?	
msclm	?	
mi	?	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PagingRes_r01	
PDU Type:	PG_RES_PDU	
Derivation Path:	PagingRes_01.	
Comments:	To match any received PAGING RESPONSE message, RR tests.	
Field Name	**Field Value**	**Comments**
mi	MiTmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PagingRes_r02	
PDU Type:	PG_RES_PDU	
Derivation Path:	PagingRes_01.	
Comments:	To match the received PAGING RESPONSE message containing IMSI of the IUT for RR tests.	
Field Name	**Field Value**	**Comments**
mi	MiImsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PagingRes_30(par:MI; cksn: BITSTRING)	
PDU Type:	PG_RES_PDU	
Derivation Path:		
Comments:	To match any received PAGING RESPONSE message. Used in MM test cases.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00100111'B	
shoct	'0000'B	
cphksn	CphKeySN_07(cksn)	
msclm	?	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_01	
PDU Type:	PG1_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE1 message requesting any channel with normal paging mode, the mobile identity is TMSI.	
Field Name	**Field Value**	**Comments**
l2_pl	'25'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100001'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	MiTmsi_01	
mi2	-	
p1roct	'2B2B2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_05	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	PAGING REQUEST TYPE1 message with mobile identity 1 being IMSI of the MS.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((3 + OC_LengthOf(MiImsi_01)) * 4) + 1, 1)	
mi1	MiImsi_01	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O, 19 - OC_LengthOf(MiImsi_01))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_06	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	PAGING REQUEST TYPE1 message with mobile identity 1 being IMSI of the MS.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((3 + OC_LengthOf(MiImsi_r01)) * 4) + 1, 1)	
pm	Pm(C_extended_paging)	
mi1	MiImsi_r01	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O, 19 - OC_LengthOf(MiImsi_r01))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_07	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	PAGING REQUEST TYPE1 message with mobile identity 1 being IMSI of the MS.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((3 + OC_LengthOf(MiImsi_01)) * 4) + 1, 1)	
pm	Pm(C_reorg_paging)	
mi1	MiImsi_01	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O, 19 - OC_LengthOf(MiImsi_01))	
Detailed Comments:	Only used in TC_26_6_2_4 within Pg_Req1_08	

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_30(par_usedmi: MI)	
PDU Type:	PG1_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE1 message requesting any channel with normal paging mode, the mobile identity is paramerised. It is used only in MM test cases.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((3 + OC_LengthOf(par_usedmi)) * 4) + 1, 1)	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100001'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	par_usedmi	
mi2	-	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O, 19 - OC_LengthOf(par_usedmi))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1Norm	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	PAGING REQUEST TYPE1 message containing normal paging mode, used as paging filling message.	
Field Name	**Field Value**	**Comments**
l2_pl	'15'O	
mi1	MiImsi_02	
p1roct	'2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1Reorg	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	PAGING REQUEST TYPE1 message containing paging reorganisation mode, used as paging filling message.	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((3 + OC_LengthOf(MiImsi_02)) * 4) + 1, 1)	
pm	Pm(C_reorg_paging)	
mi1	MiImsi_02	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B'O, 19 - OC_LengthOf(MiImsi_02))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_inv_01(skip:INTEGER)	
PDU Type:	PG1_RQ_PDU	
Derivation Path:		
Comments:	An invalid PAGING REQUEST TYPE 1 message with incorrect skip indicator.	
Field Name	**Field Value**	**Comments**
l2_pl	'25'O	
ski	INT_TO_BIT(skip, 4)	
rrpd	'0110'B	
mt	'00100001'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	MiTmsi_01	
mi2	-	
p1roct	'2B2B2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_inv_04	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	An invalid PAGING REQUEST TYPE1 message containing rest octets which are not all '2B'O	
Field Name	**Field Value**	**Comments**
p1roct	'09021234784BFFFF2B2B2B2B2B'O	
Detailed Comments:	'agressive' rest octet coding for 26.5.7.1.1. The first byte is coded in a manner that indicates comprehension required to a mobile that is incorrectly using the L2 pseudo length. The length of this IE is the octet after '09' and is set to 2. After the two bytes of content is the "next IEI". This is 78 which was the IEI allocated (but never used) in phase 1 to the RACH control parameters. The "RACH Control Parameters" are "set" to indicate that the cell barred and all access classes are prohibited.	

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_r01	
PDU Type:	PG1_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE1 message requesting any channel with normal paging mode, the first mobile identity is TMSI, the second is IMSI different from TSPX_IMSI.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100001'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	MiTmsi_01	
mi2	MiImsi_r01iei	
p1roct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_r02	
PDU Type:	PG1_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE1 message requesting any channel with normal paging mode, the first mobile identity is another TMSI, the second is IMSI of the IUT (TSPX_IMSI).	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((9 + OC_LengthOf(MiImsi_01iei)) * 4) + 1, 1)	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100001'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	MiTmsi_r01	
mi2	MiImsi_01iei	
p1roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B2B2B2B'O, 13 - OC_LengthOf(MiImsi_01iei))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_r03	
PDU Type:	PG1_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE1 message requesting any channel with normal paging mode, the first mobile identity is an another TMSI differing from different from MiTmsi_01 and _02, the second one is TMSI of the IUT, MiTmsi_01.	
Field Name	**Field Value**	**Comments**
l2_pl	'41'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100001'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	MiTmsi_r01	
mi2	MiTmsi_01iei	
p1roct	'2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp1_r04	
PDU Type:	PG1_RQ_PDU	
Derivation Path:	PgReqTp1_01.	
Comments:	An PAGING REQUEST TYPE1 message requesting any channel with normal paging mode, the mobile identity is TMSI, but the identity type is set to 'no identity'.	
Field Name	**Field Value**	**Comments**
mi1	MiTmsi_r02	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_01	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the mobile identity is TMSI of the IUT.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_01	
mi2	Tmsi_r01	
mi3	-	
p2roct	'2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_02	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with extended paging mode and not addressing the MS under test.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_extended_paging)	
mi1	Tmsi_r01	
mi2	Tmsi_r03	
mi3	-	
p2roct	'2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_03	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with paging reorganisation mode, the mobile identity is TMSI of the IUT.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_reorg_paging)	
mi1	Tmsi_01	
mi2	Tmsi_r01	
mi3	-	
p2roct	'2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_04	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with same as before paging mode, the mobile identity is TMSI of the IUT.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_sab_paging)	
mi1	Tmsi_01	
mi2	Tmsi_r01	
mi3	-	
p2roct	'2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_r01	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the 1st mobile identity is TMSI of the IUT. The 2nd one addresses an another MS.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_01	
mi2	Tmsi_r01	
mi3	-	
p2roct	'2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_r02	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the 2nd mobile identity is TMSI of the IUT. The 1st one addresses an another MS.	
Field Name	**Field Value**	**Comments**
l2_pl	'2D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_r01	
mi2	Tmsi_01	
mi3	-	
p2roct	'2B2B2B2B2B2B2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_r03	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the 3 mobile identity is TMSI of the IUT. The 1st and 2nd one address an another MS.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_r03	
mi2	Tmsi_r01	
mi3	MiTmsi_01iei	
p2roct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_r04	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the 1st and 2nd mobile identity addresses an another MS. The 3rd one is IMSI of the IUT. .	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((11 + OC_LengthOf(MiImsi_01iei)) * 4) + 1, 1)	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_r03	
mi2	Tmsi_r01	
mi3	MiImsi_01iei	
p2roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B'O, 11 - OC_LengthOf(MiImsi_01iei))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_r05	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the 3 mobile identity is TMSI of the IUT with the type of no id. The 1st and 2nd one address an another MS.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_r03	
mi2	Tmsi_r01	
mi3	MiTmsi_r02iei	
p2roct	'2B2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp2_r06	
PDU Type:	PG2_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE2 message requesting any channel with normal paging mode, the 1st and 2nd mobile identity addresses an another MS. The 3rd one is IMSI of the IUT. .	
Field Name	**Field Value**	**Comments**
l2_pl	OC_IntToOct(((11 + OC_LengthOf(MiImsi_01iei)) * 4) + 1, 1)	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100010'B	
chn_m1_2	Chneed_01	
pm	Pm(C_extended_paging)	
mi1	Tmsi_r03	
mi2	Tmsi_r01	
mi3	MiImsi_01iei	
p2roct	OC_SubOctet('2B2B2B2B2B2B2B2B2B2B'O, 11 - OC_LengthOf(MiImsi_01iei))	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_01	
PDU Type:	PG3_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE3 message requesting any channel with extended paging mode and not adressing the MS under test.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100100'B	
chn_m1_2	Chneed_01	
pm	Pm(C_extended_paging)	
mi1	Tmsi_r05	
mi2	Tmsi_r01	
mi3	Tmsi_r03	
mi4	Tmsi_r04	
p3roct	'2B2B2B'O	
Detailed Comments:	Only used in 26_6_2_2 within Pg_Req3_01.	

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_02	
PDU Type:	PG3_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE3 message containing paging mode = "same as before".	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100100'B	
chn_m1_2	Chneed_01	
pm	Pm(C_sab_paging)	
mi1	Tmsi_r05	
mi2	Tmsi_r01	
mi3	Tmsi_r03	
mi4	Tmsi_r04	
p3roct	'2B2B2B'O	
Detailed Comments:	Only used in TC_26_6_2_4 within Pg_Req3_02	

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_03	
PDU Type:	PG3_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE3 message containing paging mode = "normal paging".	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100100'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_r05	
mi2	Tmsi_r01	
mi3	Tmsi_r03	
mi4	Tmsi_r04	
p3roct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_r01	
PDU Type:	PG3_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE3 message requesting any channel with normal paging mode, the 1st mobile identity is TMSI of the IUT. The 2nd, 3rd and 4th one address another MSs.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100100'B	
chn_m1_2	Chneed_01	
pm	Pm(C_normal_paging)	
mi1	Tmsi_01	
mi2	Tmsi_r01	
mi3	Tmsi_r03	
mi4	Tmsi_r04	
p3roct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_r02	
PDU Type:	PG3_RQ_PDU	
Derivation Path:	PgReqTp3_r01.	
Comments:	An PAGING REQUEST TYPE3 message requesting any channel with normal paging mode, the 2nd mobile identity is TMSI of the IUT. The 1st, 3rd and 4th one address another MSs.	
Field Name	**Field Value**	**Comments**
mi1	Tmsi_r01	
mi2	Tmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_r03	
PDU Type:	PG3_RQ_PDU	
Derivation Path:	PgReqTp3_r01.	
Comments:	An PAGING REQUEST TYPE3 message requesting any channel with normal paging mode, the 3rd mobile identity is TMSI of the IUT. The 1st, 2nd and 4th one address another MSs.	
Field Name	**Field Value**	**Comments**
mi1	Tmsi_r03	
mi3	Tmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_r04	
PDU Type:	PG3_RQ_PDU	
Derivation Path:	PgReqTp3_r01.	
Comments:	An PAGING REQUEST TYPE3 message requesting any channel with normal paging mode, the 4th mobile identity is TMSI of the IUT. The 1st, 2nd and 3rd one address another MSs.	
Field Name	**Field Value**	**Comments**
mi1	Tmsi_r04	
mi4	Tmsi_01	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	PgReqTp3_r05	
PDU Type:	PG3_RQ_PDU	
Derivation Path:		
Comments:	An PAGING REQUEST TYPE3 message requesting any channel with normal paging mode, the 1st mobile identity is TMSI of the IUT. The 2nd, 3rd and 4th one address another MSs.	
Field Name	**Field Value**	**Comments**
l2_pl	'4D'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00100100'B	
chn_m1_2	Chneed_01	
pm	Pm(C_extended_paging)	
mi1	Tmsi_01	
mi2	Tmsi_r01	
mi3	Tmsi_r03	
mi4	Tmsi_r04	
p3roct	'2B2B2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Phyinfor_01(ta:TA)	
PDU Type:	PHYINFO_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00101101'B	
ta	ta	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Release_02	
PDU Type:	REL_PDU	
Derivation Path:		
Comments:	A CC RELEASE message to match any received CC RELEASE message.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?101101'B	
cau	*	
cau2	*	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Release_08(Ti:TI)	
PDU Type:	REL_PDU	
Derivation Path:		
Comments:	A CC RELEASE message with mandatory IE's only used in structured procedures test .	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'00101101'B	
cau	-	
cau2	-	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Release_10(Ti:TI)	
PDU Type:	REL_PDU	
Derivation Path:		
Comments:	A CC RELEASE message used to match any received CC RELEASE message with a controllable TI.	
Field Name	**Field Value**	**Comments**
ti	Ti	
ccpd	'0011'B	
mt	'0?101101'B	
cau	*	
cau2	*	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmpRcv(Ti :TI; cau:CAU)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	A RELEASE COMPLETE message containing cause value = #81. ms -> n	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'0?101010'B	
cau	cau	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_03(Ti:TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	To match any received RELEASE message.	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'0?101010'B	
cau	*	
fie	*	
uu	*	
ssvi	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_08(Ti :TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	A RELEASE COMPLETE message containing mandatory IEs only. n -> ms	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'0011'B	
mt	'00101010'B	
cau	-	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_09(Ti :TI; fie:FIE)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	A RELEASE COMPLETE message containing facility IE. n -> ms	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'1011'B	
mt	'00101010'B	
cau	-	
fie	fie	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	ReleaseCmp_47(Ti:TI)	
PDU Type:	REL_COM_PDU	
Derivation Path:		
Comments:	To send a RELEASE COMPLETE message without cause and without FIE	
Field Name	**Field Value**	**Comments**
ti	Ti	
cc_sspd	'1011'B	
mt	'00101010'B	
cau	-	
fie	-	
uu	-	
ssvi	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_02(bcap: BCAP)	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	A SETUP message containing speech bearer capability. ms -> n	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	-	
bcap1	bcap	
bcap2	-	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	*	
llcmp1	*	
llcmp2	*	
hlcri	*	
hlcmp1	*	
hlcmp2	*	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	*	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_03(ti:TI)	
PDU Type:	SETUP_MT_PDU	
Derivation Path:		
Comments:	A SETUP message with mandatory IE's only. This is default for BIBO testing.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00000101'B	
bcri	-	
bcap1	-	
bcap2	-	
fie	-	
pi	-	
sig	-	
cgpn	-	
cgps	-	
cdpn	-	
cdps	-	
llcri	-	
llcmp1	-	
llcmp2	-	
hlcri	-	
hlcmp1	-	
hlcmp2	-	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_04(bcap: BCAP)	
PDU Type:	ESETUP_PDU	
Derivation Path:		
Comments:	A Emergency call SETUP message with/without speech bearer capability. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?001110'B	
bcap	bcap IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_10_UDI(bcap: BCAP; llc1:LLCMP; hlc1:HLCMP)	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	A SETUP message containing bearer capability supported by the mobile station, used for Unrestricted digital information transfer capability. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	-	
bcap1	bcap	
bcap2	-	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	-	
llcmp1	llc1	
llcmp2	-	
hlcri	-	
hlcmp1	hlc1 IF_PRESENT	
hlcmp2	-	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	CallCntrlCap IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_10_3100(bcap: BCAP; llc1:LLCMP; hlc1:HLCMP)	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	A SETUP message containing bearer capability supported by the mobile station. used for 3.1 kHz audio information transfer capability. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	-	
bcap1	bcap	
bcap2	-	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	-	
llcmp1	llc1 IF_PRESENT	
llcmp2	-	
hlcri	-	
hlcmp1	hlc1 IF_PRESENT	
hlcmp2	-	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	CallCntrlCap IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_11_3100(bcri1: RPI; bcap, bcap2nd: BCAP; llc1, llc2:LLCMP; llcri1, hlcri1:RPI; hlc1, hlc2:HLCMP)	
PDU Type:	SETUP_MO_PDU	
Derivation Path:		
Comments:	A SETUP message containing two bearer capabilities supported by the mobile station, used for 3.1 kHz audio information transfer capability. ms -> n.	
Field Name	**Field Value**	**Comments**
ti	?	
ccpd	'0011'B	
mt	'0?000101'B	
bcri	bcri1	
bcap1	bcap	
bcap2	bcap2nd	
fie	*	
cgps	*	
cdpn	?	
cdps	*	
llcri	llcri1 IF_PRESENT	
llcmp1	llc1 IF_PRESENT	
llcmp2	llc2 IF_PRESENT	
hlcri	hlcri1 IF_PRESENT	
hlcmp1	hlc1 IF_PRESENT	
hlcmp2	hlc2 IF_PRESENT	
uu	*	
ssvi	*	
clirsup	*	
clirinv	*	
cccap	CallCntrlCap IF_PRESENT	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_20(par: BCAP; ti:TI)	
PDU Type:	SETUP_MT_PDU	
Derivation Path:		
Comments:	A SETUP message containing bearer capability supported by the mobile station.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00000101'B	
bcri	-	
bcap1	par	
bcap2	-	
fie	-	
pi	-	
sig	-	
cgpn	-	
cgps	-	
cdpn	-	
cdps	-	
llcri	-	
llcmp1	-	
llcmp2	-	
hlcri	-	
hlcmp1	-	
hlcmp2	-	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	Setup_21(par1:RPI; par2, par3 :BCAP; ti:TI)	
PDU Type:	SETUP_MT_PDU	
Derivation Path:		
Comments:	A SETUP message containing two bearer capabilities supported by the mobile station.	
Field Name	**Field Value**	**Comments**
ti	ti	
ccpd	'0011'B	
mt	'00000101'B	
bcri	par1	
bcap1	par2	
bcap2	par3	
fie	-	
pi	-	
sig	-	
cgpn	-	
cgps	-	
cdpn	-	
cdps	-	
llcri	-	
llcmp1	-	
llcmp2	-	
hlcri	-	
hlcmp1	-	
hlcmp2	-	
uu	-	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SyncInfor(bcc:BCC; ncc:NCC)	
PDU Type:	SCHINFO_PDU	
Derivation Path:		
Comments:		
Field Name	**Field Value**	**Comments**
ncc	ncc	
bcc	bcc	
t1	?	
t2	?	
t3_	?	
Detailed Comments:	1. The values of t1, t2 and t3_ shall be correctly set by the L2 simulation module when this message is sent to air on the synchronization channel. It is assumed that there is a set of timebase counters in the L2 module, these counters run continuously and keep the correct timeslot number SN and TDMA frame number FN as long as the test system is poweron , the T1, T2, T3' (values for t1, t2 and t3_) can be derived from these counters.	

PDU Constraint Declaration		
Constraint Name:	SysInf1(cchd:CCHD; maxtx:B_2; txint:B_4; re:B_1)	
PDU Type:	SYSINFO1_PDU	
Derivation Path:		
Comments:	Default parameters for cell A in RR testing of GSM 900	
Field Name	**Field Value**	**Comments**
l2_pl	'55'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011001'B	
cchd	cchd	
rachcp	RachCntrlPara(maxtx, txint, re)	
si1roct	'2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf1_cba(cchd:CCHD; maxtx:B_2; txint:B_4; re:B_1; cba:B_1; acc:B_5)	
PDU Type:	SYSINFO1_PDU	
Derivation Path:		
Comments:	Default parameters for cell A in RR testing of GSM 900	
Field Name	**Field Value**	**Comments**
l2_pl	'55'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011001'B	
cchd	cchd	
rachcp	RachCntrlPara_cba(maxtx, txint, re, cba, acc)	
si1roct	'2B'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf2(bcchfl: NCD; maxtx:B_2; txint:B_4; re:B_1; nccp:NCCP)	
PDU Type:	SYSINFO2_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION TYPE2 containing default neighbour cells description	
Field Name	**Field Value**	**Comments**
l2_pl	'59'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011010'B	
bcchfl	bcchfl	
nccp	nccp	
rachcp	RachCntrlPara(maxtx, txint, re)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf2_cba(bcchfl: NCD; maxtx:B_2; txint:B_4; re:B_1; nccp:NCCP; cba:B_1; acc:B_5)	
PDU Type:	SYSINFO2_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION TYPE2 containing default neighbour cells description	
Field Name	**Field Value**	**Comments**
l2_pl	'59'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011010'B	
bcchfl	bcchfl	
nccp	nccp	
rachcp	RachCntrlPara_cba(maxtx, txint, re, cba, acc)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf2bis(bcchflegsm: NCD)	
PDU Type:	SYSINFO2bis_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION 2bis in cell A under EGSM with the ARFCN list = {988, 990, 1003}.	
Field Name	**Field Value**	**Comments**
l2_pl	'55'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00000010'B	
xbcchfl	bcchflegsm	
rachcp	RachCntrlPara_r01	
si2bisroct	'00'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf2bis_egsm(freqlst:NCD)	
PDU Type:	SYSINFO2bis_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION 2bis under EGSM with the ARFCN list = {985, 989, 995, 1010, 1014}.	
Field Name	**Field Value**	**Comments**
l2_pl	'55'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00000010'B	
xbcchfl	freqlst	
rachcp	RachCntrlPara_r01	
si2bisroct	'00'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf2ter(fl:NCD2)	
PDU Type:	SYSINFO2ter_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION 2ter with the ARFCN list = fl.	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00000011'B	
xbcchfl	fl	
si2terroct	'00000000'O	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf3(ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
PDU Type:	SYSINFO3_PDU	
Derivation Path:		
Comments:	CCCH combined or not with SDCCH	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011011'B	
ci	ci	
lai	LocAreaId(mcc, mnc, lac)	
ccd	ccd	
co	co	
csp	CellSelPara(crh, mtmc, neci)	
rachcp	RachCntrlPara(maxtx, txint, re)	
si3roct	SI3RestOctetPara('00000'B, '000000'B, '000'B, '00'B, '0'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf3_ram_cba(ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; cba, cbq:B_1; acc:B_5)	
PDU Type:	SYSINFO3_PDU	
Derivation Path:		
Comments:	CCCH combined or not with SDCCH	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011011'B	
ci	ci	
lai	LocAreaId(mcc, mnc, lac)	
ccd	ccd	
co	co	
csp	CellSelPara_ram(crh, mtmc, neci, ram)	
rachcp	RachCntrlPara_cba(maxtx, txint, re, cba, acc)	
si3roct	SI3RestOctetPara('00000'B, '000000'B, '000'B, '00'B, cbq)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf3_rpct(ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; pt:B_5; cro:B_6; to:B_3)	
PDU Type:	SYSINFO3_PDU	
Derivation Path:		
Comments:	CCCH combined or not with SDCCH	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011011'B	
ci	ci	
lai	LocAreaId(mcc, mnc, lac)	
ccd	ccd	
co	co	
csp	CellSelPara_ram(crh, mtmc, neci, ram)	
rachcp	RachCntrlPara(maxtx, txint, re)	
si3roct	SI3RestOctetPara(pt, cro, to, '00'B, '0'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf3_ti(ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; pt:B_5; cro:B_6; to:B_3)	
PDU Type:	SYSINFO3_PDU	
Derivation Path:		
Comments:	CCCH combined or not with SDCCH	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011011'B	
ci	ci	
lai	LocAreaId(mcc, mnc, lac)	
ccd	ccd	
co	co	
csp	CellSelPara_ram(crh, mtmc, neci, ram)	
rachcp	RachCntrlPara(maxtx, txint, re)	
si3roct	SI3RestOctetPara_ti(pt, cro, to, '00'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf3_rpctp(ci:CI; mcc, mnc, lac:OCTETSTRING; ccd:CCD; co:CO; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; pt:B_5; cro:B_6; to:B_3; po:B_2)	
PDU Type:	SYSINFO3_PDU	
Derivation Path:		
Comments:	CCCH combined or not with SDCCH	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011011'B	
ci	ci	
lai	LocAreaId(mcc, mnc, lac)	
ccd	ccd	
co	co	
csp	CellSelPara_ram(crh, mtmc, neci, ram)	
rachcp	RachCntrlPara(maxtx, txint, re)	
si3roct	SI3RestOctetPara(pt, cro, to, po, '0'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf4(mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1)	
PDU Type:	SYSINFO4_PDU	
Derivation Path:		
Comments:	A SYSTEM INFORMATION TYPE 4 message containing default values	
Field Name	**Field Value**	**Comments**
l2_pl	'31'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011100'B	
lai	LocAreaId(mcc, mnc, lac)	
csp	CellSelPara(crh, mtmc, neci)	
rachcp	RachCntrlPara(maxtx, txint, re)	
cbchd	-	
cbchma	-	
si4roct	SI4RestOctetPara('00000'B, '000000'B, '000'B, '00'B, '0'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf4_bis(mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6)	
PDU Type:	SYSINFO4_PDU	
Derivation Path:		
Comments:	A SYSTEM INFORMATION TYPE 4 message containing default values	
Field Name	**Field Value**	**Comments**
l2_pl	'31'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011100'B	
lai	LocAreaId(mcc, mnc, lac)	
csp	CellSelPara_bis(crh, mtmc, neci, ram)	
rachcp	RachCntrlPara(maxtx, txint, re)	
cbchd	-	
cbchma	-	
si4roct	SI4RestOctetDef	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf4_CBMS(mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; cbchd:CHD)	
PDU Type:	SYSINFO4_PDU	
Derivation Path:		
Comments:	To send a SYSTEM INFORMATION TYPE 4 message for SMSCB with the values or GSM 11.10, 34.3	
Field Name	**Field Value**	**Comments**
l2_pl	'49'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011100'B	
lai	LocAreaId(mcc, mnc, lac)	
csp	CellSelPara(crh, mtmc, neci)	
rachcp	RachCntrlPara(maxtx, txint, re)	
cbchd	cbchd	
cbchma	MoblAllc_01iei	
si4roct	SI4RestOctetPara('00000'B, '000000'B, '000'B, '00'B, '0'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf4_ram_cba(mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; cba, cbq:B_1; acc:B_5)	
PDU Type:	SYSINFO4_PDU	
Derivation Path:		
Comments:	A SYSTEM INFORMATION TYPE 4 message containing default values	
Field Name	**Field Value**	**Comments**
l2_pl	'31'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011100'B	
lai	LocAreaId(mcc, mnc, lac)	
csp	CellSelPara_ram(crh, mtmc, neci, ram)	
rachcp	RachCntrlPara_cba(maxtx, txint, re, cba, acc)	
cbchd	-	
cbchma	-	
si4roct	SI4RestOctetPara('00000'B, '000000'B, '000'B, '00'B, cbq)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf4_rpct(mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; pt:B_5; cro:B_6; to:B_3)	
PDU Type:	SYSINFO4_PDU	
Derivation Path:		
Comments:	A SYSTEM INFORMATION TYPE 4 message containing default values	
Field Name	**Field Value**	**Comments**
l2_pl	'31'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011100'B	
lai	LocAreaId(mcc, mnc, lac)	
csp	CellSelPara_ram(crh, mtmc, neci, ram)	
rachcp	RachCntrlPara(maxtx, txint, re)	
cbchd	-	
cbchma	-	
si4roct	SI4RestOctetPara(pt, cro, to, '00'B, '0'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf4_rpctp(mcc, mnc, lac:OCTETSTRING; crh, mtmc:INTEGER; neci:B_1; maxtx:B_2; txint:B_4; re:B_1; ram:B_6; pt:B_5; cro:B_6; to:B_3; po:B_2)	
PDU Type:	SYSINFO4_PDU	
Derivation Path:		
Comments:	A SYSTEM INFORMATION TYPE 4 message containing default values	
Field Name	**Field Value**	**Comments**
l2_pl	'31'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011100'B	
lai	LocAreaId(mcc, mnc, lac)	
csp	CellSelPara_ram(crh, mtmc, neci, ram)	
rachcp	RachCntrlPara(maxtx, txint, re)	
cbchd	-	
cbchma	-	
si4roct	SI4RestOctetPara(pt, cro, to, po, '0'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf5(bcchfl: NCD)	
PDU Type:	SYSINFO5_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION TYPE 5 containing default neighbour cells description for GSM900	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00011101'B	
bcchfl	bcchfl	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf5bis(bcchfl: NCD)	
PDU Type:	SYSINFO5bis_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION TYPE 5bis containing partial neighbour cells description with 1 frequency for DCS1800.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00000101'B	
xbcchfl	bcchfl	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf5ter(bcchfl: NCD2)	
PDU Type:	SYSINFO5ter_PDU	
Derivation Path:		
Comments:	SYSTEM INFORMATION TYPE 5ter containing partial neighbour cells description.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00000110'B	
xbcchfl	bcchfl	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf6(ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; nccp:NCCP)	
PDU Type:	SYSINFO6_PDU	
Derivation Path:		
Comments:	A SYSTEM INFORMATION TYPE 6 message containing default parameters for RR testing.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
rrpd	'0110'B	
mt	'00011110'B	
ci	ci	
lai	LocAreaId(mcc, mnc, lac)	
co	co	
nccp	nccp	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf7(pt:B_5; cro:B_6; to:B_3)	
PDU Type:	SYSINFO7_PDU	
Derivation Path:		
Comments:	A SYSTEM INFORMATION TYPE 7 message containing default values	
Field Name	**Field Value**	**Comments**
l2_pl	'01'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011111'B	
si7roct	SI7RestOctetPara(pt, cro, to, '00'B, '0'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	SysInf8(pt:B_5; cro:B_6; to:B_3)	
PDU Type:	SYSINFO8_PDU	
Derivation Path:		
Comments:	A SYSTEM INFORMATION TYPE 8 message containing default values	
Field Name	**Field Value**	**Comments**
l2_pl	'01'O	
ski	'0000'B	
rrpd	'0110'B	
mt	'00011000'B	
si8roct	SI7RestOctetPara(pt, cro, to, '00'B, '0'B)	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	TmsiReallocCmd(par:MI; mcc, mnc, lac: OCTETSTRING)	
PDU Type:	TMSIRE_CMD_PDU	
Derivation Path:		
Comments:	TMSI REALLOCATION COMMAND message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'00011010'B	
lai	LocAreaId(mcc, mnc, lac)	
mi	par	
Detailed Comments:		

PDU Constraint Declaration		
Constraint Name:	TmsiReallocComp	
PDU Type:	TMSIRE_COM_PDU	
Derivation Path:		
Comments:	TMSI REALLOCATION COMPLETE message.	
Field Name	**Field Value**	**Comments**
ski	'0000'B	
mmpd	'0101'B	
mt	'0?011011'B	
Detailed Comments:		

Dynamic Part

Test Cases

Test Group CellSelection

Test Case Dynamic Behaviour					
Test Case Name:	TC_20_1				
Group:	GSM_MS_CS_4220/CellSelection/				
Purpose:	1. To verify that: The MS shall be able to select the correct (forth strongest) cell and be able to respond to paging on that cell within 30 seconds of switch on, when the three strongest cells are not suitable. This assumes that a valid SIM, with PIN disabled and ideal radio conditions; GSM 05.08, 6.1. 2. To verify that: 2.1 The MS does not select a cell of a PLMN which is not the selected PLMN. 2.2 The MS does not select a cell which is "barred". 2.3 The MS does not select a cell with C1<0. 3. To verify that the MS selects suitable cells in descending order of received signal strength. 4. To verify that the MS does not select a cell with C1<0.				
Default:	OtherEventsFail				
Comments:	This test case achieves the test purpose in one phase rather than the two described in ETS 300-607-2				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_Omit, BcchFreqLst_51, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		+ltree_StartCarriers			
5		+PageOnCarriers(C_CellA, C_CellB, C_CellC, C_CellD, "dummy", "dummy")			
6		+MM_PwrOrSimOn(C_SIMIn)			
7		START T_dly(33000)			
8	CS_0001	?TIMEOUT T_dly		F	
9	CS_0002	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)	F	
10	CS_0003	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
11	CS_0004	L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)	F	
12		L?DL_RacInChRq	ChReq_01(C_RACH_D_1, ChRequest_02)		
13		+MM_PwrOrSimOff(C_SIMIn)			
14		(TCV_Null := OM_StopCell(C_CellB))			
15		+ltree_StartCarrier5			
16		+PageOnCarriers("dummy", "dummy", "dummy", "dummy", C_CellE, "dummy")			
17		+MM_PwrOrSimOn(C_SIMIn)			
18		START T_dly(33000)			
19	Label1	L?DL_RacInChRq	ChReq_01(C_RACH_D_1, ChRequest_02)		

20		->Label1			
21		L?DL_RacInChRq	ChReq_01(C_RACH_E_1, ChRequest_02)		
22		->Label1			
23	CS_0005	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)	F	
24	CS_0006	L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)	F	
25	CS_0007	?TIMEOUT T_dly		P	
		ltree_StartCarriers			
26		+ltree_StartCarrier1			
27		+ltree_StartCarrier2			
28		+ltree_StartCarrier3			
29		+ltree_StartCarrier4			
		ltree_StartCarrier1			
30		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
31		+CombinedBCCH_A(48, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
32		+SysInfoSending_fh_rcc(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '1'B, '0'B, '00000'B)			
		ltree_StartCarrier2			
33		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
34		[TSPC_EGSM]			
35		+CombinedBCCH_A(48, C_MaxPwrLvlG, C_MaxPwrLvlD, 985, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
36		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '101010'B, '0'B, '0'B, '00000'B)			
37		[NOT TSPC_EGSM]			
38		+CombinedBCCH_B(38, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
39		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '101010'B, '0'B, '0'B, '00000'B)			

ltree_StartCarrier3

(TCV_sacch := OC_SubchOfSacch4(
TSPX_SDCCH4SubDef, C_CellC))

+CombinedBCCH_C(43, C_MaxPwrLvlG,
C_MaxPwrLvlD, 18, 568, C_Immass, TCV_slot,
TCV_tsc, TimingAdv(0), C_BABR_0,
C_cch_1Comb, C_BPM_3)

+SysInfoSending_fh_rcc(C_SCH_C,
C_BCCH_C_1, C_TxInt_5, C_Max_2, C_NECI_0,
C_ATT_0, C_BABR_0, C_cch_1Comb,
C_BPM_3, C_T3212_0, C_ci_cellC, C_MCC_2,
C_PLMN_1, C_LAC_1111, CellOpt_01,
CellChDes_24, CellChDes_24d,
C_CellReselectHys4, C_MaxPwrLvlG,
C_MaxPwrLvlD, BcchFreqLst_50,
BcchFreqLst_51, BcchFreqLst_52,
C_Restablishment, C_BCC, C_NCC, C_NCCP_2,
'010100'B, '0'B, '0'B, '00000'B)

ltree_StartCarrier4

(TCV_sacch := OC_SubchOfSacch4(
TSPX_SDCCH4SubDef, C_CellD))

[TSPC_EGSM]

+CombinedBCCH_D(33, C_MaxPwrLvlG,
C_MaxPwrLvlD, 989, 589, C_Immass, TCV_slot,
TCV_tsc, TimingAdv(0), C_BABR_0,
C_cch_1Comb, C_BPM_3)

+SysInfoSending_fh_rcc(C_SCH_D,
C_BCCH_D_1, C_TxInt_5, C_Max_2,
C_NECI_0, C_ATT_0, C_BABR_0,
C_cch_1Comb, C_BPM_3, C_T3212_0,
C_ci_cellD, C_MCC_1, C_PLMN_1,
C_LAC_1111, CellOpt_01, CellChDes_24,
CellChDes_24d, C_CellReselectHys4,
C_MaxPwrLvlG, C_MaxPwrLvlD,
BcchFreqLst_50, BcchFreqLst_51,
BcchFreqLst_52, C_Restablishment, C_BCC,
C_NCC, C_NCCP_2, '010110'B, '0'B, '0'B,
'00000'B)

[NOT TSPC_EGSM]

+CombinedBCCH_D(33, C_MaxPwrLvlG,
C_MaxPwrLvlD, 25, 589, C_Immass, TCV_slot,
TCV_tsc, TimingAdv(0), C_BABR_0,
C_cch_1Comb, C_BPM_3)

+SysInfoSending_fh_rcc(C_SCH_D,
C_BCCH_D_1, C_TxInt_5, C_Max_2,
C_NECI_0, C_ATT_0, C_BABR_0,
C_cch_1Comb, C_BPM_3, C_T3212_0,
C_ci_cellD, C_MCC_1, C_PLMN_1,
C_LAC_1111, CellOpt_01, CellChDes_24,
CellChDes_24d, C_CellReselectHys4,
C_MaxPwrLvlG, C_MaxPwrLvlD,
BcchFreqLst_50, BcchFreqLst_51,
BcchFreqLst_52, C_Restablishment, C_BCC,
C_NCC, C_NCCP_2, '010110'B, '0'B, '0'B,
'00000'B)

ltree_StartCarrier5

(TCV_sacch := OC_SubchOfSacch4(
TSPX_SDCCH4SubDef, C_CellE))

+CombinedBCCH_E(28, C_MaxPwrLvlG,
C_MaxPwrLvlD, 41, 602, C_Immass, TCV_slot,
TCV_tsc, TimingAdv(0), C_BABR_0,
C_cch_1Comb, C_BPM_3)

+SysInfoSending_fh_rcc(C_SCH_E,
C_BCCH_E_1, C_TxInt_5, C_Max_2, C_NECI_0,
C_ATT_0, C_BABR_0, C_cch_1Comb,
C_BPM_3, C_T3212_0, C_ci_cellE, C_MCC_1,
C_PLMN_1, C_LAC_1111, CellOpt_01,
CellChDes_24, CellChDes_24d,
C_CellReselectHys4, C_MaxPwrLvlG,
C_MaxPwrLvlD, BcchFreqLst_50,
BcchFreqLst_51, BcchFreqLst_52,

		C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '001100'B, '0'B, '0'B, '00000'B)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_20_2

Group: GSM_MS_CS_4220/CellSelection/

Purpose: 1. The MS shall:

1.1 The MS shall search all RF channels in the system, take readings of received RF signal strength on each RF channel, and calculate the received level average for each.

1.2 The averaging is based on at least five measurement samples per RF carrier spread over Tav (3 to 5 s).

1.3 The measurement samples from the different RF carriers being spread evenly during this period.

2. These quantities are termed the "receive level averages", shall be unweighted averages of the received signal strengths measured in dBm.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_Omit, BcchFreqLst_51, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		+ltree_StartCarriers			
5		+PageOnCarriers(C_CellA, C_CellB, "dummy", "dummy", "dummy", "dummy")			
6		(TCV_Null := OM_VaryRFOfCell(C_BCCH_B_1, TSPX_bTav, 28, TSPX_aTav, 58))			
7		+MM_PwrOrSimOn(C_SIMIn)			
8		START T_dly(33000)			
9	CS_0008	?TIMEOUT T_dly		F	
10	CS_0009	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)	F	
11		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
12	CS_0010	(TCV_Null := OM_StopVaryRFOfCell(C_BCCH_B_1))		P	
		ltree_StartCarriers			
13		+ltree_StartCarrier1			
14		+ltree_StartCarrier2			
		ltree_StartCarrier1			
15		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
16		[TSPC_EGSM]			
17		+CombinedBCCH_A(23, C_MaxPwrLvlG, C_MaxPwrLvlD, 985, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
18		+SysInfoSending_fh_rcc(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4,			

		C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '001010'B, '0'B, '0'B, '00000'B)			
19		[NOT TSPC_EGSM]			
20		+CombinedBCCH_A(23, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
21		+SysInfoSending_fh_rcc(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '001010'B, '0'B, '0'B, '00000'B)			
		ltree_StartCarrier2			
22		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
23		+CombinedBCCH_B(58, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
24		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '001010'B, '0'B, '0'B, '00000'B)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_20_4

Group: GSM_MS_CS_4220/CellSelection/

Purpose: 1. To verify that the MS shall be able to calculate the path loss criterion parameter C2 used for cell reselection.

2. To verify that the MS can correctly calculate the C2 parameter correctly when the CELL_RESELECT_OFFSET, TEMPORARY_OFFSET and PENALTY_TIME parameters are used.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_Omit, BcchFreqLst_51, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		(TCV_K := 1)			
5		+ltree_TestReselect			
6		[TSPC_DCS_Class3]			
7		(TCV_K := 2)			
8		+ltree_TestReselect			
		ltree_TestReselect			
9		+ltree_StartCarriers1_and_2			
10		+PageOnCarriers("dummy", C_CellB, "dummy", "dummy", "dummy", "dummy")			
11		+MM_PwrOrSimOn(C_SIMIn)			
12		START T_dly(50000)			
13	Label 1	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
14		->Label1			
15	CS_0011	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
16		?TIMEOUT T_dly			
17		(TCV_Null := OM_ChangeRFOfCell(C_CellB, 54))			
18		START T_dly(20000)			
19	Label 2	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_01)		
20		->Label2			
21	CS_0012	?TIMEOUT T_dly		F	
22		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
23		CANCEL T_dly			
24		(TCV_Null := OM_PgFill(C_CellB, PgReqTp1Norm))			
25		+Pause(20000)			
26		+ltree_StartCarriers3_and_4			
27		+PageOnCarriers("dummy", "dummy", C_CellC, C_CellD, "dummy", "dummy")			
28		START T_dly(38000)			
29	Label 3	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_01)		
30		->Label3			
31		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_01)		
32		->Label3			

33	CS_0013		L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)	F	
34	CS_0014		L?DL_RacInChRq	ChReq_01(C_RACH_D_1, ChRequest_02)	F	
35			?TIMEOUT T_dly			
36			START T_dly(52000)			
37	Label4		L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_01)		
38			->Label4			
39			L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_01)		
40			->Label4			
41	CS_0015		?TIMEOUT T_dly		F	
42	CS_0016		L?DL_RacInChRq	ChReq_01(C_RACH_D_1, ChRequest_02)	F	
43	CS_0017		L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)	P	
		ltree_StartCarriers1_and_2				
44		(TCV_PO := '00'B)				
45		+ltree_StartCarrier1				
46		[TCV_K = 1]				
47		(TCV_PO := '00'B)				
48		+ltree_StartCarrier2				
49		[TCV_K <> 1]				
50		(TCV_PO := '01'B)				
51		+ltree_StartCarrier2				
		ltree_StartCarriers3_and_4				
52		[TCV_K = 1]				
53		(TCV_PO := '00'B)				
54		+ltree_StartCarrier3				
55		+ltree_StartCarrier4				
56		[TCV_K <> 1]				
57		(TCV_PO := '11'B)				
58		+ltree_StartCarrier3				
59		+ltree_StartCarrier4				
		ltree_StartCarrier1				
60		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))				
61		+CombinedBCCH_A(53, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)				
62		+SysInfoSending_fh_rpctp(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_53, BcchFreqLst_54, BcchFreqLst_55, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '011110'B, '00000'B, '000000'B, '000'B, TCV_PO)				
		ltree_StartCarrier2				
63		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))				
64		+CombinedBCCH_B(43, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)				
65		+SysInfoSending_fh_rpctp(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01,				

		CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_53, BcchFreqLst_54, BcchFreqLst_55, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '001010'B, '11111'B, '001000'B, '000'B, TCV_PO)			
		ltree_StartCarrier3			
66		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellC))			
67		+CombinedBCCH_C(48, C_MaxPwrLvlG, C_MaxPwrLvlD, 18, 568, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
68		+SysInfoSending_fh_rpctp(C_SCH_C, C_BCCH_C_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellC, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_53, BcchFreqLst_54, BcchFreqLst_55, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '011001'B, '00001'B, '001010'B, '010'B, TCV_PO)			
		ltree_StartCarrier4			
69		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellD))			
70		[TSPC_EGSM]			
71		+CombinedBCCH_D(48, C_MaxPwrLvlG, C_MaxPwrLvlD, 985, 589, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
72		+SysInfoSending_fh_rpctp(C_SCH_D, C_BCCH_D_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellD, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_53, BcchFreqLst_54, BcchFreqLst_55, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '011001'B, '00010'B, '001010'B, '010'B, TCV_PO)			
73		[NOT TSPC_EGSM]			
74		+CombinedBCCH_D(48, C_MaxPwrLvlG, C_MaxPwrLvlD, 25, 589, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
75		+SysInfoSending_fh_rpctp(C_SCH_D, C_BCCH_D_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellD, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_53, BcchFreqLst_54, BcchFreqLst_55, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '011001'B, '00010'B, '001010'B, '010'B, TCV_PO)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_20_5

Group: GSM_MS_CS_4220/CellSelection/

Purpose: 1. To verify that the MS shall be able to calculate the path loss criterion parameter C2 used for cell reselection.

2. Whilst in idle mode, an MS shall continue to monitor all BCCH carriers as indicated by the BCCH allocation.

3. Mobile stations shall treat all ARFCNs in the set (0..1023) as valid ARFCN values even if the mobile station is unable to transmit or receive on that ARFCN.

4. An E-GSM MS shall correctly decode parameters transmitted in the system information type 2 ter message.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_1, C_cch_1Comb, C_BPM_2, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_Omit, BcchFreqLst_51, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		+ltree_StartCarriers			
5		[TSPC_EGSM]			
6		+PageOnCarriers("dummy", C_CellB, C_CellC, "dummy", "dummy", "dummy")			
7		+ltree_continue			
8		[NOT TSPC_EGSM]			
9		+PageOnCarriers("dummy", C_CellB, "dummy", "dummy", "dummy", "dummy")			
10		+ltree_continue			
		ltree_continue			
11		+MM_PwrOrSimOn(C_SIMIn)			
12		START T_dly(50000)			
13	CS_0018	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
14	CS_0019	L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)	F	
15		?TIMEOUT T_dly			
16		(TCV_Null := OM_ChangeRFOfCell(C_CellB, 42))			
17		START T_dly(20000)			
18		L?DL_RacInChRq CANCEL T_dly	ChReq_01(C_RACH_B_1, ChRequest_02)		
19		[TSPC_EGSM]			
20		+Pause(30000)			
21		(TCV_Null := OM_ChangeRFOfCell(C_CellC, 60))			
22		+Pause(20000)			
23	CS_0020	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_01)	I	
24	CS_0021	L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)	P	
25	CS_0022	[NOT TSPC_EGSM]		P	
		ltree_StartCarriers			

26		+ltree_StartCarrier1			
27		+ltree_StartCarrier2			
28		[TSPC_EGSM]			
29		+ltree_StartCarrier3			
30		[NOT TSPC_EGSM]			
		ltree_StartCarrier1			
31		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
32		+CombinedBCCH_A_fn(53, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_1, C_cch_1Comb, C_BPM_2, 11)			
33		+SysInfoSending_fh_bisA(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_1, C_cch_1Comb, C_BPM_2, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '00000'B, '000000'B, '000'B)			
		ltree_StartCarrier2			
34		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
35		+CombinedBCCH_B_fn(32, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_1, C_cch_1Comb, C_BPM_2, 27)			
36		+SysInfoSending_fh_bisB(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_1, C_cch_1Comb, C_BPM_2, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '00000'B, '001000'B, '000'B)			
		ltree_StartCarrier3			
37		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellC))			
38		+CombinedBCCH_C(43, C_MaxPwrLvlG, C_MaxPwrLvlD, 985, 568, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_1, C_cch_1Comb, C_BPM_2)			
39		+SysInfoSending_fh_rpct(C_SCH_C, C_BCCH_C_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_1, C_cch_1Comb, C_BPM_2, C_T3212_0, C_ci_cellC, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '011011'B, '00000'B, '000101'B, '000'B)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_20_6

Group: GSM_MS_CS_4220/CellSelection/

Purpose: 1. At least every 5 s the MS shall calculate the value of C1 and C2 for the serving cell and recalculate C1 and C2 values for non serving cells (if necessary). The MS shall then check whether:

1.1 The calculated value of C2 for a non serving suitable cell exceeds the value of C2 for the serving cell for a period of 5 seconds.

1.2 Cell reselection shall not take place if there was a cell reselection within the previous 15 seconds.

1.3 Cell reselection for any other reason shall take place immediately, but the cell that the MS was camped on shall not be returned to within 5 seconds if another suitable cell can be found.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_59, BcchFreqLst_Omit, BcchFreqLst_61, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		+ltree_StartCarriers1_and_2			
5		+MM_PwrOrSimOn(C_SIMIn)			
6		+NoReaction(50000)			
7		+PageOnCarriers(C_CellA, C_CellB, "dummy", "dummy", "dummy", "dummy")			
8		START T_dly(20000)			
9		+ltree_ChRq1_and_2			
10		+ltree_continue			
11		+ltree_continue1			
		ltree_continue			
12		(TCV_Null := OM_PgFill(C_CellA, PgReqTp1Norm), TCV_Null := OM_PgFill(C_CellB, PgReqTp1Norm))			
13		START T_dly(20000)			
14	Label 1	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
15		->Label1			
16		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
17		->Label1			
18		?TIMEOUT T_dly			
19		(TCV_Null := OM_PgFill(C_CellB, PgReqTp1_01), TCV_Null := OM_ChangeRFOfCell(C_CellB, 61))			
20		START T_dly(4000)			
21	Label 2	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
22		->Label2			
23	CS_0023	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
24		?TIMEOUT T_dly			
25		(TCV_Null := OM_ChangeRFOfCell(C_CellB, 41))			
26		START T_dly(30000)			
27	Label 3	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
28		->Label3			

29	CS_0024	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
30		?TIMEOUT T_dly			
31		(TCV_Null := OM_ChangeRFOfCell(C_CellB, 61))			
32		START T_dly(100000)			
33	Label4	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
34		->Label4			
35	CS_0025	?TIMEOUT T_dly		F	
36		L?DL_RacInChRq READTIMER T_dly(TCV_Time)	ChReq_01(C_RACH_B_1, ChRequest_02)		
37		CANCEL T_dly			
	ltree_continue1				
38	(TCV_Null := OM_PgFill(C_CellB, PgReqTp1Norm), TCV_Null := OM_ChangeRFOfCell(C_CellB, 41))				
39	START T_dly(20000)				
40	Label5	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
41		->Label5			
42		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
43		->Label5			
44		?TIMEOUT T_dly			
45		(TCV_Time := TCV_Time + 2000, TCV_Null := OM_ChangeRFOfCell(C_CellB, 61))			
46		START T_dly(TCV_Time)			
47	Label6	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
48		->Label6			
49		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
50		->Label6			
51		?TIMEOUT T_dly			
52		(TCV_Null := OM_PgFill(C_CellA, PgReqTp1_01), TCV_Null := OM_ChangeRFOfCell(C_CellB, 41))			
53		START T_dly(11000)			
54	Label7	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
55		->Label7			
56	CS_0026	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)	F	
57	CS_0027	?TIMEOUT T_dly		P	
	ltree_ChRq1_and_2				
58	(TCV_Cnt := 0)				
59	Label8	(TCV_Cnt := TCV_Cnt)			dummy line to allow label
60	CS_0028	?TIMEOUT T_dly		F	
61	CS_0029	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_01)	F	
62		L?DL_RacInChRq (TCV_Cnt := TCV_Cnt + 1)	ChReq_01(C_RACH_A_1, ChRequest_02)		
63		[TCV_Cnt = 1]			Rxed 2 RACHs on cell A?
64		->Label8			
65		[TCV_Cnt <> 1]			
66		START T_dly1(4500)			
67		(TCV_Cnt := 0)			
68	Label9	(TCV_Cnt := TCV_Cnt)			dummy line to allow label
69	CS_0030	?TIMEOUT T_dly		F	
70		?TIMEOUT T_dly1			

71		->Label9			
72		L?DL_RacInChRq READTIMER T_dly1(TCV_Time)	ChReq_01(C_RACH_A_1, ChRequest_02)		
73	CS_0031	[TCV_Time <> 0]		F	
74		[TCV_Time = 0]			
75		->Label9			
76		L?DL_RacInChRq (TCV_Cnt := TCV_Cnt + 1)	ChReq_01(C_RACH_B_1, ChRequest_02)		
77		[TCV_Cnt = 1]			Rxed 2 RACHs on cell B?
78		->Label9			
79		[TCV_Cnt <> 1]			
80	Label10	L?DL_RacInChRq READTIMER T_dly1(TCV_Time)	ChReq_01(C_RACH_A_1, ChRequest_01)		
81	CS_0032	[TCV_Time <> 0]		F	
82		[TCV_Time = 0]			
83		->Label10			
84		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_01)		
85		->Label10			
86		?TIMEOUT T_dly1			
87		->Label10			
88		?TIMEOUT T_dly			
89		CANCEL T_dly1			
		ltree_StartCarriers1_and_2			
90		+ltree_StartCarrier1			
91		+ltree_StartCarrier2			
		ltree_StartCarrier1			
92		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
93		[TSPC_EGSM]			
94		+CombinedBCCH_A(51, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
95		+SysInfoSending_fh_rcc(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_60, BcchFreqLst_61, BcchFreqLst_55, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010101'B, '0'B, '0'B, '00000'B)			
96		[NOT TSPC_EGSM]			
97		+CombinedBCCH_A(51, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
98		+SysInfoSending_fh_rcc(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_59, BcchFreqLst_61, BcchFreqLst_55, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010101'B, '0'B, '0'B, '00000'B)			
		ltree_StartCarrier2			
99		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			

100		[TSPC_EGSM]			
101		+CombinedBCCH_B(41, C_MaxPwrLvlG, C_MaxPwrLvlD, 985, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
102		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_60, BcchFreqLst_61, BcchFreqLst_55, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '011001'B, '0'B, '0'B, '00000'B)			
103		[NOT TSPC_EGSM]			
104		+CombinedBCCH_B(41, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
105		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_1, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_59, BcchFreqLst_61, BcchFreqLst_55, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '011001'B, '0'B, '0'B, '00000'B)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_20_7

Group: GSM_MS_CS_4220/CellSelection/

Purpose: 1. To verify that the MS does not select a cell of low priority when a suitable cell of normal priority exists with a lower received signal strength.

2. To verify that the MS takes into account CELL_BAR_ACCESS and CELL_BAR_QUALIFY when performing cell selection and reselection.

3. To verify that if the following conditions are met:

- the cell belongs to the MS HPLMN
- the MS is in cell test operation mode
- the CELL_BAR_ACCESS is set to 1
- the CELL_BAR_QUALIFY is set to 0
- the Access Control class 15 is barred

then the "Cell Selection priority" and the "Status for cell reselection" shall be set to normal.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_Omit, BcchFreqLst_51, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		+ltree_StartCarriers			
5		+PageOnCarriers(C_CellA, C_CellB, C_CellC, C_CellD, "dummy", "dummy")			
6		+MM_PwrOrSimOn(C_SIMIn)			
7		START T_dly(33000), START T_dly1(50000)			
8	CS_0033	?TIMEOUT T_dly		F	
9	CS_0034	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)	F	
10	CS_0035	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
11	CS_0036	L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)	F	
12		L?DL_RacInChRq	ChReq_01(C_RACH_D_1, ChRequest_02)		
13		CANCEL T_dly			
14	Label1	L?DL_RacInChRq	ChReq_01(C_RACH_D_1, ChRequest_02)		
15		->Label1			
16	CS_0037	?TIMEOUT T_dly1		F	
17	CS_0038	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
18	CS_0039	L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)	F	
19		L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
20	Label2	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
21		->Label2			
22	CS_0040	L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)	F	

23	CS_0041	L?DL_RacInChRq	ChReq_01(C_RACH_D_1, ChRequest_02)	F	
24		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
25		+ltree_continue			
26		?TIMEOUT T_dly			
27		+ltree_continue			
		ltree_continue			
28		CANCEL T_dly, CANCEL T_dly1			
29		+MM_PwrOrSimOff(C_SIMIn)			
30		(TCV_Null := OM_StopAllBCCH(C_CellA, C_CellB, C_CellC, C_CellD, "dummy", "dummy", "dummy", "dummy"))			
31		(TCV_Null := OO_SetCellTestMode())			Switch MS to cell test mode
32		+ltree_StartCarriers			
33		+PageOnCarriers(C_CellA, C_CellB, C_CellC, C_CellD, "dummy", "dummy")			
34		+MM_PwrOrSimOn(C_SIMIn)			
35		START T_dly(33000), START T_dly1(50000)			
36	CS_0042	?TIMEOUT T_dly		F	
37	CS_0043	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)	F	
38	CS_0044	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
39	CS_0045	L?DL_RacInChRq	ChReq_01(C_RACH_D_1, ChRequest_02)	F	
40		L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)		
41		CANCEL T_dly			
42	Label3	L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)		
43		->Label3			
44	CS_0046	?TIMEOUT T_dly1		F	
45	CS_0047	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
46	CS_0049	L?DL_RacInChRq	ChReq_01(C_RACH_D_1, ChRequest_02)	F	
47		L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
48	Label4	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
49		->Label4			
50		L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)		
51		->Label4			
52		L?DL_RacInChRq	ChReq_01(C_RACH_D_1, ChRequest_02)		
53		->Label4			
54	CS_0050	?TIMEOUT T_dly1		P	
55	CS_0051	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	P	
		ltree_StartCarriers			
56		+ltree_StartCarrier1			
57		+ltree_StartCarrier2			
58		+ltree_StartCarrier3			
59		+ltree_StartCarrier4			
		ltree_StartCarrier1			
60		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
61		+CombinedBCCH_A(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot,			

		TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
62		+SysInfoSending_fh_rcc(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '000000'B, '0'B, '1'B, '10000'B)			
		ltree_StartCarrier2			
63		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
64		+CombinedBCCH_B(43, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
65		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '1'B, '1'B, '10000'B)			
		ltree_StartCarrier3			
66		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellC))			
67		+CombinedBCCH_C(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 18, 568, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
68		+SysInfoSending_fh_rcc(C_SCH_C, C_BCCH_C_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellC, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '001010'B, '1'B, '0'B, '10000'B)			
		ltree_StartCarrier4			
69		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellD))			
70		[TSPC_EGSM]			
71		+CombinedBCCH_D(23, C_MaxPwrLvlG, C_MaxPwrLvlD, 985, 589, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
72		+SysInfoSending_fh_rcc(C_SCH_D, C_BCCH_D_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellD, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '001010'B, '0'B, '0'B, '10000'B)			
73		[NOT TSPC_EGSM]			
74		+CombinedBCCH_D(23, C_MaxPwrLvlG,			

		C_MaxPwrLvlD, 25, 589, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
75		+SysInfoSending_fh_rcc(C_SCH_D, C_BCCH_D_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellD, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '001010'B, '0'B, '0'B, '10000'B)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_20_8

Group: GSM_MS_CS_4220/CellSelection/

Purpose: 1. To verify that at least every 5 s the MS shall calculate the value of C1 and C2 for the serving cell and recalculate the C1 and C2 for non serving cells (if necessary). The MS shall then check whether the path loss criterion (C1) for current serving cell falls below zero for a period of 5 seconds. This indicates that the path loss has become too high.

2. To verify that whilst camped on a cell of the selected PLMN, the MS may need to select a different cell. The path loss criterion (C1) for the current serving cell falling below zero triggers cell reselection.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_Omit, BcchFreqLst_51, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		+ltree_StartCarriers			
5		+PageOnCarriers("dummy", C_CellB, "dummy", "dummy", "dummy", "dummy")			
6		+MM_PwrOrSimOn(C_SIMIn)			
7		START T_dly(50000)			
8	CS_0052	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)	F	
9	CS_0053	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
10		?TIMEOUT T_dly			
11		(TCV_Null := OM_ChangeRFOfCell(C_CellA, 33))			
12		+Pause(4000)			
13		(TCV_Null := OM_ChangeRFOfCell(C_CellA, 63))			
14		START T_dly(30000)			
15	Label1	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
16		->Label1			
17	CS_0054	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
18		?TIMEOUT T_dly			
19		(TCV_Null := OM_ChangeRFOfCell(C_CellA, 33))			
20		START T_dly(20000)			
21	Label2	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
22		->Label2			
23	CS_0055	?TIMEOUT T_dly		F	
24	CS_0056	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	P	
		ltree_StartCarriers			
25		+ltree_StartCarrier1			
26		+ltree_StartCarrier2			
		ltree_StartCarrier1			

27		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
28		+CombinedBCCH_A(63, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
29		+SysInfoSending_fh_rpct(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '101000'B, '00000'B, '001111'B, '000'B)			
		ltree_StartCarrier2			
30		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
31		[TSPC_EGSM]			
32		+CombinedBCCH_B(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
33		+SysInfoSending_fh_rpct(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '00000'B, '000000'B, '000'B)			
34		[NOT TSPC_EGSM]			
35		+CombinedBCCH_B(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
36		+SysInfoSending_fh_rpct(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '00000'B, '000000'B, '000'B)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_20_9

Group: GSM_MS_CS_4220/CellSelection/

Purpose: 1. To verify that if the MS calculates a received level average (over 5 seconds) for a non-serving suitable cell which results in the value of C2 exceeding the value of C2 for the serving cell, then cell reselection takes place to the non-serving cell.

2. To verify that by using suitable varying levels of signal strength for non serving cells, the MS samples on non serving cell BCCH carriers are as far as possible distributed uniformly over each evaluated period.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_2, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_62, BcchFreqLst_Omit, BcchFreqLst_64, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		+ltree_StartCarriers			
5		+PageOnCarriers("dummy", C_CellB, "dummy", "dummy", "dummy", "dummy")			
6		+MM_PwrOrSimOn(C_SIMIn)			
7		START T_dly(50000)			
8	CS_0057	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)	F	
9	CS_0058	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
10		?TIMEOUT T_dly			
11		(TCV_Null := OM_VaryRFOfCell(C_BCCH_B_1, 2700, 56, 2700, 33))			
12		START T_dly(25000)			
13	CS_0059	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)	F	
14	CS_0060	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
15		?TIMEOUT T_dly			
16		(TCV_Null := OM_ChangeRFOfCell(C_CellA, 37))			
17		START T_dly(20000)			
18	Label1	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
19		->Label1			
20	CS_0061	?TIMEOUT T_dly		F	
21		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
22	CS_0062	(TCV_Null := OM_StopVaryRFOfCell(C_BCCH_B_1))		P	
		ltree_StartCarriers			
23		+ltree_StartCarrier1			
24		+ltree_StartCarrier2			
		ltree_StartCarrier1			
25		(TCV_sacch := OC_SubchOfSacch4(			

		TSPX_SDCCH4SubDef, C_CellA))			
26		[TSPC_EGSM]			
27		+CombinedBCCH_A(53, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_2)			
28		+SysInfoSending_fh_rcc(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_2, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_63, BcchFreqLst_64, BcchFreqLst_55, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
29		[NOT TSPC_EGSM]			
30		+CombinedBCCH_A(53, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_2)			
31		+SysInfoSending_fh_rcc(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_2, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_62, BcchFreqLst_64, BcchFreqLst_55, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
		ltree_StartCarrier2			
32		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
33		[TSPC_EGSM]			
34		+CombinedBCCH_B(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_2)			
35		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_2, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_63, BcchFreqLst_64, BcchFreqLst_55, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
36		[NOT TSPC_EGSM]			
37		+CombinedBCCH_B(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_2)			
38		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_2, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_62, BcchFreqLst_64, BcchFreqLst_55, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_20_10
Group: GSM_MS_CS_4220/CellSelection/
Purpose: 1. To verify that by using suitable varying levels of signal strength for the serving cell, the MS performs a running average over 5 consecutive paging blocks.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_Omit, BcchFreqLst_51, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		+ltree_StartCarriers			
5		+PageOnCarriers("dummy", C_CellB, "dummy", "dummy", "dummy", "dummy")			
6		+MM_PwrOrSimOn(C_SIMIn)			
7		START T_dly(50000)			
8	Label 1	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
9		->Label1			
10		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
11		->Label1			
12		?TIMEOUT T_dly			
13		(TCV_Null := OM_VaryRFOfCell(C_BCCH_A_1, 3000, 33, 3000, 63))			
14		START T_dly(25000)			
15	Label 2	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
16		->Label2			
17	CS_0063	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
18		?TIMEOUT T_dly			
19		(TCV_Null := OM_ChangeRFOfCell(C_CellB, 57))			
20		START T_dly(20000)			
21	Label 3	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
22		->Label3			
23	CS_0064	?TIMEOUT T_dly		F	
24		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
25	CS_0065	(TCV_Null := OM_StopVaryRFOfCell(C_BCCH_A_1))		P	
		ltree_StartCarriers			
26		+ltree_StartCarrier1			
27		+ltree_StartCarrier2			
		ltree_StartCarrier1			
28		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
29		+CombinedBCCH_A(63, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot,			

		TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
30		+SysInfoSending_fh_rcc(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
		ltree_StartCarrier2			
31		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
32		[TSPC_EGSM]			
33		+CombinedBCCH_B(39, C_MaxPwrLvlG, C_MaxPwrLvlD, 985, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
34		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
35		[NOT TSPC_EGSM]			
36		+CombinedBCCH_B(39, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
37		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_20_12

Group: GSM_MS_CS_4220/CellSelection/

Purpose: 1. To verify that the list of the strongest 6 non-serving carriers shall be updated at least as often as the duration of the running average defined for measurements on the BCCH allocation and may be updated more frequently.

2. To verify that when the MS recognizes that a new BCCH carrier has become one of the 6 strongest, and the BCCH data shall be decoded for the new carrier within 30 seconds.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_Omit, BcchFreqLst_51, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		+ltree_StartCarriers			
5		+PageOnCarriers("dummy", C_CellB, "dummy", "dummy", "dummy", "dummy")			
6		+MM_PwrOrSimOn(C_SIMIn)			
7		START T_dly(50000)			
8	CS_0066	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)	F	
9	CS_0067	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
10		?TIMEOUT T_dly			
11		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '001010'B, '0'B, '0'B, '00000'B)			
12		START T_dly(345000)			
13	Label 1	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
14		->Label1			
15	CS_0068	?TIMEOUT T_dly		F	
16	CS_0069	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	P	
		ltree_StartCarriers			
17		+ltree_StartCarrier1			
18		+ltree_StartCarrier2			
		ltree_StartCarrier1			
19		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
20		+CombinedBCCH_A(38, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			

21		+SysInfoSending_fh_rcc(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
		ltree_StartCarrier2			
22		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
23		[TSPC_EGSM]			
24		+CombinedBCCH_B(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 985, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
25		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
26		[NOT TSPC_EGSM]			
27		+CombinedBCCH_B(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
28		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_20_13

Group: GSM_MS_CS_4220/CellSelection/

Purpose: 1. To verify that the MS will check the BSIC of the non-serving cell, which is in the list of 6 stringest neighbour cells, by changing the BSIC and the BCCH data of the non-serving cell such that the value of C2 for that cell exceeds the value of C2 of the serving cell, and observing that the MS performs cell reselection within the time allowed to check the BSIC, redetermine the BCCH data and perform cell reselection.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_Omit, BcchFreqLst_51, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		+ltree_StartCarriers			
5		+PageOnCarriers("dummy", C_CellB, "dummy", "dummy", "dummy", "dummy")			
6		+MM_PwrOrSimOn(C_SIMIn)			
7		START T_dly(50000)			
8	CS_0070	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)	F	
9	CS_0071	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
10		?TIMEOUT T_dly			
11		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC_3, C_NCC, C_NCCP_2, '001010'B, '0'B, '0'B, '00000'B)			
12		START T_dly(85000)			
13	Label 1	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
14		->Label1			
15	CS_0072	?TIMEOUT T_dly		F	
16	CS_0073	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	P	
		ltree_StartCarriers			
17		+ltree_StartCarrier1			
18		+ltree_StartCarrier2			
		ltree_StartCarrier1			
19		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
20		+CombinedBCCH_A(38, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
21		+SysInfoSending_fh_rcc(C_SCH_A,			

		C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
		ltree_StartCarrier2			
22		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
23		[TSPC_EGSM]			
24		+CombinedBCCH_B(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 985, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
25		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
26		[NOT TSPC_EGSM]			
27		+CombinedBCCH_B(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
28		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_20_15

Group: GSM_MS_CS_4220/CellSelection/

Purpose: 1. To verify that if an LU is rejected with cause "LA not allowed" that the LAI of that cell is written into a forbidden list which prevents the MS from performing LU onto another cell in that LA. This is verified indirectly in test purpose 2, 3 and 4.

2. To verify that the MS will not reject a cell for camping on because the cell is part of a LA in the list of "forbidden LAIs for regional provision of service". This is verified indirectly by making the MS attempt an emergency call and checking that the channel request message is transmitted on the correct cell.

3. To verify that the MS when receiving an LU reject with cause value "LA not allowed", the MS continues to perform cell-reselection. Cell reselection is triggered if there is a better cell in the same LA, or a much better cell in another LA of the selected PLMN (using the CRH parameter).

4. To verify that a new LU attempt will be performed when a new LA (or new PLMN) is entered.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys14, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_Omit, BcchFreqLst_51, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		+ltree_StartCarriers			
5		+MM_PwrOrSimOn(C_SIMIn)			
6		START T_dly(33000)			
7	Label 1	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
8		->Label1			
9		L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)		
10		->Label1			
11	CS_0074	?TIMEOUT T_dly		F	
12		L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
13		+MM_LupRej(C_rc_LAnotallowed, C_imsi_attach, TimingAdv(0))			
14		START T_dly(30000)			
15	Label 2	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
16		->Label2			
17		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
18		->Label2			
19		L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)		
20		->Label2			
21		?TIMEOUT T_dly			
22		+BasicServiceMO(C_EmgCall, TSPX_EmgCallRate)			
23		(TCV_Null := OO_InitCall(C_EmgCall))			
24		START T_dly(15000)			
25	Label 3	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
26		->Label3			

27		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
28		->Label3			
29	CS_0075	?TIMEOUT T_dly		F	
30		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq_01(C_RACH_C_1, ChRequest_02)		
31		[TSPC_EGSM]			
32		(TCV_chdescr_arfcn := 985)			
33		+ltree_continue			
34		[TSPC_PGSM]			
35		(TCV_chdescr_arfcn := 18)			
36		+ltree_continue			
37		[TSPC_DCS]			
38		(TCV_chdescr_arfcn := 543)			
39		+ltree_continue			
	ltree_continue				
40	L!DL_UdatRqImmass		ImmAss(C_AGCH_C_1, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
41	L?DL_EstInCmsRq		CMSerReq(CMServiceReq_01)		
42	(TCV_ch := OC_SubchOfSdcch4(TSPX_SDCCH4SubDef, C_CellC))				
43	L!DL_DatRqCmsRej		CMSerRej(TCV_ch, CMServiceRej_30(C_rc_networkfailure))		
44	+ChanRel(TCV_ch)				
45	START T_dly(10000)				
46	Label 4	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
47		->Label4			
48		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
49		->Label4			
50		L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)		
51		->Label4			
52		?TIMEOUT T_dly			
53		(TCV_Null := OM_ChangeRFOfCell(C_CellB, 65))			
54		START T_dly(20000)			
55	Label 5	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
56		->Label5			
57		L?DL_RacInChRq	ChReq_01(C_RACH_C_1, ChRequest_02)		
58		->Label5			
59	CS_0076	?TIMEOUT T_dly		F	
60		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq_01(C_RACH_B_1, ChRequest_02)		
61		[TSPC_PGSM OR TSPC_EGSM]			
62		(TCV_ch := OC_SubchOfSdcch4(TSPX_SD			

		CCH4SubDef, C_CellB), TCV_chdescr_arfcn := 9)			
63		+ltree_continue2			
64		[TSPC_DCS]			
65		(TCV_ch := OC_SubchOfSdcch4(TSPX_SD CCH4SubDef, C_CellB), TCV_chdescr_arfcn := 543)			
66		+ltree_continue2			
		ltree_continue2			
67		L!DL_UdatRqImmass	ImmAss(C_AGCH_B_1, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubD ef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, TimingAdv(0)))		
68	CS_0 077	L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_05)	P	
		ltree_StartCarriers			
69		+ltree_StartCarrier1			
70		+ltree_StartCarrier2			
71		+ltree_StartCarrier3			
72		[TSPC_PGSM OR TSPC_EGSM]			
73		(TCV_chdescr_arfcn:= 3)			
74		[TSPC_DCS]			
75		(TCV_chdescr_arfcn:= 512)			
		ltree_StartCarrier1			
76		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
77		+CombinedBCCH_A(63, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
78		+SysInfoSending_fh_rcc(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys14, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '110010'B, '0'B, '0'B, '00000'B)			
		ltree_StartCarrier2			
79		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
80		+CombinedBCCH_B(54, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
81		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_2222, CellOpt_01, CellChDes_24, CellChDes_24d, 0, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '100000'B, '0'B, '0'B, '00000'B)			
		ltree_StartCarrier3			
82		(TCV_sacch := OC_SubchOfSacch4(			

		TSPX_SDCCH4SubDef, C_CellC))			
83		[TSPC_EGSM]			
84		+CombinedBCCH_C(44, C_MaxPwrLvlG, C_MaxPwrLvlD, 985, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
85		+SysInfoSending_fh_rcc(C_SCH_C, C_BCCH_C_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellC, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys10, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '011010'B, '0'B, '0'B, '00000'B)			
86		[NOT TSPC_EGSM]			
87		+CombinedBCCH_C(44, C_MaxPwrLvlG, C_MaxPwrLvlD, 18, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
88		+SysInfoSending_fh_rcc(C_SCH_C, C_BCCH_C_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_1, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellC, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys10, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '011010'B, '0'B, '0'B, '00000'B)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_20_16

Group: GSM_MS_CS_4220/CellSelection/

Purpose: 1. To verify that the MS initializes the DSC counter in accordance with the conformance requirement. This is verified indirectly.

2. To verify that whenever the MS successfully decodes a message on paging subchannel , the DSC is increased by 1, (however never beyond the nearest integer to 90/N). This is verified indirectly.

3. To verify that whenever the MS can not successfully decode a message on paging subchannel, the DSC is decreased by 4. This is verified indirectly.

4. To verify that when the DSC reaches 0, a downlink signalling failure shall be declared and the MS will perform cell reselection.

Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_Omit, BcchFreqLst_51, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		+ltree_StartCarriers			
5		+MM_PwrOrSimOn(C_SIMIn)			
6		+PageOnCarriers("dummy", C_CellB, "dummy", "dummy", "dummy", "dummy")			
7		START T_dly(40000)			
8	Label 1	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
9		->Label1			
10	CS_0078	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
11		?TIMEOUT T_dly			
12		+CCCH_group_Paging_group(TCV_Ccd0A, TSPX_IMSI)			
13		+SelectPagingCh(C_CellA)			
14		(TCV_Null := OM_CorruptPgBlks(TCV_PgCh, TCV_Pgg, 4))			
15		START T_dly(30000)			
16	Label 2	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
17		->Label2			
18	CS_0079	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	F	
19		?TIMEOUT T_dly			
20		+CCCH_group_Paging_group(TCV_Ccd0A, TSPX_IMSI)			
21		+SelectPagingCh(C_CellA)			
22		(TCV_Null := OM_CorruptPgBlks(TCV_PgCh, TCV_Pgg, 5))			
23		+ltree_continue			
		ltree_continue			
24		START T_dly(15000)			
25	Label	L?DL_RacInChRq	ChReq_01(C_RACH_		

	3		A_1, ChRequest_02)		
26		->Label3			
27	CS_0080	?TIMEOUT T_dly		F	
28		L?DL_RacInChRq READTIMER T_dly(TCV_Time), CANCEL T_dly	ChReq_01(C_RACH_B_1, ChRequest_02)		
29		START T_dly1(30000 - TCV_Time)			
30	Label4	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
31		->Label4			
32		L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)		
33		->Label4			
34	CS_0081	?TIMEOUT T_dly1		P	
		ltree_StartCarriers			
35		+ltree_StartCarrier1			
36		+ltree_StartCarrier2			
		ltree_StartCarrier1			
37		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
38		+CombinedBCCH_A(43, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
39		+SysInfoSending_fh_rcc(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
40		(TCV_Ccd0A := TCV_Ccd0H)			
		ltree_StartCarrier2			
41		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
42		[TSPC_EGSM]			
43		+CombinedBCCH_B(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 985, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
44		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
45		(TCV_Ccd0B := TCV_Ccd0H)			
46		[NOT TSPC_EGSM]			
47		+CombinedBCCH_B(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 9, 543, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
48		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24,			

		CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
49		(TCV_Ccd0B := TCV_Ccd0H)			
Detailed Comments:					

Test Case Dynamic Behaviour

Test Case Name: TC_20_17
Group: GSM_MS_CS_4220/CellSelection/
Purpose: If no suitable cell is found withiin cell reselection during a 10 second period, then the cell selection algorithm of GSM 03.22 shall be performed, GSM 05.08; 6.6.2.
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_guard(600)			
2		+IdleUpdated(C_E_default, C_CellA, C_SCH_A, C_BCCH_A_1, C_Immass, C_S0, C_SlotNU, C_SlotNU, C_SlotNU, C_BCC, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_Omit, BcchFreqLst_51, BcchFreqLst_Omit, C_Restablishment, C_BCC, C_NCC, C_Start_Tmsi, TSPX_SDCCH4SubDef, TSPX_CKSNDef, TSPX_RANDDef, C_arfcnA, C_arfcnAd, TSPX_IMSI, C_Test_fh, C_NCCP_2)			
3	body	+MM_PwrOrSimOff(C_SIMIn)			
4		+ltree_StartCarrier1			
5		+MM_PwrOrSimOn(C_SIMIn)			
6		(TCV_Null := OO_PressKeyWhenInService())			
7		(TCV_Null := OM_ChangeRFOfCell(C_CellA, 13))			
8		+ltree_StartCarrier2			
9		+PageOnCarriers("dummy", C_CellB, "dummy", "dummy", "dummy", "dummy")			
10		START T_dly(60000)			
11	Label 1	L?DL_RacInChRq	ChReq_01(C_RACH_A_1, ChRequest_02)		
12		->Label1			
13	CS_0082	?TIMEOUT T_dly		F	
14	CS_0083	L?DL_RacInChRq	ChReq_01(C_RACH_B_1, ChRequest_02)	P	
		ltree_StartCarrier1			
15		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellA))			
16		+CombinedBCCH_A(43, C_MaxPwrLvlG, C_MaxPwrLvlD, 3, 512, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			
17		+SysInfoSending_fh_rcc(C_SCH_A, C_BCCH_A_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellA, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
		ltree_StartCarrier2			
18		(TCV_sacch := OC_SubchOfSacch4(TSPX_SDCCH4SubDef, C_CellB))			
19		[TSPC_EGSM]			
20		+CombinedBCCH_B(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 1000, 520, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			uses AFRCN 1000 (not in BcchFreqLst_52 - NCD)
21		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2,			

		C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
22		[NOT TSPC_EGSM]			
23		+CombinedBCCH_B(33, C_MaxPwrLvlG, C_MaxPwrLvlD, 10, 520, C_Immass, TCV_slot, TCV_tsc, TimingAdv(0), C_BABR_0, C_cch_1Comb, C_BPM_3)			uses AFRCN 10 (not in BcchFreqLst_50 - NCD)
24		+SysInfoSending_fh_rcc(C_SCH_B, C_BCCH_B_1, C_TxInt_5, C_Max_2, C_NECI_0, C_ATT_0, C_BABR_0, C_cch_1Comb, C_BPM_3, C_T3212_0, C_ci_cellB, C_MCC_1, C_PLMN_1, C_LAC_1111, CellOpt_01, CellChDes_24, CellChDes_24d, C_CellReselectHys4, C_MaxPwrLvlG, C_MaxPwrLvlD, BcchFreqLst_50, BcchFreqLst_51, BcchFreqLst_52, C_Restablishment, C_BCC, C_NCC, C_NCCP_2, '010100'B, '0'B, '0'B, '00000'B)			
Detailed Comments:					

Test Step Library

Test Step Group management

Test Step Dynamic Behaviour					
Test Step Name:		StopAllBCCH			
Group:		GSM_MS_CS_4220/management/			
Objective:		To stop the RF transmission of all BCCH channels in all active cells.			
Default:		OtherEventsFail			
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			
2		(TCV_Null := OM_StopAllBCCH(C_BCCH_A_1, C_BCCH_B_1, C_BCCH_C_1, C_BCCH_D_1, C_BCCH_E_1, C_BCCH_F_1, C_BCCH_G_1, C_BCCH_H_1))			
3		[TSPC_DCS]			
4		(TCV_Null := OM_StopAllBCCH(C_BCCH_A_1, C_BCCH_B_1, C_BCCH_C_1, C_BCCH_D_1, C_BCCH_E_1, C_BCCH_F_1, C_BCCH_G_1, "dummy"))			
Detailed Comments:					

Test Step Group ChConfig

Test Step Dynamic Behaviour					
Test Step Name:	Config_CBMS_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH, SDCCH4 and CBCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, C_FCCH_A, C_SCH_A, C_BCCH_A_1, C_PCH_A_1, C_AGCH_A_1, C_RACH_A_1, C_SDCCH4_A, C_SACCHC4_A, C_CBCH_A))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_A, C_SDCCH40_A, C_SDCCH41_A, C_SDCCH42_A, C_SDCCH43_A, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_A, C_SACCHC40_A, C_SACCHC41_A, C_SACCHC42_A, C_SACCHC43_A, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, C_FCCH_A, C_SCH_A, C_BCCH_A_1, C_PCH_A_1, C_AGCH_A_1, C_RACH_A_1, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_A_2(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map the second BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", C_BCCH_A_2, C_PCH_A_2, C_AGCH_A_2, C_RACH_A_2, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_A_3(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map the third BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", C_BCCH_A_3, C_PCH_A_3, C_AGCH_A_3, C_RACH_A_3, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_A_4(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map the fourth BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", C_BCCH_A_4, C_PCH_A_4, C_AGCH_A_4, C_RACH_A_4, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_BCCH_CCCH_B_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH and CCCH onto the physical channel which represents cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, C_FCCH_B, C_SCH_B, C_BCCH_B_1, C_PCH_B_1, C_AGCH_B_1, C_RACH_B_1, "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, C_FCCH_A, C_SCH_A, C_BCCH_A_1, C_PCH_A_1, C_AGCH_A_1, C_RACH_A_1, C_SDCCH4_A, C_SACCHC4_A, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_A, C_SDCCH40_A, C_SDCCH41_A, C_SDCCH42_A, C_SDCCH43_A, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_A, C_SACCHC40_A, C_SACCHC41_A, C_SACCHC42_A, C_SACCHC43_A, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_A_1_fn(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3; fn:INTEGER)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, fn, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, C_FCCH_A, C_SCH_A, C_BCCH_A_1, C_PCH_A_1, C_AGCH_A_1, C_RACH_A_1, C_SDCCH4_A, C_SACCHC4_A, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_A, C_SDCCH40_A, C_SDCCH41_A, C_SDCCH42_A, C_SDCCH43_A, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_A, C_SACCHC40_A, C_SACCHC41_A, C_SACCHC42_A, C_SACCHC43_A, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_B_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, C_FCCH_B, C_SCH_B, C_BCCH_B_1, C_PCH_B_1, C_AGCH_B_1, C_RACH_B_1, C_SDCCH4_B, C_SACCHC4_B, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_B, C_SDCCH40_B, C_SDCCH41_B, C_SDCCH42_B, C_SDCCH43_B, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_B, C_SACCHC40_B, C_SACCHC41_B, C_SACCHC42_B, C_SACCHC43_B, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_B_1_fn(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3; fn:INTEGER)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, fn, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, C_FCCH_B, C_SCH_B, C_BCCH_B_1, C_PCH_B_1, C_AGCH_B_1, C_RACH_B_1, C_SDCCH4_B, C_SACCHC4_B, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_B, C_SDCCH40_B, C_SDCCH41_B, C_SDCCH42_B, C_SDCCH43_B, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_B, C_SACCHC40_B, C_SACCHC41_B, C_SACCHC42_B, C_SACCHC43_B, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_C_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell C.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellC, C_FCCH_C, C_SCH_C, C_BCCH_C_1, C_PCH_C_1, C_AGCH_C_1, C_RACH_C_1, C_SDCCH4_C, C_SACCHC4_C, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_C, C_SDCCH40_C, C_SDCCH41_C, C_SDCCH42_C, C_SDCCH43_C, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_C, C_SACCHC40_C, C_SACCHC41_C, C_SACCHC42_C, C_SACCHC43_C, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_D_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell D				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellD, C_FCCH_D, C_SCH_D, C_BCCH_D_1, C_PCH_D_1, C_AGCH_D_1, C_RACH_D_1, C_SDCCH4_D, C_SACCHC4_D, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_D, C_SDCCH40_D, C_SDCCH41_D, C_SDCCH42_D, C_SDCCH43_D, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_D, C_SACCHC40_D, C_SACCHC41_D, C_SACCHC42_D, C_SACCHC43_D, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH4_E_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell E				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellE, C_FCCH_E, C_SCH_E, C_BCCH_E_1, C_PCH_E_1, C_AGCH_E_1, C_RACH_E_1, C_SDCCH4_E, C_SACCHC4_E, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH4_E, C_SDCCH40_E, C_SDCCH41_E, C_SDCCH42_E, C_SDCCH43_E, "dummy", "dummy", "dummy", "dummy", C_SACCHC4_E, C_SACCHC40_E, C_SACCHC41_E, C_SACCHC42_E, C_SACCHC43_E, "dummy", "dummy", "dummy", "dummy"))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH8_A_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set one physical channel used as SDCCH8 channel for instance 1 of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_SDCCH8_A_1, C_SACCHC8_A_1, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH8_A_1, C_SDCCH80_A_1, C_SDCCH81_A_1, C_SDCCH82_A_1, C_SDCCH83_A_1, C_SDCCH84_A_1, C_SDCCH85_A_1, C_SDCCH86_A_1, C_SDCCH87_A_1, C_SACCHC8_A_1, C_SACCHC80_A_1, C_SACCHC81_A_1, C_SACCHC82_A_1, C_SACCHC83_A_1, C_SACCHC84_A_1, C_SACCHC85_A_1, C_SACCHC86_A_1, C_SACCHC87_A_1))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH8_A_2(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set one physical channel used as SDCCH8 channel for instance 2 of cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellA, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_SDCCH8_A_2, C_SACCHC8_A_2, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH8_A_2, C_SDCCH80_A_2, C_SDCCH81_A_2, C_SDCCH82_A_2, C_SDCCH83_A_2, C_SDCCH84_A_2, C_SDCCH85_A_2, C_SDCCH86_A_2, C_SDCCH87_A_2, C_SACCHC8_A_2, C_SACCHC80_A_2, C_SACCHC81_A_2, C_SACCHC82_A_2, C_SACCHC83_A_2, C_SACCHC84_A_2, C_SACCHC85_A_2, C_SACCHC86_A_2, C_SACCHC87_A_2))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	Config_SDCCH8_B_1(par_bspwr, par_mspwr:INTEGER; chmod:CHMOD; acttype:BITSTRING; slot:SN; tsc:TSC; rf:FRQPARA; chcmbn:LOGCH; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set one physical channel used as SDCCH8 channel for instance 1 of cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OM_ChConf(par_bspwr, par_mspwr, acttype, chmod, ta, slot, tsc, rf, chcmbn, 0, 0, babr, cch_con, bpm, PgReqTp1Norm, '0'B, '0'B, C_CellB, "dummy", "dummy", "dummy", "dummy", "dummy", "dummy", C_SDCCH8_B_1, C_SACCHC8_B_1, "dummy"))			
2		(TCV_Null := OM_Assoc(C_SDCCH8_B_1, C_SDCCH80_B_1, C_SDCCH81_B_1, C_SDCCH82_B_1, C_SDCCH83_B_1, C_SDCCH84_B_1, C_SDCCH85_B_1, C_SDCCH86_B_1, C_SDCCH87_B_1, C_SACCHC8_B_1, C_SACCHC80_B_1, C_SACCHC81_B_1, C_SACCHC82_B_1, C_SACCHC83_B_1, C_SACCHC84_B_1, C_SACCHC85_B_1, C_SACCHC86_B_1, C_SACCHC87_B_1))			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: CombinedBCCH_A_CBMS(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_CS_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.

Default: OtherEvents

Comments: time slot = 0, ARFCN = 20 (GSM900) or ARFCN = 590 (DCS1800) cell A for RR testing.

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_CBMS_A_1(par_bspwr,mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_CBMS_A_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_CBCH_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			

Detailed Comments:
1. For P-GSM900.
2. For DCS1800.

Test Step Dynamic Behaviour

Test Step Name: CombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_CS_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_A_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_A_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			

Detailed Comments:
1. For P-GSM900.
2. For DCS1800.

Test Step Dynamic Behaviour

Test Step Name: CombinedBCCH_A_fn(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3; fn:INTEGER)

Group: GSM_MS_CS_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_A_1_fn(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm, fn)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_A_1_fn(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm, fn)			

Detailed Comments:
1. For P-GSM900.
2. For DCS1800.

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_B(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_B_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_B_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_B_fn(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3; fn:INTEGER)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell B.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_B_1_fn(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm, fn)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_B_1_fn(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm, fn)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_C(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell C.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_C_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_C_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_D(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell D.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_D_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_D_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	CombinedBCCH_E(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map FCCH, SCH, BCCH, CCCH and SDCCH4 onto the physical channel which represents cell E.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_SDCCH4_E_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_SDCCH4_E_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH_SDCCH4_SACCHC4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set one physical channel used as FCHH_SCH_BCCH_CCCH for cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_BCCH_CCCH_A_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_A_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	NonCombinedBCCH_A_2(acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map the second BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_BCCH_CCCH_A_2(63, 19, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnA), C_BCCH_CCCH_2, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_A_2(63, 15, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnAd), C_BCCH_CCCH_2, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	NonCombinedBCCH_A_3(acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map the third BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Comments:	time slot = 4, ARFCN = TSPX_BCCHcarrierA				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_BCCH_CCCH_A_3(63, 19, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnA), C_BCCH_CCCH_3, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_A_3(63, 15, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnAd), C_BCCH_CCCH_3, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour					
Test Step Name:	NonCombinedBCCH_A_4(acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)				
Group:	GSM_MS_CS_4220/management/ChConfig/				
Objective:	To set a physical channel and map the fourth BCCH, CCCH onto the physical channel which represents cell A.				
Default:	OtherEvents				
Comments:	time slot = 6, ARFCN = TSPX_BCCHcarrierA				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[TSPC_PGSM OR TSPC_EGSM]			1.
2		+Config_BCCH_CCCH_A_4(63, 19, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnA), C_BCCH_CCCH_4, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_A_4(63, 15, ChMod_sign, acttype, slot, tsc, FreqBCCH(C_arfcnAd), C_BCCH_CCCH_4, ta, babr, cch_con, bpm)			
Detailed Comments:	1. For P-GSM900. 2. For DCS1800.				

Test Step Dynamic Behaviour

Test Step Name: NonCombinedBCCH_B(par_bspwr, mtmcgsm, mtmcdcs, arfcn1, arfcn2:INTEGER; acttype:BITSTRING; slot:SN; tsc:TSC; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_CS_4220/management/ChConfig/

Objective: To set a physical channel and map FCCH, SCH, BCCH, CCCH onto the physical channel which represents cell B for RR testing.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[TSPC_PGSM]			1.
2		+Config_BCCH_CCCH_B_1(par_bspwr, mtmcgsm, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn1), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm)			
3		[TSPC_DCS]			2.
4		+Config_BCCH_CCCH_B_1(par_bspwr, mtmcdcs, ChMod_sign, acttype, slot, tsc, FreqBCCH(arfcn2), C_FCCH_SCH_BCCH_CCCH, ta, babr, cch_con, bpm)			

Detailed Comments:
1. For P-GSM900.
1.1 Frequenz and cell_id for cell B in HO cases
2. For DCS1800.
2.1 Frequenz and cell_id for cell B in HO cases

Test Step Dynamic Behaviour

Test Step Name: SDCCH8_A_1_nociph(bstring:B_3; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_CS_4220/management/ChConfig/

Objective: To set one physical channel used as SDCCH8 channel for instance 1 of cell A.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_ch := OC_SubchOfSdcch8(bstring, C_CellA, 1), TCV_sacch8 := OC_SubchOfSacch8(bstring, C_CellA, 1))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_SDCCH8_A_1(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_SDCCH8_SACCHC8_1, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacch8, TCV_sysinfo5, TCV_sysinfo6)			
5		[TSPC_DCS]			
6		+Config_SDCCH8_A_1(63, 19, chmod, acttypeT, slotT, tscT, Freqd, C_SDCCH8_SACCHC8_1, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacch8, TCV_sysinfo5, TCV_sysinfo6)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SDCCH8_A_2_nociph(bstring:B_3; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_CS_4220/management/ChConfig/

Objective: To set one physical channel used as hopping SDCCH8 channel for instance 2 of cell A, for TC_26_6_13_1.

Default: OtherEvents

Comments: hopping parameters are from PIXIT, used as before time channel.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_chTch := OC_SubchOfSdcch8(bstring, C_CellA, 2), TCV_sacchTch := OC_SubchOfSacch8(bstring, C_CellA, 2))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_SDCCH8_A_2(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_SDCCH8_SACCHC8_2, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			
5		[TSPC_DCS]			
6		+Config_SDCCH8_A_2(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_SDCCH8_SACCHC8_2, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5, TCV_sysinfo6)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SDCCH8_B_1_nociph(bstring:B_3; acttypeT:BITSTRING; slotT:SN; tscT:TSC; chmod:CHMOD; Freqg,Freqd:FRQPARA; ta:TA; babr, cch_con, bpm:B_3)

Group: GSM_MS_CS_4220/management/ChConfig/

Objective: To set one physical channel used as hopping SDCCH8 channel for instance 1 of cell B For TC_26_6_13_5.

Default: OtherEvents

Comments: hopping parameters are from PIXIT, used by handover command after time channel.

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_chTch := OC_SubchOfSdcch8(bstring, C_CellB, 1), TCV_sacchTch := OC_SubchOfSacch8(bstring, C_CellB, 1))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		+Config_SDCCH8_B_1(63, 19, chmod, acttypeT, slotT, tscT, Freqg, C_SDCCH8_SACCHC8_1, ta, babr, cch_con, bpm)			
4		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5_B, TCV_sysinfo6_B)			
5		[TSPC_DCS]			
6		+Config_SDCCH8_B_1(63, 15, chmod, acttypeT, slotT, tscT, Freqd, C_SDCCH8_SACCHC8_1, ta, babr, cch_con, bpm)			
7		+SysInfo_SacchSending(TCV_sacchTch, TCV_sysinfo5_B, TCV_sysinfo6_B)			

Detailed Comments:

Test Step Group Miscellaneous

Test Step Dynamic Behaviour					
Test Step Name:	Authentication(ch: LOGICCH; cksn: BITSTRING; rand:RAND)				
Group:	GSM_MS_CS_4220/Miscellaneous/				
Objective:					
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqAuthRq	AuthReq(ch, AuthRequest(cksn, rand))		
2		L?DL_DatInAuthRes (TCV_Sres := DL_DatInAuthRes.msg.sres)	AuthRes(AuthResponse)		
3		(TCV_Res := OC_ChkSRES(TCV_Sres, TSPX_Ki, rand))			
4	CS_0084	[NOT TCV_Res]		(F)	1)
5	CS_0085	[TCV_Res]		(P)	
Detailed Comments:	Authentication fails.				

Test Step Dynamic Behaviour

Test Step Name: CCCH_group_Paging_group(ccd: CCD; imsi: HEXSTRING)

Group: GSM_MS_CS_4220/Miscellaneous/

Objective: To calculate the CCCH group and the Paging group from the IMSI and the Control Channel Descriptor according to GSM 05.02, subclause 6.5.2.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[ccd.ccch_con = '000'B]			
2		(TCV_bs_cc_chans := 1)			
3		+localtree1			
4		[ccd.ccch_con = '001'B]			
5		(TCV_bs_cc_chans := 1)			
6		+localtree1			
7		[ccd.ccch_con = '010'B]			
8		(TCV_bs_cc_chans := 2)			
9		+localtree1			
10		[ccd.ccch_con = '100'B]			
11		(TCV_bs_cc_chans := 3)			
12		+localtree1			
13		[ccd.ccch_con = '110'B]			
14		(TCV_bs_cc_chans := 4)			
15		+localtree1			
		localtree1			
16		(TCV_bs_ag_blks_res := BIT_TO_INT(ccd.babr), TCV_bs_pa_mfrms := (2 + BIT_TO_INT(ccd.bpm)))			
17		[ccd.ccch_con = '001'B]			i.e. combined
18	CS_0086	[(TCV_bs_ag_blks_res < 0) OR(TCV_bs_ag_blks_res > 2)]		I	Stop! A tester error.
19		[(TCV_bs_ag_blks_res >= 0) AND (TCV_bs_ag_blks_res <= 2)]			
20		(TCV_tmp := 3 - TCV_bs_ag_blks_res)			
21		+localtree2			
22		[ccd.ccch_con <> '001'B]			i.e. not combined
23	CS_0087	[(TCV_bs_ag_blks_res < 0) OR(TCV_bs_ag_blks_res > 7)]		I	Stop! A tester error.
24		[(TCV_bs_ag_blks_res >= 0) AND (TCV_bs_ag_blks_res <= 7)]			
25		(TCV_tmp := 9 - TCV_bs_ag_blks_res)			
26		+localtree2			ini. TCV Pgg
		localtree2			
27		(TCV_tmp := TCV_tmp * TCV_bs_pa_mfrms)			
28		(TCV_Pgg := INT_TO_BIT((((OC_BCDtoInt(imsi, 3)) MOD(TCV_bs_cc_chans * TCV_tmp)) MOD TCV_tmp), 8), TCV_Ccchg := (((OC_BCDtoInt(imsi, 3)) MOD(TCV_bs_cc_chans * TCV_tmp)) / TCV_tmp))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: ImsiAttach(par_mi:MI; ta:TA; mcc, mnc:OCTETSTRING; sim_rmvd:BOOLEAN)
Group: GSM_MS_CS_4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv) AND(TSPC_DetachOnSIMRmv)]			SIM has been removed during the test.
2		(TCV_Null := OO_SIMIns())			
3		+ltree_imsiattach			
4		[TSPC_SwitchOnOff]			
5		(TCV_Null := OO_SwitchOn())			
6		+ltree_imsiattach			
7		[TSPC_DetachOnPwrDn]			
8		(TCV_Null := OO_PowerUp())			Restore power source
9		+ltree_imsiattach			
10		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv) OR(NOT TSPC_DetachOnSIMRmv)) AND(NOT TSPC_SwitchOnOff) AND(NOT TSPC_DetachOnPwrDn))]			
11		(TCV_Null := OO_PowerUp())			Restore power source
		ltree_imsiattach			
12		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_09)		
13		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
14		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
15	CS_0088	L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_31(par_mi, mcc, mnc, TCV_lac, C_imsi_attach, TCV_cksn))	(P)	
16		ACTIVATE(OtherEventsFail)			Restore Normal default
17		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp(mcc, mnc, TCV_lac))		No MI
18		+ChanRel(TCV_ch)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: ImsiAttachIni(par_mi:MI; ta:TA; mcc, mnc:OCTETSTRING; sim_rmvd:BOOLEAN)
Group: GSM_MS_CS_4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv) AND(TSPC_DetachOnSIMRmv)]			SIM has been removed during the test.
2		(TCV_Null := OO_SIMIns())			
3		+ltree_imsiattachIni			
4		[TSPC_SwitchOnOff]			
5		(TCV_Null := OO_SwitchOn())			
6		+ltree_imsiattachIni			
7		[TSPC_DetachOnPwrDn]			
8		(TCV_Null := OO_PowerUp())			Restore power source
9		+ltree_imsiattachIni			
10		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv) OR(NOT TSPC_DetachOnSIMRmv)) AND(NOT TSPC_SwitchOnOff) AND(NOT TSPC_DetachOnPwrDn))]			
11		(TCV_Null := OO_PowerUp())			Restore power source
		ltree_imsiattachIni			
12		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) START T_dly(15000)	ChReq(ChRequest_09)		
13		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
14		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
15	CS_0089	L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_31(par_mi, mcc, mnc, TCV_lac, C_imsi_attach, TCV_cksn))	(P)	
16		ACTIVATE(OtherEventsFail)			Restore Normal default

Detailed Comments: The test step is calld in TC_26_7_5_8_1, TC_26_7_5_8_2, TC_26_7_5_8_3.

Test Step Dynamic Behaviour

Test Step Name: ImsiAttachNoReaction(par_int: INTEGER; sim_rmvd:BOOLEAN)
Group: GSM_MS_CS_4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv) AND(TSPC_DetachOnSIMRmv)]			SIM has been removed during the test.
2		(TCV_Null := OO_SIMIns())			
3		+NoReaction(par_int)			
4		[TSPC_SwitchOnOff]			
5		(TCV_Null := OO_SwitchOn())			
6		+NoReaction(par_int)			
7		[TSPC_DetachOnPwrDn]			
8		(TCV_Null := OO_PowerUp())			Restore power source
9		+NoReaction(par_int)			
10		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv) OR((TSPC_SIMRmv) AND(NOT TSPC_DetachOnSIMRmv))) AND(NOT TSPC_SwitchOnOff) AND(NOT TSPC_DetachOnPwrDn))]			
11		(TCV_Null := OO_PowerUp())			Restore power source

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: ImsiDetach(par_mi:MI; ta:TA; sim_rmvd:BOOLEAN)
Group: GSM_MS_CS_4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv) AND(TSPC_DetachOnSIMRmv)]			SIM needs to be removed.
2		(TCV_Null :=OO_SIMRmv())			
3		+ltree_imsidetach			
4		[TSPC_SwitchOnOff]			
5		(TCV_Null := OO_SwitchOff())			
6		+ltree_imsidetach			
7		[TSPC_DetachOnPwrDn]			
8		(TCV_Null := OO_PowerDown())			Remove power source
9		+ltree_imsidetach			
10		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv) OR(NOT TSPC_DetachOnSIMRmv)) AND(NOT TSPC_SwitchOnOff) AND(NOT TSPC_DetachOnPwrDn))]			
11		(TCV_Null := OO_PowerDown())			Remove power source
		ltree_imsidetach			
12		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_02)		
13		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
14		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
15	CS_0090	L?DL_EstInImsidIn	ImsiDet(ImsiDetach_30(par_mi))	(P)	
16		ACTIVATE(OtherEventsFail)			Restore normal default
17		+ChanRel(TCV_ch)			

Detailed Comments: If IMSI detach procedure has been executed, the channel will be relesed. If ATT=0 no detach has been executed, the channel is still connected.

Test Step Dynamic Behaviour

Test Step Name: ImsiDetachNoReaction(par_int: INTEGER; sim_rmvd:BOOLEAN)
Group: GSM_MS_CS_4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		[(sim_rmvd) AND(TSPC_SIMRmv) AND(TSPC_DetachOnSIMRmv)]			SIM needs to be removed.
2		(TCV_Null :=OO_SIMRmv())			
3		+NoReaction(par_int)			
4		[TSPC_SwitchOnOff]			
5		(TCV_Null := OO_SwitchOff())			
6		+NoReaction(par_int)			
7		[TSPC_DetachOnPwrDn]			
8		(TCV_Null := OO_PowerDown())			Remove power source
9		+NoReaction(par_int)			
10		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv) OR((TSPC_SIMRmv) AND(NOT TSPC_DetachOnSIMRmv))) AND(NOT TSPC_SwitchOnOff) AND(NOT TSPC_DetachOnPwrDn))]			
11		(TCV_Null := OO_PowerDown())			Remove power source

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LupInit2(locup:B_2; ta:TA)
Group: GSM_MS_CS_4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_09)		
2		ACTIVATE(OtherEventsFail_02)			To match ChReq retrans.
3		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
4		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(locup))		
5		ACTIVATE(OtherEventsFail)			Restore Normal default

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: MM_LupRej(par_rej: REJCAU; locup:B_2; ta:TA)
Group: GSM_MS_CS_4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		+MM_LupInit2(locup, ta)			
2	CS_0091	L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(par_rej))	(P)	
3		+ChanRel(TCV_ch)			

Detailed Comments:

Test Step Dynamic Behaviour					
Test Step Name:	MM_PwrOrSimOff(sim_rmvd:BOOLEAN)				
Group:	GSM_MS_CS_4220/Miscellaneous/				
Objective:					
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[(sim_rmvd) AND(TSPC_SIMRmv)]			SIM needs to be removed.
2		(TCV_Null :=OO_SIMRmv())			
3		[TSPC_SwitchOnOff]			
4		(TCV_Null := OO_SwitchOff())			
5		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv)) AND(NOT TSPC_SwitchOnOff))]			
6		(TCV_Null := OO_PowerDown())			Remove power source
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	MM_PwrOrSimOn(sim_rmvd:BOOLEAN)				
Group:	GSM_MS_CS_4220/Miscellaneous/				
Objective:					
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[(sim_rmvd) AND(TSPC_SIMRmv)]			SIM has been removed
2		(TCV_Null := OO_SIMIns())			
3		[TSPC_SwitchOnOff]			
4		(TCV_Null := OO_SwitchOn())			
5		[(((NOT sim_rmvd) OR(NOT TSPC_SIMRmv)) AND(NOT TSPC_SwitchOnOff))]			
6		(TCV_Null := OO_PowerUp())			Restore power source
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	NoReaction(par_int:INTEGER)				
Group:	GSM_MS_CS_4220/Miscellaneous/				
Objective:					
Default:	OtherEventsFail				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		START T_dly(par_int)			
2	CS_0092	?TIMEOUT T_dly		(P)	
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: PageOnCarriers(CellA_Id, CellB_Id, CellC_Id, CellD_Id, CellE_Id, CellF_Id:CellID)
Group: GSM_MS_CS_4220/Miscellaneous/
Objective:
Default:

Nr	Label	Behaviour Description	CRef	V	Comments
1		+ltree_A			
2		+ltree_B			
3		+ltree_C			
4		+ltree_D			
5		+ltree_E			
6		+ltree_F			
		ltree_A			
7		[CellA_Id = C_CellA]			
8		(TCV_Null := OM_PgFill(C_CellA, PgReqTp1_01))			Start continuous paging on Cell A
9		[CellA_Id <> C_CellA]			
		ltree_B			
10		[CellB_Id = C_CellB]			
11		(TCV_Null := OM_PgFill(C_CellB, PgReqTp1_01))			Start continuous paging on Cell B
12		[CellB_Id <> C_CellB]			
		ltree_C			
13		[CellC_Id = C_CellC]			
14		(TCV_Null := OM_PgFill(C_CellC, PgReqTp1_01))			Start continuous paging on Cell C
15		[CellC_Id <> C_CellC]			
		ltree_D			
16		[CellD_Id = C_CellD]			
17		(TCV_Null := OM_PgFill(C_CellD, PgReqTp1_01))			Start continuous paging on Cell D
18		[CellD_Id <> C_CellD]			
		ltree_E			
19		[CellE_Id = C_CellE]			
20		(TCV_Null := OM_PgFill(C_CellE, PgReqTp1_01))			Start continuous paging on Cell E
21		[CellE_Id <> C_CellE]			
		ltree_F			
22		[CellF_Id = C_CellF]			
23		(TCV_Null := OM_PgFill(C_CellF, PgReqTp1_01))			Start continuous paging on Cell F
24		[CellF_Id <> C_CellF]			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Pause(Millisecs:INTEGER)
Group: GSM_MS_CS_4220/Miscellaneous/
Objective:
Default:
Comments: Will wait for specified delay period. Any messages received during this period will be queued and processed later.

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_pause(Millisecs)			
2		?TIMEOUT T_pause			
3		?TIMEOUT T_guard			
4	CS_0093	(TCV_Null := OO_TguardTimeOut())		R	

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SelectPagingCh(cell:CellID)
Group: GSM_MS_CS_4220/Miscellaneous/
Objective: To assign a paging channel and an access grant channel to the variable TCV_PgCh and TCV_agch respectively, depending on the parameter 'cell' (cell ID)
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		[cell = C_CellA]			
2		[TCV_Ccchg=0]			
3		(TCV_PgCh := C_PCH_A_1, TCV_agch := C_AGCH_A_1, TCV_RaCh := C_RACH_A_1)			
4		[TCV_Ccchg=1]			
5		(TCV_PgCh := C_PCH_A_2, TCV_agch := C_AGCH_A_2, TCV_RaCh := C_RACH_A_2)			
6		[TCV_Ccchg=2]			
7		(TCV_PgCh := C_PCH_A_3, TCV_agch := C_AGCH_A_3, TCV_RaCh := C_RACH_A_3)			
8		[TCV_Ccchg=3]			
9		(TCV_PgCh := C_PCH_A_4, TCV_agch := C_AGCH_A_4, TCV_RaCh := C_RACH_A_4)			
10		[cell = C_CellB]			
11		[TCV_Ccchg=0]			
12		(TCV_PgCh := C_PCH_B_1, TCV_agch := C_AGCH_B_1, TCV_RaCh := C_RACH_B_1)			
13		[TCV_Ccchg=1]			
14		(TCV_PgCh := C_PCH_B_2, TCV_agch := C_AGCH_B_2, TCV_RaCh := C_RACH_B_2)			
15		[TCV_Ccchg=2]			
16		(TCV_PgCh := C_PCH_B_3, TCV_agch := C_AGCH_B_3, TCV_RaCh := C_RACH_B_3)			
17		[TCV_Ccchg=3]			
18		(TCV_PgCh := C_PCH_B_4, TCV_agch := C_AGCH_B_4, TCV_RaCh := C_RACH_B_4)			
19		[cell = C_CellC]			
20		[TCV_Ccchg=0]			
21		(TCV_PgCh := C_PCH_C_1, TCV_agch := C_AGCH_C_1, TCV_RaCh := C_RACH_C_1)			
22		[TCV_Ccchg=1]			
23		(TCV_PgCh := C_PCH_C_2, TCV_agch := C_AGCH_C_2, TCV_RaCh := C_RACH_C_2)			
24		[TCV_Ccchg=2]			
25		(TCV_PgCh := C_PCH_C_3, TCV_agch := C_AGCH_C_3, TCV_RaCh := C_RACH_C_3)			
26		[TCV_Ccchg=3]			
27		(TCV_PgCh := C_PCH_C_4, TCV_agch := C_AGCH_C_4, TCV_RaCh := C_RACH_C_4)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Varinit_fix(activ_cell:CellID; lac:OCTETSTRING; subch:B_2; cksn:B_3; rand:RAND; arfcngsm, arfcndcs:INTEGER; ccd: CCD; imsi:HEXSTRING)
Group: GSM_MS_CS_4220/Miscellaneous/
Objective:
Default: OtherEventsFail

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_cellid := activ_cell, TCV_ch := OC_SubchOfSdcch4(subch, activ_cell), TCV_sacch := OC_SubchOfSacch4(subch, activ_cell), TCV_ia_ts := '000'B, TCV_lac := lac, TCV_cksn :=cksn, TCV_CphKey := OC_CphKeyGen(TSPX_Ki, rand))			
2		+CCCH_group_Paging_group(ccd, imsi)			
3		+SelectPagingCh(activ_cell)			
4		[TSPC_PGSM OR TSPC_EGSM]			
5		(TCV_chdescr_arfcn:= arfcngsm)			
6		[TSPC_DCS]			
7		(TCV_chdescr_arfcn:= arfcndcs)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: Wait(par_int:INTEGER)
Group: GSM_MS_CS_4220/Miscellaneous/
Objective:
Default: OtherEventsFail
Comments: To check whether MS doesn't initiate any RR connections

Nr	Label	Behaviour Description	CRef	V	Comments
1		START T_dly(par_int)			
2		?TIMEOUT T_dly			

Detailed Comments: This test step can be used, e.g.for 1s. delay of a PAGING REQUEST message following immediately a CHANNEL RELEASE.

Test Step Dynamic Behaviour

Test Step Name: WaitForInService
Group: GSM_MS_CS_4220/Miscellaneous/
Objective: To wait until the MS enters the Idle and updated state.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Null:=OM_PgFill(TCV_cellid, PgReqTp1Reorg))			
2		START T_dly(5000)			
3		?TIMEOUT T_dly			
4		(TCV_Null:=OM_PgFill(TCV_cellid, PgReqTp1Norm))			

Detailed Comments:

Test Step Group SysInfo

Test Step Dynamic Behaviour					
Test Step Name:		SysInfoSending_cbms(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcchfldcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP; Sub:B_2)			
Group:		GSM_MS_CS_4220/Miscellaneous/SysInfo/			
Objective:		To send system information messages for the CBMS tests.			
Default:		OtherEvents			
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212), TCV_Tchtype := INT_TO_BIT((4 + BIT_TO_INT(Sub)), 5))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+ltree_sysinfo			
		ltree_sysinfo			
5		[TSPC_PGSM OR TSPC_EGSM]			
6		L!DL_UdatRqSysinfo4	SysInfo4_CBMS(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re, ChDescrp_nfh(TCV_Tchtype, C_S0, C_BCC, C_arfcnA))		
7		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
8		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
9		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
10		(TCV_sysinf5 := SysInf5(bcchflgsm))			
11		[TSPC_DCS]			
12		L!DL_UdatRqSysinfo4	SysInfo4_CBMS(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re, ChDescrp_nfh(TCV_Tchtype, C_S0, C_BCC, C_arfcnAd))		
13		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		
14		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
15		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
16		(TCV_sysinf5 := SysInf5(bcchfldcs))			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_fh(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm, cchddcs:CCHD; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcchfldcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP)

Group: GSM_MS_CS_4220/Miscellaneous/SysInfo/

Objective: To send system information messages for the L3 tests. The following parameters specified by input parameters:
- bcch: BCCH
- sch: SCH
- bcchfl: BCCH frequency list
- ci: Cell identity
- lai: Location area identification
- ccd: Control channel description
- csp: Cell selection parameters
- cchd: Cell channel description
- rachcpar: RACH controll paramenters
- nccp: NCC permitted

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+ltree_sysinfo			
		ltree_sysinfo			
5		[TSPC_PGSM OR TSPC_EGSM]			
6		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchdgsm, TCV_Max, TCV_Tx, Re)		
7		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
8		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
9		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
10		(TCV_sysinf5 := SysInf5(bcchflgsm))			
11		[TSPC_DCS]			
12		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchddcs, TCV_Max, TCV_Tx, Re)		
13		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		
14		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
15		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
16		(TCV_sysinf5 := SysInf5(bcchfldcs), TCV_sysinf6.co.pwrc := '1'B)			

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_fh_bisA(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm, cchddcs:CCHD; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcchfldcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP; ram:B_6; pt:B_5; cro:B_6; to:B_3)

Group: GSM_MS_CS_4220/Miscellaneous/SysInfo/

Objective: To send system information messages for the L3 tests. The following parameters specified by input parameters:
- bcch: BCCH
- sch: SCH
- bcchfl: BCCH frequency list
- ci: Cell identity
- lai: Location area identification
- ccd: Control channel description
- csp: Cell selection parameters
- cchd: Cell channel description
- rachcpar: RACH controll paramenters
- nccp: NCC permitted

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+ltree_sysinfo			
		ltree_sysinfo			
5		[TSPC_PGSM OR TSPC_EGSM]			
6		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchdgsm, TCV_Max, TCV_Tx, Re)		
7		L!DL_UdatRqSysinfo2	SysInfo2(bcch, BcchFreqLst_57, TCV_Max, TCV_Tx, Re, nccp)		
8		L!DL_UdatRqSysinfo3	SysInfo3_rpct(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, pt, cro, to)		
9		L!DL_UdatRqSysinfo4	SysInfo4_rpct(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, pt, cro, to)		
10		(TCV_sysinf5 := SysInf5(bcchflgsm))			
11		+SysInfo_SacchSending(TCV_sacch, TCV_sysinf5, TCV_sysinf6)			
12		L!DL_UdatRqSysinfo2bis	SysInfo2bis_egsm(bcch, BcchFreqLst_56)		
13		[TSPC_DCS]			
14		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchddcs, TCV_Max, TCV_Tx, Re)		
15		L!DL_UdatRqSysinfo2	SysInfo2(bcch, BcchFreqLst_58, TCV_Max, TCV_Tx, Re, nccp)		
16		L!DL_UdatRqSysinfo2bis	SysInfo2bis_egsm(bcch, BcchFreqLst_56)		
17		L!DL_UdatRqSysinfo3	SysInfo3_rpct(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx,		

			Re, ram, pt, cro, to)		
18		L!DL_UdatRqSysinfo4	SysInfo4_rpct(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, pt, cro, to)		
19		(TCV_sysinf5 := SysInf5(bcchfldcs), TCV_sysinf6.co.pwrc := '1'B)			
20		+SysInfo_SacchSending(TCV_sacch, TCV_sysinf5, TCV_sysinf6)			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_fh_bisB(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm, cchddcs:CCHD; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcchfldcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP; ram:B_6; pt:B_5; cro:B_6; to:B_3)

Group: GSM_MS_CS_4220/Miscellaneous/SysInfo/

Objective: To send system information messages for the L3 tests. The following parameters specified by input parameters:
- bcch: BCCH
- sch: SCH
- bcchfl: BCCH frequency list
- ci: Cell identity
- lai: Location area identification
- ccd: Control channel description
- csp: Cell selection parameters
- cchd: Cell channel description
- rachcpar: RACH controll paramenters
- nccp: NCC permitted

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+ltree_sysinfo			
		ltree_sysinfo			
5		[TSPC_PGSM OR TSPC_EGSM]			
6		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchdgsm, TCV_Max, TCV_Tx, Re)		
7		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
8		L!DL_UdatRqSysinfo3	SysInfo3_rpct(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, pt, cro, to)		
9		L!DL_UdatRqSysinfo4	SysInfo4_bis(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re, ram)		
10		(TCV_sysinf5 := SysInf5(bcchflgsm))			
11		+SysInfo_SacchSending(TCV_sacch, TCV_sysinf5, TCV_sysinf6)			
12		L!DL_UdatRqSysinfo7	SysInfo7(bcch, pt, cro, to)		
13		L!DL_UdatRqSysinfo8	SysInfo8(bcch, pt, cro, to)		
14		[TSPC_EGSM]			
15		L!DL_UdatRqSysinfo2ter	SysInfo2ter(bcch, BcchFreqLst2_65)		
16		[NOT TSPC_EGSM]			
17		[TSPC_DCS]			
18		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchddcs, TCV_Max, TCV_Tx, Re)		
19		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		
20		L!DL_UdatRqSysinfo3	SysInfo3_rpct(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh,		

			mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, pt, cro, to)		
21		L!DL_UdatRqSysinfo4	SysInfo4_bis(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re, ram)		
22		(TCV_sysinf5 := SysInf5(bcchfldcs), TCV_sysinf6.co.pwrc := '1'B)			
23		+SysInfo_SacchSending(TCV_sacch, TCV_sysinf5, TCV_sysinf6)			
24		L!DL_UdatRqSysinfo7	SysInfo7(bcch, pt, cro, to)		
25		L!DL_UdatRqSysinfo8	SysInfo8(bcch, pt, cro, to)		
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_fh_rcc(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm, cchddcs:CCHD; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcchfldcs, bcchflegsm:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP; ram:B_6; cba, cbq:B_1; acc:B_5)

Group: GSM_MS_CS_4220/Miscellaneous/SysInfo/

Objective: To send system information messages for the L3 tests. The following parameters specified by input parameters:
- bcch: BCCH
- sch: SCH
- bcchfl: BCCH frequency list
- ci: Cell identity
- lai: Location area identification
- ccd: Control channel description
- csp: Cell selection parameters
- cchd: Cell channel description
- rachcpar: RACH controll paramenters
- nccp: NCC permitted

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+ltree_sysinfo			
		ltree_sysinfo			
5		[TSPC_PGSM OR TSPC_EGSM]			
6		L!DL_UdatRqSysinfo1	SysInfo1_cba(bcch, cchdgsm, TCV_Max, TCV_Tx, Re, cba, acc)		
7		L!DL_UdatRqSysinfo2	SysInfo2_cba(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp, cba, acc)		
8		L!DL_UdatRqSysinfo3	SysInfo3_ram_cba(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, cba, cbq, acc)		
9		L!DL_UdatRqSysinfo4	SysInfo4_ram_cba(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, cba, cbq, acc)		
10		(TCV_sysinf5 := SysInf5(bcchflgsm))			
11		+SysInfo_SacchSending(TCV_sacch, TCV_sysinf5, TCV_sysinf6)			
12		[TSPC_EGSM]			
13		L!DL_UdatRqSysinfo2bis	SysInfo2bis_egsm(bcch, bcchflegsm)		
14		[NOT TSPC_EGSM]			
15		[TSPC_DCS]			
16		L!DL_UdatRqSysinfo1	SysInfo1_cba(bcch, cchddcs, TCV_Max, TCV_Tx, Re, cba, acc)		
17		L!DL_UdatRqSysinfo2	SysInfo2_cba(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp, cba, acc)		
18		L!DL_UdatRqSysinfo3	SysInfo3_ram_cba(		

			bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, cba, cbq, acc)		
19		L!DL_UdatRqSysinfo4	SysInfo4_ram_cba(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, cba, cbq, acc)		
20		(TCV_sysinf5 := SysInf5(bcchfldcs), TCV_sysinf6.co.pwrc := '1'B)			
21		+SysInfo_SacchSending(TCV_sacch, TCV_sysinf5, TCV_sysinf6)			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_fh_rpct(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm, cchddcs:CCHD; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcchfldcs, bcchflegsm:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP; ram:B_6; pt:B_5; cro:B_6; to:B_3)

Group: GSM_MS_CS_4220/Miscellaneous/SysInfo/

Objective: To send system information messages for the L3 tests. The following parameters specified by input parameters:
- bcch: BCCH
- sch: SCH
- bcchfl: BCCH frequency list
- ci: Cell identity
- lai: Location area identification
- ccd: Control channel description
- csp: Cell selection parameters
- cchd: Cell channel description
- rachcpar: RACH controll paramenters
- nccp: NCC permitted

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+ltree_sysinfo			
		ltree_sysinfo			
5		[TSPC_PGSM OR TSPC_EGSM]			
6		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchdgsm, TCV_Max, TCV_Tx, Re)		
7		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
8		L!DL_UdatRqSysinfo3	SysInfo3_rpct(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, pt, cro, to)		
9		L!DL_UdatRqSysinfo4	SysInfo4_rpct(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, pt, cro, to)		
10		(TCV_sysinf5 := SysInf5(bcchflgsm))			
11		+SysInfo_SacchSending(TCV_sacch, TCV_sysinf5, TCV_sysinf6)			
12		[TSPC_EGSM]			
13		L!DL_UdatRqSysinfo2bis	SysInfo2bis_egsm(bcch, bcchflegsm)		
14		[NOT TSPC_EGSM]			
15		[TSPC_DCS]			
16		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchddcs, TCV_Max, TCV_Tx, Re)		
17		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		
18		L!DL_UdatRqSysinfo3	SysInfo3_rpct(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, pt, cro, to)		
19		L!DL_UdatRqSysinfo4	SysInfo4_rpct(bcch,		

			mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, pt, cro, to)		
20		(TCV_sysinf5 := SysInf5(bcchfldcs), TCV_sysinf6.co.pwrc := '1'B)			
21		+SysInfo_SacchSending(TCV_sacch, TCV_sysinf5, TCV_sysinf6)			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_fh_rpctp(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm, cchddcs:CCHD; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcchfldcs, bcchflegsm:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP; ram:B_6; pt:B_5; cro:B_6; to:B_3; po:B_2)

Group: GSM_MS_CS_4220/Miscellaneous/SysInfo/

Objective: To send system information messages for the L3 tests. The following parameters specified by input parameters:
- bcch: BCCH
- sch: SCH
- bcchfl: BCCH frequency list
- ci: Cell identity
- lai: Location area identification
- ccd: Control channel description
- csp: Cell selection parameters
- cchd: Cell channel description
- rachcpar: RACH controll paramenters
- nccp: NCC permitted

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+ltree_sysinfo			
		ltree_sysinfo			
5		[TSPC_PGSM OR TSPC_EGSM]			
6		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchdgsm, TCV_Max, TCV_Tx, Re)		
7		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
8		L!DL_UdatRqSysinfo3	SysInfo3_rpctp(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, pt, cro, to, po)		
9		L!DL_UdatRqSysinfo4	SysInfo4_rpctp(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, pt, cro, to, po)		
10		(TCV_sysinf5 := SysInf5(bcchflgsm))			
11		+SysInfo_SacchSending(TCV_sacch, TCV_sysinf5, TCV_sysinf6)			
12		[TSPC_EGSM]			
13		L!DL_UdatRqSysinfo2bis	SysInfo2bis_egsm(bcch, bcchflegsm)		
14		[NOT TSPC_EGSM]			
15		[TSPC_DCS]			
16		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchddcs, TCV_Max, TCV_Tx, Re)		
17		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		
18		L!DL_UdatRqSysinfo3	SysInfo3_rpctp(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx,		

			Re, ram, pt, cro, to, po)		
19		L!DL_UdatRqSysinfo4	SysInfo4_rpctp(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re, ram, pt, cro, to, po)		
20		(TCV_sysinf5 := SysInf5(bcchfldcs), TCV_sysinf6.co.pwrc := '1'B)			
21		+SysInfo_SacchSending(TCV_sacch, TCV_sysinf5, TCV_sysinf6)			
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_nfh(sch, bcch:LOGICCH; t, retr, neci, att: INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP)

Group: GSM_MS_CS_4220/Miscellaneous/SysInfo/

Objective: To send system information messages with default parameters defined for the L3 tests for which no special parameters indicated.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+gsmOrDcs			
		gsmOrDcs			
5		[TSPC_PGSM OR TSPC_EGSM]			1.
6		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
7		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
8		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
9		[NOT TSPX_AltNb]			3.
10		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
11		(TCV_sysinf5 := SysInf5(bcchflgsm))			
12		[TSPX_AltNb]			4.
13		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcflaltgsm, TCV_Max, TCV_Tx, Re, nccp)		
14		(TCV_sysinf5 := SysInf5(bcflaltgsm))			
15		[TSPC_DCS]			2.
16		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
17		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
18		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
19		[NOT TSPX_AltNb]			3.
20		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		

21		(TCV_sysinf5 := SysInf5(bcchfldcs))			
22		[TSPX_AltNb]			4.
23		L!DL_UdatRqSysinfo2	SysInfo2(bcch, bcflaltdcs, TCV_Max, TCV_Tx, Re, nccp)		
24		(TCV_sysinf5 := SysInf5(bcflaltdcs))			

Detailed Comments:

1. For GSM900 mobile station testing.
2. For DCS1800 mobile station testing.
3. To use alternative neighbour cells description.
4. To use default neighbour cells description.

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_nfhCb(sch, bcch:LOGICCH; t, retr, neci, att: INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP)

Group: GSM_MS_CS_4220/Miscellaneous/SysInfo/

Objective: To send system information messages with default parameters defined for the L3 tests for which no special parameters indicated.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
3		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		
4		+gsmOrDcs			
		gsmOrDcs			
5		[TSPC_PGSM OR TSPC_EGSM]			1.
6		L!DL_UdatRqSysinfo4 (DL_UdatRqSysinfo4.msg.rachcp.cba := '1'B)	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
7		L!DL_UdatRqSysinfo1_nh (DL_UdatRqSysinfo1_nh.msg.rachcp.cba := '1'B)	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
8		L!DL_UdatRqSysinfo3 (DL_UdatRqSysinfo3.msg.rachcp.cba := '1'B)	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
9		[NOT TSPX_AltNb]			3.
10		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.rachcp.cba := '1'B)	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
11		(TCV_sysinf5 := SysInf5(bcchflgsm))			
12		[TSPX_AltNb]			4.
13		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.rachcp.cba := '1'B)	SysInfo2(bcch, bcflaltgsm, TCV_Max, TCV_Tx, Re, nccp)		
14		(TCV_sysinf5 := SysInf5(bcflaltgsm))			
15		[TSPC_DCS]			2.
16		L!DL_UdatRqSysinfo4 (DL_UdatRqSysinfo4.msg.rachcp.cba := '1'B)	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
17		L!DL_UdatRqSysinfo1_nh (DL_UdatRqSysinfo1_nh.msg.rachcp.cba := '1'B)	SysInfo1_nh(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
18		L!DL_UdatRqSysinfo3 (DL_UdatRqSysinfo3.msg.rachcp.cba := '1'B)	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcdcs, TCV_Neci, TCV_Max, TCV_Tx, Re)		
19		[NOT TSPX_AltNb]			3.
20		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.rachcp.cba := '1'B)	SysInfo2(bcch, bcchfldcs, TCV_Max, TCV_Tx, Re, nccp)		

21		(TCV_sysinf5 := SysInf5(bcchfldcs))			
22		[TSPX_AltNb]			4.
23		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.rachcp.cba := '1'B)	SysInfo2(bcch, bcflaltdcs, TCV_Max, TCV_Tx, Re, nccp)		
24		(TCV_sysinf5 := SysInf5(bcflaltdcs))			

Detailed Comments:
1. For GSM900 mobile station testing.
2. For DCS1800 mobile station testing.
3. To use alternative neighbour cells description.
4. To use default neighbour cells description.

Test Step Dynamic Behaviour

Test Step Name: SysInfoSending_e(sch, bcch:LOGICCH; t, retr, neci, att:INTEGER; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm:CCHD; crh, mtmcgsm:INTEGER; bcchflgsm, bcchflegsm:NCD; Re:B_1; bcc:BCC; ncc:NCC; nccp:NCCP)

Group: GSM_MS_CS_4220/Miscellaneous/SysInfo/

Objective: To send SYNCHRONIZATION CHANNEL INFORMATION and SYSTEM INFORMATION messages with default parameters except CCCH_CONF, Tx-integer, Max-Retrans and ATT which are specified by input parameters.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		L!DL_UdatRqSchinfo	SyncInfo(sch, bcc, ncc)		1.
3		(TCV_sysinf6 := SysInf6(ci, mcc, mnc, lac, co, nccp))			
4		L!DL_UdatRqSysinfo1	SysInfo1(bcch, cchdgsm, TCV_Max, TCV_Tx, Re)		2.
5		L!DL_UdatRqSysinfo2 (DL_UdatRqSysinfo2.msg.bcchfl.extind := '1'B)	SysInfo2(bcch, bcchflgsm, TCV_Max, TCV_Tx, Re, nccp)		
6		L!DL_UdatRqSysinfo2bis	SysInfo2bis(bcch, bcchflegsm)		
7		L!DL_UdatRqSysinfo4	SysInfo4(bcch, mcc, mnc, lac, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		
8		(TCV_sysinf5 := SysInf5(bcchflgsm), TCV_sysinfo5.bcchfl.extind := '1'B)			
9		(TCV_sysinfo5bis := SysInf5bis(bcchflegsm))			
10		L!DL_UdatRqSysinfo3	SysInfo3(bcch, ci, mcc, mnc, lac, TCV_Ccd0H, co, crh, mtmcgsm, TCV_Neci, TCV_Max, TCV_Tx, Re)		

Detailed Comments:
1. To send SYNCHRONIZATION INFORMATION message with default parameter for cell A.
2. System Information Type 1 with Cell Channel Description for HO-testin in cell A (GSM900).

Test Step Dynamic Behaviour					
Test Step Name:	SysInfo_SacchSending(ch:LOGICCH; sysinfo5_pdu:SYSINFO5_PDU; sysinfo6_pdu:SYSINFO6_PDU)				
Group:	GSM_MS_CS_4220/Miscellaneous/SysInfo/				
Objective:	To send SYSTEM INFORMATION 5 and 6 messages defined by parameters 'sysinfo5_pdu' and 'sysinfo6_pdu' in the parametrised 'ch' channel.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[ch <> "dummy"]			
2		L!DL_UdatRqSysinfo6	SysInfo6(ch, sysinfo6_pdu)		
3		L!DL_UdatRqSysinfo5	SysInfo5(ch, sysinfo5_pdu)		
4	CS_0094	[ch = "dummy"]		I	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	SysInfo_5bisSending(ch:LOGICCH; sysinfo5bis_pdu:SYSINFO5bis_PDU)				
Group:	GSM_MS_CS_4220/Miscellaneous/SysInfo/				
Objective:	To send SYSTEM INFORMATION 5bis message defined by parameters 'sysinfo5bis_pdu' in the parametrised 'ch' channel.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		[ch <> "dummy"]			
2		L!DL_UdatRqSysinfo5bis	SysInfo5bis(ch, sysinfo5bis_pdu)		
3	CS_0095	[ch = "dummy"]		I	
Detailed Comments:					

Test Step Dynamic Behaviour

Test Step Name: SetNECI(att:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING)
Group: GSM_MS_CS_4220/Miscellaneous/SysInfo/
Objective: To set the NECI =1.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := '0010'B, TCV_Max := '11'B, TCV_Neci := '1'B, TCV_Ccd0A := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		[TSPC_PGSM OR TSPC_EGSM]			
3		L!DL_UdatRqSysinfo4	SysInfo4(C_BCCH_A_1, mcc, mnc, lac, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
4		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
5		[TSPC_DCS]			
6		L!DL_UdatRqSysinfo4	SysInfo4(C_BCCH_A_1, mcc, mnc, lac, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
7		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: SetATT(t, retr, att, neci:INTEGER; babr, cch_con, bpm:B_3; t3212, mcc, mnc, lac:OCTETSTRING)
Group: GSM_MS_CS_4220/Miscellaneous/SysInfo/
Objective: To set the ATT flag to "MS's in the cell should apply IMSI attach and detach procedure"
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_Tx := OC_CnvtTx(t), TCV_Max := OC_CnvtMax(retr), TCV_Neci := INT_TO_BIT(neci,1), TCV_Ccd0A := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
2		[TSPC_PGSM OR TSPC_EGSM]			1.
3		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
4		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlG, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
5		START T_dly(5000)			
6		?TIMEOUT T_dly			3.
7		[TSPC_DCS]			2.
8		L!DL_UdatRqSysinfo1_nh	SysInfo1_nh(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
9		L!DL_UdatRqSysinfo3	SysInfo3(C_BCCH_A_1, C_ci_cellA, mcc, mnc, lac, TCV_Ccd0A, CellOpt_01, C_CellReselectHys12, C_MaxPwrLvlD, TCV_Neci, TCV_Max, TCV_Tx, C_Restablishment)		
10		START T_dly(5000)			3.
11		?TIMEOUT T_dly			

Detailed Comments:
1. For P-GSM900.
2. For DCS1800.
3. Wait for 5 seconds to allow the MS read BCCH information.

Test Step Group OperatorOP

Test Step Dynamic Behaviour					
Test Step Name:	InitCall(srv: SERVICES)				
Group:	GSM_MS_CS_4220/OperatorOP/				
Objective:	To initiate a call of the basic service 'srv'.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Null := OO_InitCall(srv))			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	PLMNsCHK				
Group:	GSM_MS_CS_4220/OperatorOP/				
Objective:	To check whether the MS presents a list of available PLMNs.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_Res := OO_PLMNsCHK())			
2	CS_0096	[TCV_Res]		P	
3	CS_0097	[NOT TCV_Res]		F	
Detailed Comments:					

Test Step Group Postambles

Test Step Dynamic Behaviour					
Test Step Name:	PostLinkRelEnd(ch: LOGICCH)				
Group:	GSM_MS_CS_4220/Postambles/				
Objective:	To release the RR connection and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(ch, ChRelease_01)		
2		L?DL_RelIn	DLRelInd_01		
3	CS_0098	CANCEL		R	1.
4		?TIMEOUT T_release			
5	CS_0099	CANCEL		R	1.
Detailed Comments:		1. Cancel of all running timers and final verdict.			

Test Step Dynamic Behaviour					
Test Step Name:	PostMainLinkRel(chnl:LOGICCH)				
Group:	GSM_MS_CS_4220/Postambles/				
Objective:	To release the main signalling link `ch`, and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(chnl, ChRelease_01)		
2		L?DL_RelIn CANCEL T_release	DLRelInd_01		
3		?TIMEOUT T_release			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	ChanRel(ch: LOGICCH)				
Group:	GSM_MS_CS_4220/Postambles/				
Objective:	To release the RR connection on the channel TCV_ch and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(ch, ChRelease_01)		
2		L?DL_RelIn CANCEL T_release	DLRelInd_01		
3	CS_0100	?TIMEOUT T_release		(F)	
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	ChanRel_P(ch: LOGICCH)				
Group:	GSM_MS_CS_4220/Postambles/				
Objective:	To release the RR connection on the channel TCV_chmaindcch and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(ch, ChRelease_01)		
2	CS_0101	L?DL_RelIn	DLRelInd_01	(P)	
3		?TIMEOUT T_release			
Detailed Comments:					

Test Step Dynamic Behaviour					
Test Step Name:	ChanRel_end(ch: LOGICCH)				
Group:	GSM_MS_CS_4220/Postambles/				
Objective:	To release the RR connection and bring the MS back to Idle state.				
Default:	OtherEvents				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L!DL_DatRqChRel START T_release	ChRel(ch, ChRelease_01)		
2		L?DL_RelIn	DLRelInd_01		
3	CS_0102	CANCEL		(P)	1.
4		?TIMEOUT T_release			
5	CS_0103	CANCEL		(F)	1.
Detailed Comments:		1. Cancel of all running timers.			

Test Step Group Preambles

Test Step Dynamic Behaviour					
Test Step Name:	BasicServiceMO(svc:SERVICES; rate:RATE)				
Group:	GSM_MS_CS_4220/Preambles/				
Objective:	To generate a Bcap, an MO SETUP message and an appropriate channel mode based on the service and rate, as well as on several IXIT parameters, itc, ce, ur, synchronous bit for an MO call.				
Default:	OtherEvents				
Comments:	Used for CC tests				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		+local_init			Default values
2		[TCV_Service = C_EmgCall]			
3		[(NOT TSPC_EFR_Speech_v2) AND(NOT TSPC_EFR_Speech_v3)]			
4		(TCV_ChMod.mode := C_ChMod_r, TCV_Esetup := Setup_04(Bcap_Speech_MO(TCV_Rchr)), TCV_Ecall := C_Yes)			Emergency Call - Non EFR Mobile
5		[TSPC_EFR_Speech_v2 OR TSPC_EFR_Speech_v3]			
6		(TCV_ChMod.mode := C_ChMod_r, TCV_Esetup := Setup_04(Bcap_Speech_Efr_MO(TCV_Rchr)), TCV_Ecall := C_Yes)			Emergency Call - EFR Mobile
7		[TCV_Service = C_Telephony]			
8		[(NOT TSPC_EFR_Speech_v2) AND(NOT TSPC_EFR_Speech_v3)]			
9		(TCV_ChMod.mode := C_ChMod_r, TCV_Setup_mo := Setup_02(Bcap_Speech_MO(TCV_Rchr)))			Speech - Non EFR Mobile
10		[TSPC_EFR_Speech_v2 OR TSPC_EFR_Speech_v3]			
11		(TCV_ChMod.mode := C_ChMod_r, TCV_Setup_mo := Setup_02(Bcap_Speech_Efr_MO(TCV_Rchr)))			Speech - EFR Mobile
12		[TCV_Service = C_AltSpchFax]			
13		[rate = C_Full]			
14		+TS61MO(C_AltSpchFax)			Speech/fax
15	CS_0104	[rate <> C_Full]		I	IXIT error
16		[TCV_Service = C_AutoFax]			
17		[rate = C_Full]			
18		+TS62MO(C_AutoFax)			Auto fax
19	CS_0105	[rate <> C_Full]		I	IXIT error
20		[TCV_Service = C_Async300]			
21		+BS2xMO(C_Async300, TSPX_BS_21_sacp, TSPX_BS_21_T_NT, TSPX_BS_21_itc1, C_300bs, C_modemt_V21, TSPX_BS_21_ce)			BS21
22		[TCV_Service = C_Async1200]			
23		+BS2xMO(C_Async1200, TSPX_BS_22_sacp, TSPX_BS_22_T_NT, TSPX_BS_22_itc1, C_1200bs, C_modemt_V22, TSPX_BS_22_ce)			BS22
24		[TCV_Service = C_Async120075]			
25		+BS2xMO(C_Async120075, TSPX_BS_23_sacp, TSPX_BS_23_T_NT, TSPX_BS_23_itc, C_120075bs, C_modemt_V23, TSPX_BS_23_ce)			BS23
26		[TCV_Service = C_Async2400]			
27		+BS2xMO(C_Async2400, TSPX_BS_24_sacp, TSPX_BS_24_T_NT, TSPX_BS_24_itc1, C_2400bs, TCV_MODEM.V22bisOrV26ter, TSPX_BS_24_ce)			BS24
28		[TCV_Service = C_Async4800]			
29		+BS2xMO(C_Async4800, TSPX_BS_25_sacp, TSPX_BS_25_T_NT, TSPX_BS_25_itc1, C_4800bs, C_modemt_V32, TSPX_BS_25_ce)			BS25
30		[TCV_Service = C_Async9600]			
31		+BS2xMO(C_Async9600, TSPX_BS_26_sacp, TSPX_BS_26_T_NT, TSPX_BS_26_itc1,			BS26

		C_9600bs, C_modemt_V32, TSPX_BS_26_ce)			
32		[TCV_Service = C_Sync1200]			
33		+BS3xMO(C_Sync1200, FALSE, TSPX_BS_31_itc1, TSPX_BS_31_sacp1, C_1200bs, C_transparent, C_modemt_V22)			BS31
34		[TCV_Service = C_Sync2400]			
35		+BS3xMO(C_Sync2400, TSPX_BS_32_X32_T_NT, TSPX_BS_32_itc1, TSPX_BS_32_sacp1, C_2400bs, TSPX_BS_32_X32_ce, TCV_MODEM.V22bisOrV26ter)			BS32
36		[TCV_Service = C_Sync4800]			
37		+BS3xMO(C_Sync4800, TSPX_BS_33_X32_T_NT, TSPX_BS_33_itc1, TSPX_BS_33_sacp1, C_4800bs, TSPX_BS_33_X32_ce, C_modemt_V32)			BS33
38		[TCV_Service = C_Sync9600]			
39		+BS3xMO(C_Sync9600, TSPX_BS_34_X32_T_NT, TSPX_BS_34_itc1, TSPX_BS_34_sacp1, C_9600bs, TSPX_BS_34_X32_ce, C_modemt_V32)			BS34
40		[TCV_Service = C_PAD300]			
41		+BS4xMO(C_PAD300, C_300bs, TSPX_BS_41_ce, TSPX_BS_41_T_NT)			BS41
42		[TCV_Service = C_PAD1200]			
43		+BS4xMO(C_PAD1200, C_1200bs, TSPX_BS_42_ce, TSPX_BS_42_T_NT)			BS42
44		[TCV_Service = C_PAD120075]			
45		+BS4xMO(C_PAD120075, C_120075bs, TSPX_BS_43_ce, TSPX_BS_43_T_NT)			BS43
46		[TCV_Service = C_PAD2400]			
47		+BS4xMO(C_PAD2400, C_2400bs, TSPX_BS_44_ce, TSPX_BS_44_T_NT)			BS44
48		[TCV_Service = C_PAD4800]			
49		+BS4xMO(C_PAD4800, C_4800bs, TSPX_BS_45_ce, TSPX_BS_45_T_NT)			BS45
50		[TCV_Service = C_PAD9600]			
51		+BS4xMO(C_PAD9600, C_9600bs, TSPX_BS_46_ce, TSPX_BS_46_T_NT)			BS46
52		[TCV_Service = C_Packet2400]			
53		+BS5xMO(C_Packet2400, C_2400bs)			BS51
54		[TCV_Service = C_Packet4800]			
55		+BS5xMO(C_Packet4800, C_4800bs)			BS52
56		[TCV_Service = C_Packet9600]			
57		+BS5xMO(C_Packet9600, C_9600bs)			BS53
58		[TCV_Service = C_AltSpchData]			
59		+BS61or81MO(C_AltSpchData, TSPX_BS_61_A, TSPX_BS_61_A_ur1_T_NT, TSPX_BS_61_A_ur1, TSPX_BS_61_A_ur1_ce, TSPX_BS_61_S, TSPX_BS_61_S_ur1, C_RI_alternate)			BS61
60		[TCV_Service = C_SpchData]			
61		+BS61or81MO(C_SpchData, TSPX_BS_81_A, TSPX_BS_81_A_ur1_T_NT, TSPX_BS_81_A_ur1, TSPX_BS_81_A_ur1_ce, TSPX_BS_81_S, TSPX_BS_81_S_ur1, C_RI_follow)			BS81
62	CS_0106	[C_Yes]		I	IXIT error
		local_init			
63		(TCV_Service := svc, TCV_Rchr := C_Rchr_Full, TCV_Ecall := C_No, TCV_ChRate :=rate, TCV_ChMod.iei := '01100011'B, TCV_ChMod.mode := C_ChMod_12k, TCV_ChModb.iei := '01100011'B, TCV_ChModb.mode := C_ChMod_12k, TCV_MODEM := V22bisOrV26ter)			
64		[TSPC_DualRate]			
65		(TCV_Rchr := C_Rchr_DualForDualH0)			
66		[NOT TSPC_DualRate]			

Detailed Comments:	This test step is called in the preamble of test cases where an MO call is established in the test body.

Test Step Dynamic Behaviour

Test Step Name: BS2xMO(srv:SERVICES; sacp: B_3; both:BOOLEAN; itc1:B_3; ur:B_4; modem:B_5; ce:B_2)
Group: GSM_MS_CS_4220/Preambles/
Objective: To generate an MO SETUP message with appropriate Bcap and LLC for BS2x service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_srv := srv, TCV_ur := ur, TCV_CE := NT_BothT_BothNTorT_BothT_BothNT, TCV_ir := C_ir_16kbs)			
2		+local_pixitvaluechk			validate IXIT values
3		+local_selectCe			select a connection element
4		+local_bcap			generate BC
5		+local_setupmsg			generate SETUP
6		+Chmod(TCV_ChRate, TCV_ce, ur, srv)			initiate channel mode variables
7		(TCV_Null := OO_IFsetup_BS2xorBS3x_MO(TCV_srv, itc1, sacp, TCV_ce, TCV_ChRate))			ask test operator to configure the MS
		local_bcap			
8		[NOT both]			
9		[TCV_ce = C_notransparent]			
10		[itc1 = C_UDI]			
11		+local_BS2x_UDI_NT(C_notransparent, sacp)			
12		[itc1 = C_3100Hz]			
13		+local_BS2x_3100_NT(C_notransparent, sacp)			
14		[TCV_ce = C_transparent]			CE:T only MS
15		[itc1 = C_UDI]			
16		+local_BS2x_UDI_T(sacp)			
17		[itc1 = C_3100Hz]			
18		+local_BS2x_3100_T(sacp)			
19		[both]			
20		[TCV_ce = C_transparent]			
21		[itc1 = C_UDI]			
22		+local_BS2x_UDI_Tboth(TCV_CE.TorBothTorBothNT,sacp)			
23		[itc1 = C_3100Hz]			3.1kHz
24		+local_BS2x_3100_Tboth(TCV_CE.TorBothTorBothNT, sacp)			
25		[(TCV_ce = C_notransparent)]			
26		[itc1 = C_UDI]			
27		+local_BS2x_UDI_NT(TCV_CE.NTorBothTorBothNT,sacp)			
28		[itc1 = C_3100Hz]			3.1kHz
29		+local_BS2x_3100_NT(TCV_CE.NTorBothTorBothNT, sacp)			
		local_BS2x_3100_NT(cel: B_2; sacp:B_3)			
30		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_3100Hz, C_SDUintegrity, C_nirr_dontcare, C_no_rate_adaption, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, cel, modem, C_uil2p_dontcare))			11.8.2.1.1.2
		local_BS2x_UDI_NT(cel: B_2; sacp:B_3)			
31		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_UDI, C_SDUintegrity, C_nirr_dontcare, C_rate_adaption_V110, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, cel, C_modemt_none, C_uil2p_dontcare))			11.8.2.1.1.4
		local_BS2x_3100_T(sacp:B_3)			
32		[ur = C_9600bs]			
33		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr,			11.8.2.1.1.1

		C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, C_transparent, modem))			
34		[(ur = C_300bs) OR (ur = C_1200bs) OR (ur = C_120075bs) OR (ur = C_2400bs) OR (ur = C_4800bs)]			
35		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent, modem), TCV_ir := C_ir_8kbs)			11.8.2.1.1.1
		local_BS2x_UDI_T(sacp:B_3)			
36		[ur = C_9600bs]			
37		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, C_transparent, C_modemt_none))			11.8.2.1.1.3
38		[(ur = C_300bs) OR (ur = C_1200bs) OR (ur = C_120075bs) OR (ur = C_2400bs) OR (ur = C_4800bs)]			
39		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent, C_modemt_none), TCV_ir := C_ir_8kbs)			11.8.2.1.1.3
		local_BS2x_3100_Tboth(cel: B_2; sacp:B_3)			
40		[ur <> C_9600bs]			
41		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_3100Hz, C_struc_dontchk, C_nirr_dontcare, C_no_rate_adaption, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8or16kbs, C_parity_dontcare, cel, modem, C_uil2p_dontcare))			11.8.2.1.1.1 conbinrd with 11.8.2.1.1.2 , CE and structure without exact match
42		[ur = C_9600bs]			
43		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_3100Hz, C_struc_dontchk, C_nirr_nomeaning, C_no_rate_adaption, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8or16kbs, C_parity_dontcare, cel, modem, C_uil2p_dontcare))			11.8.2.1.1.1 conbinrd with 11.8.2.1.1.2 , CE and structure without exact match
		local_BS2x_UDI_Tboth(cel: B_2; sacp:B_3)			
44		[ur = C_9600bs]			
45		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_UDI, C_struc_dontchk, C_nirr_nomeaning, C_rate_adaption_V110, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8or16kbs, C_parity_dontcare, cel, C_modemt_none, C_uil2p_dontcare))			11.8.2.1.1.3 conbined with 11.8.2.1.1.4, CE and structure without exact match
46		[ur <> C_9600bs]			
47		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_UDI, C_struc_dontchk, C_nirr_dontcare, C_rate_adaption_V110, sacp, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8or16kbs, C_parity_dontcare, cel, C_modemt_none, C_uil2p_dontcare))			11.8.2.1.1.3 conbined with 11.8.2.1.1.4, CE and structure without exact match
		local_setupmsg			
48		[itc1 = C_UDI]			
49		(TCV_Setup_mo := Setup_10_UDI(TCV_BcapMO1, Llcmp_BS2xOrBS4x_UDI, Hlcmp_AnyOrOmit))			
50		[itc1 = C_3100Hz]			
51		(TCV_Setup_mo := Setup_10_3100(TCV_BcapMO1,			

		Llcmp_BS2xOrBS61aOrBS81a_3100, Hlcmp_AnyOrOmit))			
		local_selectCe			
52		[both]			
53		[(ur = C_300bs) OR (ur = C_120075bs) OR (ur = C_4800bs)]			
54		(TCV_ce := C_notransparent)			
55		[(ur = C_1200bs) OR (ur = C_2400bs) OR (ur = C_9600bs)]			
56		(TCV_ce := C_transparent)			
57		[NOT both]			
58		(TCV_ce := ce)			
		local_pixitvaluechk			
59	CS_0107	[(ce <> C_transparent) AND(ce <> C_notransparent) AND(NOT both)]		I	IXIT error
60	CS_0108	[(itc1 <> C_UDI) AND(itc1 <> C_3100Hz)]		I	IXIT error
61	CS_0109	[(sacp <> C_I440_450) AND(sacp <> C_X28_nond)]		I	IXIT error
62		[C_Yes]			IXIT values OK

Detailed Comments:

The BS2x Bearer Capabilities and SETUP messages for MO is constructed according to section 11.8.2.1.1.1, section 11.8.2.1.1.2, section 11.8.2.1.1.3 and section 11.8.2.1.1.4 of GSM 11.10 and section B.1.2.1, section B.1.2.2, section B.1.2.1 and section B.2.2.2 of GSM 07.01.

The algorithm for selecting parameters:

```
IF (TSPX_BS_2x_T_NT)
        TCV_ce := C_notransparent
ELSE
        TCV_ce := TSPX_BS_2x_ce
```

use (TCV_ce, TSPX_BS_2x_itc1, TSPX_BS_2x_sacp1, svc) for OO_IFsetup_BS2xorBS3x_MO

Test Step Dynamic Behaviour

Test Step Name: BS3xMO(srv:SERVICES; both:BOOLEAN; itc1, sacp1:B_3; ur:B_4; ce:B_2; modem:B_5)
Group: GSM_MS_CS_4220/Preambles/
Objective: To generate an MO SETUP message with appropriate Bcap and LLC for BS3x service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_srv := srv, TCV_ur := ur, TCV_CE := NT_BothT_BothNTorT_BothT_BothNT, TCV_ir := C_ir_16kbs, TCV_uil1p := C_no_rate_adaption)			
2		+local_pixitvaluechk			2.
3		+local_selectCe			select a CE
4		+local_bcap			generate BC
5		+local_setupmsg			3.
6		+Chmod(TCV_ChRate, TCV_ce, ur, srv)			initiate channel mode variables
7		(TCV_Null := OO_IFsetup_BS2xorBS3x_MO(TCV_srv, itc1, sacp1, TCV_ce, TCV_ChRate))			ask test operator to configure the MS
		local_bcap			
8		[NOT both]			
9		[TCV_ce = C_notransparent]			
10		[itc1 = C_UDI]			
11		[sacp1 = C_X32]			
12		+local_BS3x_UDI_X32_NT			
13	CS_0110	[sacp1 <> C_X32]		I	5.
14		[itc1 = C_3100Hz]			
15	CS_0111	[sacp1 <> C_X32]		I	6.
16		[sacp1 = C_X32]			
17		+local_BS3x_3100_X32_NT(C_notransparent)			
18		[TCV_ce = C_transparent]			CE:T only MS
19		[itc1 = C_UDI]			
20	CS_0112	[sacp1 = C_X32]		I	7.
21		[sacp1<> C_X32]			
22		+local_BS3x_UDI_T(sacp1)			
23		[itc1 = C_3100Hz]			
24	CS_0113	[sacp1 = C_X21]		I	8.
25		[sacp1 = C_I440_450]			
26		+local_BS3x_3100_T			
27		[sacp1 = C_X32]			
28		+local_BS3x_3100_X32_T			
29		[both]			
30		[TCV_ce = C_transparent]			
31		[itc1 = C_UDI]			
32	CS_0114	[sacp1 = C_X32]		I	7.
33		[sacp1<> C_X32]			
34		+local_BS3x_UDI_T(sacp1)			
35		[itc1 = C_3100Hz]			3.1kHz
36	CS_0115	[sacp1 = C_X21]		I	8.
37		[sacp1 = C_I440_450]			
38		+local_BS3x_3100_T			
39		[sacp1 = C_X32]			
40		+local_BS3x_3100_X32_NT(TCV_CE.TorBothTorBothNT)			
41		[(TCV_ce = C_notransparent)]			
42		[itc1 = C_UDI]			
43		[sacp1 = C_X32]			
44		+local_BS3x_UDI_X32_NT			
45	CS_0	[sacp1 <> C_X32]		I	5.

	116				
46		[itc1 = C_3100Hz]			3.1kHz
47	CS_0 117	[sacp1 <> C_X32]		I	6.
48		[sacp1 = C_X32]			
49		+local_BS3x_3100_X32_Tboth(TCV_CE.NTorBothTorBothNT)			
		local_setupmsg			
50		[itc1 = C_UDI]			
51		(TCV_Setup_mo := Setup_10_UDI(TCV_BcapMO1, Llcmp_BS3x_UDI, Hlcmp_AnyOrOmit))			
52		[itc1 = C_3100Hz]			
53		(TCV_Setup_mo := Setup_10_3100(TCV_BcapMO1, Llcmp_BS3xOrBS61sOrBS81s_3100, Hlcmp_AnyOrOmit))			
		local_BS3x_3100_T			
54		[ur = C_9600bs]			
55		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, C_transparent, modem))			11.8.2.2.1.1
56		[ur <> C_9600bs]			
57		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_8kbs, C_parity_NA, C_transparent, modem))			11.8.2.2.1.1
		local_BS3x_UDI_T(sacp:B_3)			
58		[ur = C_9600bs]			
59		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, sacp, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, C_transparent, C_modemt_none), TCV_uil1p := C_rate_adaption_V110)			11.8.2.2.1.2
60		[NOT (ur = C_9600bs)]			
61		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, sacp, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_8kbs, C_parity_NA, C_transparent, C_modemt_none), TCV_ir := C_ir_8kbs, TCV_uil1p := C_rate_adaption_V110)			11.8.2.2.1.2
		local_BS3x_3100_X32_T			
62		[ur = C_9600bs]			
63		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, C_transparent, modem))			11.8.2.2.1.3
64		[ur <> C_9600bs]			
65		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_8kbs, C_parity_NA, C_transparent, modem))			11.8.2.2.1.3
		local_BS3x_3100_X32_NT(ce2:B_2)			
66		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_3100Hz, C_SDUintegrity, C_nirr_dontcare, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, ce2, modem, C_X25_flct))			11.8.2.2.1.4

		local_BS3x_UDI_X32_NT			
67		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_UDI, C_SDUintegrity, C_nirr_dontcare, C_rate_adaption_X31, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, C_notransparent, C_modemt_none, C_X25_flct), TCV_uil1p := C_rate_adaption_X31)			11.8.2.2.1.5
		local_BS3x_3100_X32_Tboth(ce2:B_2)			
68		[ur = C_9600bs]			
69		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_3100Hz, C_struc_dontchk, C_nirr_nomeaning, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, ce2, modem, C_X25_flct))			11.8.2.2.1.3 combined with 11.8.2.2.1.4, CE and structure without exact match
70		[ur <> C_9600bs]			
71		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_3100Hz, C_struc_dontchk, C_nirr_dontcare, C_no_rate_adaption, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_8or16kbs, C_parity_NA, ce2, modem, C_X25_flct))			11.8.2.2.1.3 combined with 11.8.2.2.1.4, CE, IR and structure without exact match
		local_selectCe			
72		[both]			
73		[(ur = C_300bs) OR (ur = C_120075bs) OR (ur = C_4800bs)]			
74		(TCV_ce := C_transparent)			
75		[(ur = C_1200bs) OR (ur = C_2400bs) OR (ur = C_9600bs)]			
76		(TCV_ce := C_notransparent)			
77		[NOT both]			
78		(TCV_ce := ce)			
		local_pixitvaluechk			
79	CS_0118	[(ce <> C_transparent) AND(ce <> C_notransparent) AND(NOT both)]		I	IXIT error
80	CS_0119	[(itc1 <> C_UDI) AND(itc1 <> C_3100Hz)]		I	IXIT error
81	CS_0120	[(sacp1 <> C_I440_450) AND(sacp1 <> C_X21) AND(sacp1 <>C_X32)]		I	IXIT error
82	CS_0121	[(ur = C_1200bs) AND (sacp1 = C_X32)]		I	4.
83		[C_Yes]			IXIT values OK, but combination not checked yet

Detailed Comments:

1. The BS3x Bearer Capabilities and SETUP messages for MO is constructed according to section
   11.8.2.2.1.1, section 11.8.2.2.1.2, section 11.8.2.2.1.3, section 11.8.2.2.1.4 and section 11.8.2.2.1.5 of GSM 11.10 and section B.1.3.1.1, section B.1.3.1.2, section B.1.3.2.1, section B.1.3.2.2, section B.2.3.1 and section B.2.3.2 of GSM 07.01.
2. validate IXIT values.
3. generate SETUP.
4. IXIT error, BS31 does not support X32.
5. IXIT error, UDI non-X32 case has no non transparent mode.
6. IXIT error, 3.1kHz non-X32 case has no non transparent mode.
7. IXIT error, UDI X32 case has no transparent mode.
8. IXIT error, 3.1kHz case has no X21 signalling access protocol.

The algorithm for selecting parameters:

```
IF (TSPX_BS_3x_T_NT)
        TCV_ce := C_notransparent
ELSE
        TCV_ce := TSPX_BS_3x_ce

use (TCV_ce, TSPX_BS_3x_itc1, TSPX_BS_3x_sacp1, svc) for OO_IFsetup_BS2xorBS3x_MO
```

Test Step Dynamic Behaviour

Test Step Name: BS4xMO(srv:SERVICES; ur:B_4; ce:B_2; both:BOOLEAN)
Group: GSM_MS_CS_4220/Preambles/
Objective: To generate an MO SETUP message with appropriate Bcap and LLC for BS4x service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_srv := srv, TCV_ur := ur, TCV_CE := NT_BothT_BothNTorT_BothT_BothNT, TCV_ir := C_ir_16kbs)			
2		+local_pixitvaluechk			validate IXIT values
3		+local_selectCe			select a CE
4		+local_bcap			generate BC
5		(TCV_Setup_mo := Setup_10_UDI(TCV_BcapMO1, Llcmp_BS2xOrBS4x_UDI, Hlcmp_AnyOrOmit))			generate SETUP
6		+Chmod(TCV_ChRate, TCV_ce, ur, srv)			initiate channel mode variables
7		(TCV_Null := OO_IFsetup_BS4x_MO(TCV_srv, TCV_ce, TCV_ChRate))			ask test operator to set up the MS
		local_bcap			
8		[NOT both]			
9		[TCV_ce = C_notransparent]			
10		+local_BS4x_UDI_NT(C_notransparent)			
11		[TCV_ce = C_transparent]			CE:T only MS
12		[ur = C_9600bs]			
13		+local_BS4x_UDI_T(C_ir_16kbs)			
14		[(ur = C_300bs) OR (ur = C_1200bs) OR (ur = C_120075bs) OR (ur = C_2400bs) OR (ur = C_4800bs)]			
15		+local_BS4x_UDI_T(C_ir_8kbs)			
16		[both]			
17		[TCV_ce = C_transparent]			
18		+local_BS4x_UDI_Tboth(TCV_CE.TorBothTorBothNT)			
19		[(TCV_ce = C_notransparent)]			
20		+local_BS4x_UDI_NT(TCV_CE.NTorBothTorBothNT)			
		local_BS4x_UDI_T(ir:B_2)			
21		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_UDI, C_Unstructured, C_nirr_nomeaning, C_rate_adaption_V110, C_X28_unui, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, ir, C_parity_dontcare, C_transparent, C_modemt_none), TCV_ir := ir)			11.8.2.3.1.1
		local_BS4x_UDI_NT(ce1:B_2)			
22		[(ur = C_300bs) OR (ur = C_1200bs) OR (ur = C_120075bs) OR (ur = C_2400bs) OR (ur = C_4800bs)]			
23		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_UDI, C_SDUintegrity, C_nirr_dontcare, C_rate_adaption_V110, C_X28_unui, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, ce1, C_modemt_none, C_uil2p_dontcare))			11.8.2.3.1.2
24		[ur = C_9600bs]			
25		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_UDI, C_SDUintegrity, C_nirr_nomeaning, C_rate_adaption_V110, C_X28_unui, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, ce1, C_modemt_none, C_uil2p_dontcare))			11.8.2.3.1.2
		local_BS4x_UDI_Tboth(ce1:B_2)			

26		[(ur = C_300bs) OR (ur = C_1200bs) OR (ur = C_120075bs) OR (ur = C_2400bs) OR (ur = C_4800bs)]			
27		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_UDI, C_struc_dontchk, C_nirr_dontcare, C_rate_adaption_V110, C_X28_unui, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8or16kbs, C_parity_dontcare, ce1, C_modemt_none, C_uil2p_dontcare))			11.8.2.3.1.2 conbined with 11.8.2.3.1.1, CE, IR and structure without exact match
28		[ur = C_9600bs]			
29		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_UDI, C_struc_dontchk, C_nirr_nomeaning, C_rate_adaption_V110, C_X28_unui, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8or16kbs, C_parity_dontcare, ce1, C_modemt_none, C_uil2p_dontcare))			11.8.2.3.1.2 conbined with 11.8.2.3.1.1, CE, IR and structure without exact match
		local_pixitvaluechk			
30	CS_0122	[(ce <> C_transparent) AND(ce <> C_notransparent) AND(NOT both)]		I	IXIT error
31		[C_Yes]			IXIT values OK
		local_selectCe			
32		[both]			
33		[(ur = C_300bs) OR (ur = C_120075bs) OR (ur = C_4800bs)]			
34		(TCV_ce := C_notransparent)			
35		[(ur = C_1200bs) OR (ur = C_2400bs) OR (ur = C_9600bs)]			
36		(TCV_ce := C_transparent)			
37		[NOT both]			
38		(TCV_ce := ce)			

Detailed Comments:

The BS4x Bearer Capabilities and SETUP messages for MO is constructed according to section 11.8.2.3.1.1 and section 11.8.2.3.1.2 of GSM 11.10 and section B.1.4 and section B.2.2.1 of GSM 07.01.

The algorithm for selecting parameters:

```
IF (TSPX_BS_4x_T_NT)
        IF (user rate is one of the C_300bs, C_120075bs, C_4800bs)
                TCV_ce := C_notransparent
        ELSE   /* user rate is one of C_1200bs, C_2400bs, C_9600bs \
```

Test Step Dynamic Behaviour

Test Step Name: BS5xMO(srv:SERVICES; ur:B_4)
Group: GSM_MS_CS_4220/Preambles/
Objective: To generate an MO SETUP message with appropriate Bcap for BS5x service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_srv := srv, TCV_ur := ur)			
2		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_UDI, C_SDUintegrity, C_nirr_nomeaning, C_rate_adaption_X31, C_X32, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, C_notransparent, C_modemt_none, C_X25_flct))			11.8.2.4.1
3		(TCV_Setup_mo := Setup_10_UDI(TCV_BcapMO1, Llcmp_BS5x, Hlcmp_AnyOrOmit))			
4		+Chmod(TCV_ChRate, C_notransparent, ur, srv)			channel mode
5		(TCV_Null := OO_IFsetup_BS5x_MO(TCV_srv, TCV_ur, TCV_ChRate))			

Detailed Comments:
1. Bearer Caoability is coded according to section 11.8.2.4.1 of GSM 11.10 ver 4.16
2. low layer compatibility is coded according to section B.2.5 of GSM 07.01 ver 4.90

Test Step Dynamic Behaviour					
Test Step Name:		BS61or81MO(srv:SERVICES; async, both: BOOLEAN; a_ur1: B_4; ce:B_2; sync:BOOLEAN; s_ur1: B_4; ri:B_8)			
Group:		GSM_MS_CS_4220/Preambles/			
Objective:		To get a MO SETUP message with right BC IE and LLC for BS61 or BS81 service.			
Default:		OtherEvents			
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		(TCV_srv := srv, TCV_CE := NT_BothT_BothNTorT_BothT_BothNT, TCV_ir := C_ir_8kbs, TCV_MODEM := V22bisOrV26ter)			
2		+local_pixitvaluechk			1.
3		+local_selectSaAndCe			2.
4		+local_bcap			generate BC
5		+local_setupmsg			generate SETUP
6		+Chmod(TCV_ChRate, TCV_ce, TCV_ur, srv)			
7		(TCV_Null := OO_IFsetup_BS61orBS81_MO(TCV_srv, TCV_ur, TCV_sa, TCV_ce, TCV_ChRate))			
		local_bcap			
8		[TCV_sa = C_Synchronous]			
9		[TCV_ur = C_1200bs]			
10		+local_BS61or81_S_T(C_ir_8kbs, C_1200bs, C_modemt_V22)			
11		[TCV_ur = C_2400bs]			
12		+local_BS61or81_S_T(C_ir_8kbs, C_2400bs, TCV_MODEM.V22bisOrV26ter)			
13		[TCV_ur = C_4800bs]			
14		+local_BS61or81_S_T(C_ir_8kbs, C_4800bs, C_modemt_V32)			
15		[TCV_ur = C_9600bs]			
16		+local_BS61or81_S_T(C_ir_16kbs, C_9600bs, C_modemt_V32)			
17		(TCV_ir := C_ir_16kbs)			
18		[TCV_sa = C_Asynchronous]			
19		[(TCV_ce = C_transparent) AND(NOT both)]			CE:T only MS
20		+local_BS61or81_A_T(TCV_ur)			
21		[(TCV_ce = C_notransparent) AND(NOT both)]			CE:NT only MS
22		+local_BS61or81_A_NT(TCV_ur, C_notransparent, C_SDUintegrity, C_nirr_dontcare, C_ir_16kbs)			
23		[(TCV_ce = C_notransparent) AND both]			
24		+local_BS61or81_A_NT(TCV_ur, TCV_CE.NTorBothTorBothNT, C_SDUintegrity, C_nirr_dontcare, C_ir_16kbs)			
25		[(TCV_ce = C_transparent) AND both]			
26		+local_BS61or81_A_NT(TCV_ur, TCV_CE.TorBothTorBothNT, C_struc_dontchk, C_nirr_dontcare, C_ir_8or16kbs)			
		local_BS61or81_A_T(ur:B_4)			
27		[ur = C_300bs]			
28		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent, C_modemt_V21), TCV_modemt := C_modemt_V21)			11.8.2.5.1
29		[ur = C_1200bs]			
30		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent,			11.8.2.5.1

		C_modemt_V22), TCV_modemt := C_modemt_V22)			
31		[ur = C_120075bs]			
32		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent, C_modemt_V23), TCV_modemt := C_modemt_V23)			11.8.2.5.1
33		[ur = C_2400bs]			
34		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent, TCV_MODEM.V22bisOrV26ter), TCV_modemt := TCV_MODEM.V22bisOrV26ter)			11.8.2.5.1
35		[ur = C_4800bs]			
36		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_8kbs, C_parity_dontcare, C_transparent, C_modemt_V32), TCV_modemt := C_modemt_V32)			11.8.2.5.1
37		[ur = C_9600bs]			
38		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, C_transparent, C_modemt_V32), TCV_modemt := C_modemt_V32)			11.8.2.5.1
39		(TCV_ir := C_ir_16kbs)			
		local_BS61or81_A_NT(ur:B_4; ce:B_2; struc:B_2; nirr :B_1; ir:B_2)			
40		(TCV_ir := C_ir_16kbs)			
41		[ur = C_300bs]			
42		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_3100Hz, struc, nirr, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, ir, C_parity_dontcare, ce, C_modemt_V21, C_uil2p_dontcare), TCV_modemt := C_modemt_V21)			11.8.2.5.2
43		[ur = C_1200bs]			
44		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_3100Hz, struc, nirr, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, ir, C_parity_dontcare, ce, C_modemt_V22, C_uil2p_dontcare), TCV_modemt := C_modemt_V22)			11.8.2.5.2
45		[ur = C_120075bs]			
46		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_3100Hz, struc, nirr, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, ir, C_parity_dontcare, ce, C_modemt_V23, C_uil2p_dontcare), TCV_modemt := C_modemt_V23)			11.8.2.5.2
47		[ur = C_2400bs]			
48		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_3100Hz, struc, nirr, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, ir, C_parity_dontcare, ce, TCV_MODEM.V22bisOrV26ter,			11.8.2.5.2

		C_uil2p_dontcare), TCV_modemt := TCV_MODEM.V22bisOrV26ter)			
49		[ur = C_4800bs]			
50		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_3100Hz, struc, nirr, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, ir, C_parity_dontcare, ce, C_modemt_V32, C_uil2p_dontcare), TCV_modemt := C_modemt_V32)			11.8.2.5.2
51		[ur = C_9600bs]			
52		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_3100Hz, struc, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Asynchronous, C_stopbit_dontcare, C_databit_dontcare, ur, C_ir_16kbs, C_parity_dontcare, ce, C_modemt_V32, C_uil2p_dontcare), TCV_modemt := C_modemt_V32)			11.8.2.5.2
		local_BS61or81_S_T(ir:B_2; ur: B_4; modemt:B_5)			
53		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_3100Hz, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_I440_450, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, ir, C_parity_NA, C_transparent, modemt), TCV_modemt := modemt)			11.8.2.5.3
		local_pixitvaluechk			
54	CS_0123	[(ce <> C_transparent) AND(ce <> C_notransparent) AND(NOT both) AND async]		I	IXIT error
55	CS_0124	[(a_ur1 <> C_300bs) AND(a_ur1 <> C_1200bs) AND(a_ur1 <> C_2400bs) AND(a_ur1 <> C_4800bs) AND(a_ur1 <> C_9600bs) AND(a_ur1 <> C_120075bs) AND async]		I	IXIT error
56	CS_0125	[(s_ur1 <> C_1200bs) AND(s_ur1 <> C_2400bs) AND(s_ur1 <> C_4800bs) AND(s_ur1 <> C_9600bs) AND sync]		I	IXIT error
57		[C_Yes]			IXIT values OK
		local_selectSaAndCe			
58		[async]			3.
59		(TCV_sa := C_Asynchronous, TCV_ur := a_ur1)			
60		[both]			
61		[(a_ur1 = C_300bs) OR (a_ur1 = C_120075bs) OR (a_ur1 = C_4800bs)]			
62		(TCV_ce := C_notransparent)			
63		[(a_ur1 = C_1200bs) OR (a_ur1 = C_2400bs) OR (a_ur1 = C_9600bs)]			
64		(TCV_ce := C_transparent)			
65		[NOT both]			
66		(TCV_ce := ce)			
67		[sync]			
68		(TCV_sa := C_Synchronous, TCV_ce := C_transparent, TCV_ur := s_ur1)			
		local_setupmsg			
69		[TCV_sa = C_Asynchronous]			
70		[(NOT TSPC_EFR_Speech_v2) AND(NOT TSPC_EFR_Speech_v3)]			
71		(TCV_Setup_mo := Setup_11_3100(ri, Bcap_Speech_MO(C_Rchr_dontcare), TCV_BcapMO1, Llcmp_NotApplicable, Llcmp_BS2xOrBS61aOrBS81a_3100, ri, ri, Hlcmp_NotApplicable, Hlcmp_AnyOrOmit))			Non EFR Mobile
72		[TSPC_EFR_Speech_v2 OR TSPC_EFR_Speech_v3]			
73		(TCV_Setup_mo := Setup_11_3100(ri, Bcap_Speech_Efr_MO(C_Rchr_dontcare), TCV_BcapMO1, Llcmp_NotApplicable, Llcmp_BS2xOrBS61aOrBS81a_3100, ri, ri,			EFR Mobile

		Hlcmp_NotApplicable, Hlcmp_AnyOrOmit))			
74		[TCV_sa = C_Synchronous]			
75		[(NOT TSPC_EFR_Speech_v2) AND(NOT TSPC_EFR_Speech_v3)]			
76		(TCV_Setup_mo := Setup_11_3100(ri, Bcap_Speech_MO(C_Rchr_dontcare), TCV_BcapMO1, Llcmp_NotApplicable, Llcmp_BS3xOrBS61sOrBS81s_3100, ri, ri, Hlcmp_NotApplicable, Hlcmp_AnyOrOmit))			Non EFR Mobile
77		[TSPC_EFR_Speech_v2 OR TSPC_EFR_Speech_v3]			
78		(TCV_Setup_mo := Setup_11_3100(ri, Bcap_Speech_Efr_MO(C_Rchr_dontcare), TCV_BcapMO1, Llcmp_NotApplicable, Llcmp_BS3xOrBS61sOrBS81s_3100, ri, ri, Hlcmp_NotApplicable, Hlcmp_AnyOrOmit))			EFR Mobile

Detailed Comments:

1. validate IXIT values
2. select sync or async mode and corresponding CE
3. If the MS supports both synchronous and asynchronous mode, selete the asynchronous mode
4. The BS61 or BS81 Bearer Capabilities and SETUP messages for MO is constructed according to section 11.8.2.5.1, section 11.8.2.5.2 and section 11.8.2.5.3 of GSM 11.10 and section B.1.6.1, section B.1.6.2.1, section B.1.6.2.2, section B.2.2.2 and section B.2.3.2 of GSM 07.01.

The algorithm for selecting parameters for BS61 basice service:

```
IF (TSPX_BS_61_A)
        TCV_sa := C_Asynchronous
        IF (TSPX_BS_61_A_ur1_T_NT)
                TCV_ce := C_notransparent
        ELSE
                TCV_ce := TSPX_BS_61_A_ur1_ce
        use (TCV_sa, TCV_ce, TSPX_BS_61_A_ur1, svc) for OO_IFsetup_BS61orBS81_MO
ELSE
        TCV_sa := C_Synchronous
        use (TCV_sa, C_transparent, TSPX_BS_61_S_ur1, svc) for
OO_IFsetup_BS61orBS81_MO
```

The algorithm for selecting parameters for BS61 basice service is the same as above except using corresponding IXIT parameters.

Test Step Dynamic Behaviour

Test Step Name:	Chmod(rate:RATE; ce:B_2; ur:B_4; svc:SERVICES)
Group:	GSM_MS_CS_4220/Preambles/
Objective:	To assign correct values to the test case variables TCV_ChMod and TCV_ChModb.
Default:	OtherEvents
Comments:	The rules for assignment are in Table 10-1 of GSM 11.10 section 10.3. If the value for ce is "BothT", the rule for transparent is used. If the value for ce is "BothNT", the rule for non transparent is used. That means the preferred value will be used in test configuration.

Nr	Label	Behaviour Description	CRef	V	Comments
1		+local_tree			
2		[(svc = C_AltSpchData) OR (svc = C_SpchData) OR (svc = C_AltSpchFax)]			dual mode call
3		(TCV_ChMod.mode := C_ChMod_r)			
4		[(svc<> C_AltSpchData) AND(svc <> C_SpchData) AND(svc <> C_AltSpchFax)]			
		local_tree			
5		[((ce = C_notransparent) OR (ce = C_BothNT)) AND (rate = C_Full)]			
6		(TCV_ChMod.mode := C_ChMod_12k, TCV_ChModb.mode := C_ChMod_12k)			
7		[((ce = C_notransparent) OR (ce = C_BothNT)) AND (rate = C_Half)]			
8		[ur <> C_9600bs]			
9		(TCV_ChMod.mode := C_ChMod_6k, TCV_ChModb.mode := C_ChMod_6k)			
10	CS_0126	[ur = C_9600bs]		I	not applicable
11		[(ce = C_transparent) OR (ce = C_BothT)]			
12		[(ur = C_9600bs) AND (rate = C_Full)]			
13		(TCV_ChMod.mode := C_ChMod_12k, TCV_ChModb.mode := C_ChMod_12k)			
14		[ur = C_4800bs]			
15		(TCV_ChMod.mode := C_ChMod_6k, TCV_ChModb.mode := C_ChMod_6k)			
16		[(ur <> C_4800bs) AND(ur <> C_9600bs)]			
17		(TCV_ChMod.mode := C_ChMod_3k, TCV_ChModb.mode := C_ChMod_3k)			
18	CS_0127	[(ur = C_9600bs) AND (rate = C_Half)]		I	not applicable

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: IdleUpdated(par_bspwr:INTEGER; activ_cell:CellID; sch, bcch:LOGICCH; acttype:BITSTRING; slot, slot2, slot3, slot4:SN; tsc:TSC; t, retr, neci, att:INTEGER; ta:TA; babr, cch_con, bpm:B_3; t3212:OCTETSTRING; ci:CI; mcc, mnc, lac:OCTETSTRING; co:CO; cchdgsm, cchddcs:CCHD; crh, mtmcgsm, mtmcdcs:INTEGER; bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs:NCD; Re:B_1; bcc:BCC; ncc:NCC; StartType:INTEGER; subch:B_2; cksn:B_3; rand:RAND; arfcngsm, arfcndcs:INTEGER; imsi:HEXSTRING; sysinfonb:INTEGER; nccp:NCCP)

Group: GSM_MS_CS_4220/Preambles/

Objective: To ensure that the SIM is either updated to the initial conditions and the MS with CKSN valid, TMSI valid and idle updated in cell A, B or C; or idle updated without TMSI; or totally not updated and all LAI information deleted.

Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_slot := slot, TCV_tsc := tsc, TCV_Ccd0H := CntrlChDscrp(att, babr, C_cch_1Comb, bpm, t3212))			
2		+MM_PwrOrSimOff(C_SIMneedRmv)			
3		+Varinit_fix(activ_cell, lac, subch, cksn, rand, arfcngsm, arfcndcs, TCV_Ccd0H, imsi)			
4		+ltree_chconf(C_cch_1Comb)			Config LT
5		+localtree1			
		localtree1			
6		+ltree_sysinfo(1, C_cch_1Comb)			ATT= '1'B
7		+MM_PwrOrSimOn(C_SIMneedRmv)			
8		+localtree_Lup_Auth			
		localtree_Lup_Auth			
9		START T_dly(C_T_Wait1stChReq)			
10		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_02)		Any channel request PDU
11		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
12		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, subch, slot, tsc, TCV_chdescr_arfcn, ta))		
13		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(C_norm_period_attach))		Any location update type
14		ACTIVATE(OtherEvents)			Restore Normal default
15		[StartType = C_Start_Rej]			
16		L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(C_rc_LAnotallowed))		Delete LAI, tmsi, cksn
17		+PostMainLinkRel(TCV_ch)			Release Ch
18		+MM_PwrOrSimOff(C_SIMneedRmv)			
19		+localtree2			
20		[StartType = C_Start_Imsi]			
21		+Authentication(TCV_ch, cksn, rand)			
22		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiImsi_01iei, mcc, mnc, TCV_lac))		Location Accept with IMSI
23		+PostMainLinkRel(TCV_ch)			Release Ch
24		+localtree2			
25		[(StartType = C_Start_Tmsi) OR (StartType = C_Start_TmsiOff)]			
26		+Authentication(TCV_ch, cksn, rand)			
27		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiTmsi_01iei, mcc, mnc, TCV_lac))		Location Accept with TMSI
28		L?DL_DatInTmsireCom	TmsiReallocCmp(TCV_ch)		
29		+PostMainLinkRel(TCV_ch)			Release Ch

30		+localtree2			
31		[StartType = C_Start_PLMNnot]			
32		L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(C_rc_plmn_not))		Delete LAI, tmsi, cksn
33		+PostMainLinkRel(TCV_ch)			Release Ch
34		+localtree2			
35	CS_0128	[(StartType <> C_Start_Rej) AND(StartType <> C_Start_Tmsi) AND(StartType <> C_Start_TmsiOff) AND(StartType <> C_Start_Imsi) AND(StartType <> C_Start_PLMNnot)]		I	
36		?TIMEOUT T_dly			
37		+ltree_rmvPLMNfbdnlist			
		localtree2			
38		[(att = 1) AND(StartType <> C_Start_TmsiOff)]			ATT no change
39		[cch_con <> C_cch_1Comb]			If non combined
40		+localtree3			
41		+ltree_PgReorg			
42		[cch_con = C_cch_1Comb]			If combined
43		[(att = 1) AND (StartType = C_Start_TmsiOff)]			
44		+ImsiDetach(MiTmsi_01, ta, C_SIMneedRmv)			Switch off, detach
45		[cch_con <> C_cch_1Comb]			If non combined
46		+localtree3			
47		[cch_con = C_cch_1Comb]			If combined
48		[att = 0]			
49		[cch_con = C_cch_1Comb]			If combined
50		+ltree_sysinfo(0, C_cch_1Comb)			ATT=0
51		+ltree_PgReorg			
52		[cch_con <> C_cch_1Comb]			If non combined
53		+localtree3			
54		+ltree_PgReorg			
		localtree3			
55		(TCV_Ccd0H := CntrlChDscrp(att, babr, cch_con, bpm, t3212))			
56		+Varinit_fix(activ_cell, lac, subch, cksn, rand, arfcngsm, arfcndcs, TCV_Ccd0H, imsi)			
57		+ltree_chconf(cch_con)			Config LT
58		+ltree_sysinfo(att, cch_con)			
		ltree_PgReorg			
59		(TCV_Null:=OM_PgFill(activ_cell, PgReqTp1Reorg))			
60		START T_dly(5000)			
61		?TIMEOUT T_dly			
62		(TCV_Null:=OM_PgFill(activ_cell, PgReqTp1Norm))			
		ltree_sysinfo(attach:INTEGER; con_ch:B_3)			
63		[activ_cell = C_CellA]			
64		+SysInfoSending(attach, con_ch)			SI1-4, cellA
65		(TCV_Ccd0A := TCV_Ccd0H, TCV_sysinfo5 := TCV_sysinf5, TCV_sysinfo6 := TCV_sysinf6)			
66		[con_ch = C_cch_1Comb]			If combined
67		+SysInfo_SacchSending(TCV_sacch, TCV_sysinfo5, TCV_sysinfo6)			SI5, 6 on SACCH/C4, cell A
68		[con_ch <> C_cch_1Comb]			
69		+ltree_sysinfo_other(attach)			More SIs, cellA
70		[activ_cell = C_CellB]			
71		+SysInfoSending(attach, con_ch)			SI1-4, cellB
72		(TCV_Ccd0B := TCV_Ccd0H, TCV_sacch_B := TCV_sacch, TCV_sysinfo5_B := TCV_sysinf5, TCV_sysinfo6_B := TCV_sysinf6)			SI5, 6 on SACCH/C4, cell B
73		[con_ch = C_cch_1Comb]			If combined
74		+SysInfo_SacchSending(TCV_sacch_B, TCV_sysinfo5_B, TCV_sysinfo6_B)			SI5, 6 on SACCH/C4, cell A

75		[con_ch <> C_cch_1Comb]			No more SIs, cell B
76		[activ_cell = C_CellC]			
77		+SysInfoSending(attach, con_ch)			SI1-4, cellC
78		(TCV_Ccd0C := TCV_Ccd0H, TCV_sacch_C := TCV_sacch, TCV_sysinfo5_C := TCV_sysinf5, TCV_sysinfo6_C := TCV_sysinf6)			
79		[con_ch = C_cch_1Comb]			If combined
80		+SysInfo_SacchSending(TCV_sacch_C, TCV_sysinfo5_C, TCV_sysinfo6_C)			SI5, 6 on SACCH/C4, cell C
81		[con_ch <> C_cch_1Comb]			No more SIs, cell C
		SysInfoSending(attachb:INTEGER; con_chs:B_3)			
82		[sysinfonb = C_Test_fh]			Hopping
83		+SysInfoSending_fh(sch, bcch, t, retr, neci, attachb, babr, con_chs, bpm, t3212, ci, mcc, mnc, lac, co, cchdgsm, cchddcs, crh, mtmcgsm, mtmcdcs, bcchflgsm, bcchfldcs, Re, bcc, ncc, nccp)			
84		[sysinfonb = C_Test_nfh]			Non hopping
85		+SysInfoSending_nfh(sch, bcch, t, retr, neci, attachb, babr, con_chs, bpm, t3212, ci, mcc, mnc, lac, co, crh, mtmcgsm, mtmcdcs, bcchflgsm, bcflaltgsm, bcchfldcs, bcflaltdcs, Re, bcc, ncc, nccp)			
86		[sysinfonb = C_Test_egsm]			EGSM
87		+SysInfoSending_e(sch, bcch, t, retr, neci, attachb, babr, con_chs, bpm, t3212, ci, mcc, mnc, lac, co, cchdgsm, crh, mtmcgsm, bcchflgsm, bcchfldcs, Re, bcc, ncc, nccp)			
88		[sysinfonb = C_Test_cbms]			CBMS
89		+SysInfoSending_cbms(sch, bcch, t, retr, neci, attachb, babr, con_chs, bpm, t3212, ci, mcc, mnc, lac, co, crh, mtmcgsm, mtmcdcs, bcchflgsm, bcchfldcs, Re, bcc, ncc, nccp, subch)			
		ltree_chconf(conf:B_3)			
90		[activ_cell = C_CellA]			
91		+ltree_chconf_A(conf)			Config LT
92		[activ_cell = C_CellB]			
93		+ltree_chconf_B(conf)			Config LT
94		[activ_cell = C_CellC]			
95		+ltree_chconf_C(conf)			Config LT
		ltree_chconf_A(conf_a:B_3)			
96		[conf_a = C_cch_1nonComb]			
97		+NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1 non combined CCCH, cell A
98		(TCV_Upd := C_NotCombined)			
99		[conf_a = C_cch_1Comb]			
100		[sysinfonb <> C_Test_cbms]			
101		+CombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1 combined CCCH, cell A
102		(TCV_Upd := C_Combined)			
103		[sysinfonb = C_Test_cbms]			
104		+CombinedBCCH_A_CBMS(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1 combined CCCH + CBMS, cell A
105		(TCV_Upd := C_Combined)			
106		[conf_a = C_cch_2nonComb]			2 CCCHs
107		+NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1st non combined CCCH, cell A
108		+NonCombinedBCCH_A_2(acttype, slot2, tsc, ta, babr, conf_a, bpm)			2nd non combined CCCH, cell A
109		(TCV_Upd := C_NotCombined)			
110		[conf_a = C_cch_3nonComb]			
111		+NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta,			1st non combined CCCH, cell A

		babr, conf_a, bpm)			
112		+NonCombinedBCCH_A_2(acttype, slot2, tsc, ta, babr, conf_a, bpm)			2nd non combined CCCH, cell A
113		+NonCombinedBCCH_A_3(acttype, slot3, tsc, ta, babr, conf_a, bpm)			3rd non combined CCCH, cell A
114		(TCV_Upd := C_NotCombined)			
115		[conf_a = C_cch_4nonComb]			
116		+NonCombinedBCCH_A(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_a, bpm)			1st non combined CCCH, cell A
117		+NonCombinedBCCH_A_2(acttype,slot2, tsc, ta, babr, conf_a, bpm)			2nd non combined CCCH, cell A
118		+NonCombinedBCCH_A_3(acttype, slot3, tsc, ta, babr, conf_a, bpm)			3rd non combined CCCH, cell A
119		+NonCombinedBCCH_A_4(acttype, slot4, tsc, ta, babr, conf_a, bpm)			4th non combined CCCH, cell A
120		(TCV_Upd := C_NotCombined)			
121	CS_0129	[(conf_a <> C_cch_1nonComb) AND(conf_a <> C_cch_2nonComb) AND(conf_a <> C_cch_3nonComb) AND(conf_a <> C_cch_4nonComb) AND(conf_a <> C_cch_1Comb)]		I	A wrong CCCH-CONF value, cell A
		ltree_chconf_B(conf_b:B_3)			
122		[conf_b = C_cch_1nonComb]			
123		+NonCombinedBCCH_B(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype,slot,tsc, ta, babr, conf_b, bpm)			1 non combined CCCH, cell B
124		(TCV_Upd := C_NotCombined)			
125		[conf_b = C_cch_1Comb]			
126		+CombinedBCCH_B(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_b, bpm)			1 combined CCCH, cell B
127		(TCV_Upd := C_Combined)			
128	CS_0130	[(conf_b <> C_cch_1nonComb) AND(conf_b <> C_cch_1Comb)]		I	A wrong CCCH-CONF value,cell B
		ltree_chconf_C(conf_c:B_3)			
129		[conf_c = C_cch_1Comb]			
130		+CombinedBCCH_C(par_bspwr, mtmcgsm, mtmcdcs, arfcngsm, arfcndcs, acttype, slot, tsc, ta, babr, conf_c, bpm)			1 combined CCCH, cell C
131		(TCV_Upd := C_Combined)			
132	CS_0131	[conf_c <> C_cch_1Comb]		I	
		ltree_sysinfo_other(attach:INTEGER)			
133		[(cch_con = C_cch_1nonComb) OR (cch_con = C_cch_1Comb)]			
134		[cch_con = C_cch_2nonComb]			
135		+SysInfoSending_fh(sch, C_BCCH_A_2, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 2nd CCCH
136		[cch_con = C_cch_3nonComb]			
137		+SysInfoSending_fh(sch, C_BCCH_A_2, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 2nd CCCH
138		+SysInfoSending_fh(sch, C_BCCH_A_3, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 3rd CCCH
139		[cch_con = C_cch_4nonComb]			
140		+SysInfoSending_fh(sch, C_BCCH_A_2, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 2nd CCCH

141		+SysInfoSending_fh(sch, C_BCCH_A_3, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 3rd CCCH
142		+SysInfoSending_fh(sch, C_BCCH_A_4, t, retr, neci, attach, babr, cch_con, bpm, t3212, ci, mcc, mnc, lac, co, CellChDes_02, CellChDes_03, crh, mtmcgsm, mtmcdcs, BcchFreqLst_01, BcchFreqLst_48, Re, bcc, ncc, nccp)			SIs on 4th CCCH
		ltree_rmvPLMNfbdnlist			
143		(TCV_Null := OO_PLMNselModeMan())			
144		(TCV_Null := OO_SelPLMN(mnc))			
145		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq. msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn) CANCEL T_dly	ChReq(ChRequest_02)		Any channel request PDU
146		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
147		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, subch, slot, tsc, TCV_chdescr_arfcn, ta))		
148		L?DL_EstInLupRq	LocUp(TCV_ch, LocUpdtReq_01(C_norm_period_attach))		Any location update type
149		ACTIVATE(OtherEvents)			Restore Normal default
150		[StartType = C_Start_PLMNnot]			The PLMN is already in forbidden list on short SIM
151		L!DL_DatRqLupRej	LocRej(TCV_ch, LocUpdtRej_01(C_rc_plmn_not))		Delete LAI, tmsi, cksn, PLMN still in list
152		+PostMainLinkRel(TCV_ch)			Release Ch
153		(TCV_Null := OO_PLMNselModeAuto())			
154		[StartType <> C_Start_PLMNnot]			
155		L!DL_DatRqLupAcp	LocAcp(TCV_ch, LocUpdtAcp_01(MiTmsi_r02iei, mcc, mnc, TCV_lac))		Location Accept with no TMSI
156		+PostMainLinkRel(TCV_ch)			Release Ch
157	CS_0132	(TCV_Null := OO_PLMNselModeAuto())		I	No ChReq received: Inconclusive

Detailed Comments:

Formal Parameter Type Comments & Explanations Possible values

par_bspwr INTEGER Base station power level C_E_default, C_E_suitable

activ_cell CellID Activated cell identity C_CellA, C_CellB, C_CellC

sch LOGICCH Sync. channel identity
If formal para activ_cell = C_CellA, C_SCH_A
If formal para activ_cell = C_CellB, C_SCH_B
If formal para activ_cell = C_CellC, C_SCH_C

bcch LOGICCH 1st broadcast channel id
If formal para activ_cell = C_CellA, C_BCCH_A_1
If formal para activ_cell = C_CellB, C_BCCH_B_1
If formal para activ_cell = C_CellC, C_BCCH_C_1

acttype BITSTRING Type for activation of a channel C_Immass

slot SN Slot number for the 1st slot C_S0

slot2 SN Slot number for the 2nd slot
In case of 2 or 3 or 4 non combined CCCH C_S2

Otherwise (1 combined or 1 non combined) C_SlotNU

slot3 SN Slot number for the 3rd slot
In case of 3 or 4 non combined CCCH C_S4
Otherwise (1 combined or 1, 2 non combined) C_SlotNU

slot4 SN Slot number for the 4th slot
In case of 4 non combined CCCH channels C_S6
Otherwise (1 combined,1, 2, 3 non combined) C_SlotNU

tsc TSC Training Sequence code C_BCC

t INTEGER Rach Control Parameter field, Tx-Integer
for SysInfo1, SysInfo2, SysInfo3, SysInfo4 C_TxInt_5, C_TxInt_7, C_TxInt_10

retr INTEGER Rach Control Parameter field, Max-Retrans
for SysInfo1, SysInfo2, SysInfo3, SysInfo4 C_Max_1, C_Max_2, C_Max_7

neci INTEGER Requested value f. NECI bit in test case C_NECI_0, C_NECI_1

att INTEGER Requested value f. ATTACH bit in test case C_ATT_0, C_ATT_1

ta TA Requested Timing Advance value

babr B_3 Number of blocks reserved for acces grant
If arbitrary TSPX_AGBLKSx
If fixed C_BABR_0, C_BABR_1, C_BABR_2

cch_con B_3 Type of CCCH channel.
If arbitrary TSPX_CcchConfx
1 conbined CCCH channel C_cch_1Comb
1 non conbined CCCH channel C_cch_1nonComb
2 non conbined CCCH channels C_cch_2nonComb
3 non conbined CCCH channels C_cch_3nonComb
4 non conbined CCCH channels C_cch_4nonComb

bpm B_3 Number of multiframes for transmission
of PAGING REQUEST messages to the same
paging subgroup
If arbitrary TSPX_PAMFRMSx
If fixed (2, 4, 5, 9 multiframes) C_BPM_0, C_BPM_2, C_BPM_3, C_BPM_7

t3212 OCTETSTR. Requested value for T3212 in test case (Periodic Location Update Value)
infinite C_T3212_0
6 mn / 1 deci hours C_T3212_1
12 mn / 2 deci hours C_T3212_2
30 mn / 5 deci hours C_T3212_5

ci CI Cell Identity
If formal para activ_cell = C_CellA C_ci_CellA
If formal para activ_cell = C_CellB C_ci_CellB
If formal para activ_cell = C_CellC C_ci_CellC

mcc OCTETSTR. Mobile Contry Code for the LAI C_MCC, C_MCC_x

mnc OCTETSTR. Mobile Network Code for the LAI C_PLMN_Home, C_PLMN_x

lac OCTETSTR. Local Area Code for the LAI C_LAC_x

co CO Cell Options CellOpt_xx

cchdgsm CCHD Cell Channels Description (GSM)
If formal para sysinfonb=C_Test_nfh CellChDes_omit
If formal para sysinfonb=C_Test_cbms CellChDes_omit
If formal para sysinfonb=C_Test_fh GSM CellChDes_xx

cchddcs CCHD Cell Channels Description (DCS)
If formal para sysinfonb=C_Test_nfh CellChDes_omit
If formal para sysinfonb=C_Test_cbms CellChDes_omit
If formal para sysinfonb=C_Test_fh DCS CellChDes_xx

crh INTEGER Cell Reselection Hysteresis C_CellReselectHys12, C_CellReselectHys4

mtmcgsm INTEGER Requested maximum power level for GSM MS C_MaxPwrLvlG

mtmcdcs INTEGER Requested maximum power level for DCS MS C_MaxPwrLvlD

bcchflgsm NCD Neighbour Cells Description for GSM BcchFreqLst_xx

bcflaltgsm NCD Alternative Neighbour Cells Description for GSM.
If sysinfonb = C_Test_nfh Alternat.GSM BcchFreqLst_xx
Otherwise: BcchFreqLst_Omit

bcchfldcs NCD Neighbour Cells Description for DCS BcchFreqLst_xx

bcflaltdcs NCD Alternative Neighbour Cells Description for DCS
If sysinfonb = C_Test_nfh Alternat.DCS BcchFreqLst_xx
Otherwise: BcchFreqLst_Omit

Re B_1 Rach Control Parameter field,
Re-establishment bit for SysInfo1,
SysInfo2, SysInfo3, SysInfo4 C_Restablishment, C_noRestablishmen

bcc BCC Base station Color Code C_BCC

ncc NCC Network Color Code C_NCC

StartType INTEGER Initial conditions for MS
Starting without TMSI C_Start_Imsi
Starting with Tmsi_01 C_Start_Tmsi
Starting with Tmsi_01, MS deactivated C_Start_TmsiOff
Starting with LAI deleted, MS deactivated C_Start_Rej
Starting with PLMN not allowed C_Start_PLMNnot

subch BITSTRING Selected sub-channel
If Formal para cch_con = C_cch_1Comb, TSPX_SDCCH4SubDef TSPX_SDCCH4SubA, SubB, SubC

cksn BITSTRING Ciphering Key Sequence Number used for authentication TSPX_CKSNDef, TSPX_CKSNA, B, C

rand BITSTRING Random Number for authentication TSPX_RANDDef, TSPX_RANDA, B , C

arfcngsm INTEGER BCCH channel ARFCN (GSM)
If sysinfonb = C_Test_fh C_Noarfcn
Otherwise: If activ_cell = C_CellA C_arfcnA_x If activ_cell = C_CellB C_arfcnB_x If activ_cell = C_CellC C_arfcnC_x

arfcndcs INTEGER BCCH channel ARFCN (DCS)
If sysinfonb = C_Test_fh C_Noarfcn

	Otherwise:	If activ_cell = C_CellA C_arfcnAd_x If activ_cell = C_CellB C_arfcnBd_x If activ_cell = C_CellC C_arfcnCd_x
imsi	HEXSTRING	The expected IMSI In TC_26_7_3_2 TSPX_IMSI C_shortIMSI
sysinfonb	INTEGER	Type of sysinfo sending
		Frequency Hopping C_Test_fh
		Non Frequency hopping C_Test_nfh
		EGSM sequence C_Test_egsm
		CBMS sequence C_Test_cbms
nccp	NCCP	Network Country Code Permitted in SysInfo2 C_NCCP_2, C_NCCP_5

Test Step Dynamic Behaviour

Test Step Name: PreEstRRConn(slot:SN; tsc:TSC; ta:TA)
Group: GSM_MS_CS_4220/Preambles/
Objective: To establish a RR connection on SDCCH subchannel defined by TSPX_SDCCH4SubDef in cell A
Default: OtherEvents
Comments: The calling tree shall prepare variable for the step: TCV_ch for the SDCCH4 subchannel TSPX_SDCCH4SubDef. This test step is used for non RR testing.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_01)		
2		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
3		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
4		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, slot, tsc, TCV_chdescr_arfcn, ta))		
5		L?DL_EstInPgRes	PagingRes(PagingRes_01)		
6		ACTIVATE(OtherEvents)			Restore Normal default

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: PreEstRRC_MM(par_mi:MI; cksn:BITSTRING; ta:TA)
Group: GSM_MS_CS_4220/Preambles/
Objective: To establish a RR connection on C_SDCCH4_A_1
Default: OtherEvents
Comments: used var's:
TCV_Rr, TCV_Fn, TCV_Pgch, TCV_ia_ts, TCV_chdescr_arfcn

Nr	Label	Behaviour Description	CRef	V	Comments
1		L!DL_UdatRqPg1Rq	PgReq1(TCV_PgCh, TCV_Pgg, PgReqTp1_30(par_mi))		
2		L?DL_RacInChRq (TCV_Rr := DL_RacInChRq.msg.ecau_rrf, TCV_Fn := DL_RacInChRq.fn)	ChReq(ChRequest_17)		
3		ACTIVATE(OtherEvents_02)			To match ChReq retrans.
4		L!DL_UdatRqImmass	ImmAss(TCV_agch, ImmAsgn_01Def(TCV_Rr, TCV_Fn, TSPX_SDCCH4SubDef, TCV_slot, TCV_tsc, TCV_chdescr_arfcn, ta))		
5	CS_0133	L?DL_EstInPgRes	PagingRes(PagingRes_30(par_mi, cksn))	(P)	
6		ACTIVATE(OtherEvents)			Restore Normal default

Detailed Comments:

Test Step Dynamic Behaviour

Test Step Name: TS61MO(srv:SERVICES)
Group: GSM_MS_CS_4220/Preambles/
Objective: To generate an MO SETUP message with appropriate Bcap for TS61 service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_srv := srv, TCV_ur := TSPX_TS_61_ur1, TCV_CE := NT_BothT_BothNTorT_BothT_BothNT)			
2		+local_selectCe			
3		+local_bcap			get the right BC
4		[(NOT TSPC_EFR_Speech_v2) AND(NOT TSPC_EFR_Speech_v3)]			
5		(TCV_Setup_mo := Setup_11_3100(C_RI_alternate, Bcap_Speech_MO(C_Rchr_dontcare), TCV_BcapMO1, Llcmp_NotApplicable, Llcmp_AnyOrOmit, C_RI_alternate, C_RI_alternate, Hlcmp_NotApplicable, Hlcmp_TS61))			Non EFR Mobile - generate SETUP
6		+Chmod(TCV_ChRate, TCV_ce, TCV_ur, srv)			initiate channel mode variables
7		(TCV_Null := OO_IFsetup_TS6x_MO(TCV_srv, TCV_ur, TCV_ce, TCV_ChRate))			ask operator to configure the MS
8		[TSPC_EFR_Speech_v2 OR TSPC_EFR_Speech_v3]			
9		(TCV_Setup_mo := Setup_11_3100(C_RI_alternate, Bcap_Speech_Efr_MO(C_Rchr_dontcare), TCV_BcapMO1, Llcmp_NotApplicable, Llcmp_AnyOrOmit, C_RI_alternate, C_RI_alternate, Hlcmp_NotApplicable, Hlcmp_TS61))			EFR Mobile - generate SETUP
10		+Chmod(TCV_ChRate, TCV_ce, TCV_ur, srv)			initiate channel mode variables
11		(TCV_Null := OO_IFsetup_TS6x_MO(TCV_srv, TCV_ur, TCV_ce, TCV_ChRate))			ask operator to configure the MS
		local_bcap			
12		[TCV_ce = C_transparent]			
13		[TCV_ur = C_9600bs]			
14		+local_TS61_FAX3_T(C_9600bs, C_ir_16kbs)			
15		[(TCV_ur = C_4800bs) OR (TCV_ur = C_2400bs)]			
16		+local_TS61_FAX3_T(TCV_ur, C_ir_8kbs)			
17	CS_0134	[(TCV_ur <> C_4800bs) AND(TCV_ur <> C_2400bs) AND(TCV_ur <> C_9600bs)]		I	IXIT parameter error
18		[TCV_ce = C_notransparent]			
19		+local_TS61_FAX3_NT(TCV_ur)			
		local_TS61_FAX3_T(ur:B_4; ir:B_2)			
20		(TCV_BcapMO1 := Bcap_MO(TCV_Rchr, C_FAX3, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, ir, C_parity_NA, C_transparent, C_modemt_none))			11.8.2.7.1
		local_TS61_FAX3_NT(ur:B_4)			
21		[ur=C_9600bs]			
22		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_FAX3, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, TCV_CE.NTorBothTorBothNT, C_modemt_none, C_X25_flct))			11.8.2.7.2
23		[(ur=C_2400bs) OR (ur = C_4800bs)]			
24		(TCV_BcapMO1 := BcapX_MO(TCV_Rchr, C_FAX3, C_SDUintegrity, C_nirr_dontcare, C_no_rate_adaption, C_sacp_NA,			11.8.2.7.2

		C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, TCV_CE.NTorBothTorBothNT, C_modemt_none, C_X25_flct))			
25	CS_0135	[(ur <> C_2400bs) AND(ur <> C_4800bs) AND(ur <> C_9600bs)]		I	IXIT parameter error
		local_selectCe			
26		[TSPX_TS_61_T_NT]			
27		(TCV_ce := C_notransparent)			
28		[NOT TSPX_TS_61_T_NT]			
29		(TCV_ce := TSPX_TS_61_ce)			

Detailed Comments:

The TS61 Bearer Capabilities and SETUP messages for MO is constructed according to section 11.8.2.7.1 and section 11.8.2.7.2 of GSM 11.10 and section B.1.10.2 and section B.2.10 of GSM 07.01.

The algorithm for selecting parameters:

```
IF (TSPX_BS_61_T_NT)
        TCV_ce := C_notransparent
ELSE
        TCV_ce := TSPX_BS_61_ce

use (TCV_ce, TSPX_BS_61_ur1, svc) for OO_IFsetup_TS6x_MO
```

Test Step Dynamic Behaviour

Test Step Name: TS62MO(srv:SERVICES)
Group: GSM_MS_CS_4220/Preambles/
Objective: To generate an MO SETUP message with appropriate Bcap for TS62 service.
Default: OtherEvents

Nr	Label	Behaviour Description	CRef	V	Comments
1		(TCV_srv := srv, TCV_ur := TSPX_TS_62_ur1, TCV_CE := NT_BothT_BothNTorT_BothT_BothNT)			variables used in InitCall
2		+local_selectCe			
3		+local_bcap			get the right BC
4		(TCV_Setup_mo := Setup_10_3100(TCV_BcapMO1, Llcmp_AnyOrOmit, Hlcmp_TS62))			
5		+Chmod(TCV_ChRate, TCV_ce, TCV_ur, srv)			
6		(TCV_Null := OO_IFsetup_TS6x_MO(TCV_srv, TCV_ur, TCV_ce, TCV_ChRate))			
		local_bcap			
7		[TCV_ce = C_transparent]			
8		[TCV_ur = C_9600bs]			
9		+local_TS62_FAX3_T(C_9600bs, C_ir_16kbs)			
10		[(TCV_ur = C_4800bs) OR (TCV_ur = C_2400bs)]			
11		+local_TS62_FAX3_T(TCV_ur, C_ir_8kbs)			
12	CS_0136	[(TCV_ur <> C_4800bs) AND(TCV_ur <> C_2400bs) AND(TCV_ur <> C_9600bs)]		I	IXIT parameter error
13		[TCV_ce = C_notransparent]			
14		+local_TS62_FAX3_NT(TCV_ur)			
		local_TS62_FAX3_T(ur:B_4; ir:B_2)			
15		(TCV_BcapMO1 := Bcap_MO(C_Rchr_dontcare, C_FAX3, C_Unstructured, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, ir, C_parity_NA, C_transparent, C_modemt_none))			11.8.2.7.1
		local_TS62_FAX3_NT(ur:B_4)			
16		[ur = C_9600bs]			
17		(TCV_BcapMO1 := BcapX_MO(C_Rchr_dontcare, C_FAX3, C_SDUintegrity, C_nirr_nomeaning, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, TCV_CE.NTorBothTorBothNT, C_modemt_none, C_X25_flct))			11.8.2.7.2
18		[(ur = C_2400bs) OR (ur = C_4800bs)]			
19		(TCV_BcapMO1 := BcapX_MO(C_Rchr_dontcare, C_FAX3, C_SDUintegrity, C_nirr_dontcare, C_no_rate_adaption, C_sacp_NA, C_Synchronous, C_stopbit_NA, C_databit_NA, ur, C_ir_16kbs, C_parity_NA, TCV_CE.NTorBothTorBothNT, C_modemt_none, C_X25_flct))			11.8.2.7.2
20	CS_0137	[(ur <> C_2400bs) AND(ur <> C_4800bs) AND(ur <> C_9600bs)]		I	IXIT parameter error
		local_selectCe			
21		[TSPX_TS_62_T_NT]			
22		(TCV_ce := C_notransparent)			
23		[NOT TSPX_TS_62_T_NT]			
24		(TCV_ce := TSPX_TS_62_ce)			

Detailed Comments: The TS62 Bearer Capabilities and SETUP messages for MO is constructed according to section 11.8.2.7.1 and section 11.8.2.7.2 of GSM 11.10 and section B.1.10.2 and section B.2.11 of GSM 07.01.

The algorithm for selecting parameters:
IF (TSPX_BS_62_T_NT)

```
        TCV_ce := C_notransparent
ELSE
        TCV_ce := TSPX_BS_62_ce

use (TCV_ce, TSPX_BS_62_ur1, svc) for OO_IFsetup_TS6x_MO
```

Defaults Library

Default Dynamic Behaviour

Default Name: OtherEvents
Group: GSM_MS_CS_4220/
Objective: To match unexpected events and sign final verdict for preambles.
Comments: used in preambles.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_02		4.
2		RETURN			
3		?TIMEOUT T_guard			1.
4		+ltree_SDCCH1_release			
5		+ltree_SDCCH2_release			
6		+ltree_TCH1_release			
7		+ltree_TCH2_release			
8		(TCV_Null := OO_TguardTimeOut())			
9	CS_0138	CANCEL		I	3.
10		L?OTHERWISE			2.
11		+ltree_SDCCH1_release			
12		+ltree_SDCCH2_release			
13		+ltree_TCH1_release			
14		+ltree_TCH2_release			
15		L?DL_RelIn	DLRelInd_01		
16	CS_0139	CANCEL		I	3.
17		?TIMEOUT T_guard			1.
18	CS_0140	CANCEL		I	3.
		ltree_SDCCH1_release			
19		[TCV_ch <> "dummy"]			
20		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
21		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
22		[TCV_ch1 <> "dummy"]			
23		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
24		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
25		[TCV_chTch <> "dummy"]			
26		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
27		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
28		[TCV_chTch1 <> "dummy"]			
29		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
30		[TCV_chTch1 = "dummy"]			

Detailed Comments:
1. The guard timer times out, inconclusive.
2. Unexpected events, inconclusive.
3. Cancel of all running timers.
4. Local end termination request allowed.

Default Dynamic Behaviour

Default Name: OtherEventsFail
Group: GSM_MS_CS_4220/
Objective: To match unexpected events and fail the test case.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_02		
2		RETURN			
3		?TIMEOUT T_guard			1.
4		+ltree_SDCCH1_release			
5		+ltree_SDCCH2_release			
6		+ltree_TCH1_release			
7		+ltree_TCH2_release			
8		(TCV_Null := OO_TguardTimeOut())			
9	CS_0141	CANCEL		I	3.
10		L?OTHERWISE			2.
11		+ltree_SDCCH1_release			
12		+ltree_SDCCH2_release			
13		+ltree_TCH1_release			
14		+ltree_TCH2_release			
15		L?DL_RelIn	DLRelInd_01		
16	CS_0142	CANCEL		F	3.
17		?TIMEOUT T_guard			
18	CS_0143	CANCEL		F	3.
		ltree_SDCCH1_release			
19		[TCV_ch <> "dummy"]			
20		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
21		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
22		[TCV_ch1 <> "dummy"]			
23		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
24		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
25		[TCV_chTch <> "dummy"]			
26		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
27		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
28		[TCV_chTch1 <> "dummy"]			
29		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
30		[TCV_chTch1 = "dummy"]			

Detailed Comments:
1. The guard timer times out, inconclusive.
2. Unexpected events, fail.
3. Cancel of all running timers.

Default Dynamic Behaviour

Default Name: OtherEventsFail_01
Group: GSM_MS_CS_4220/
Objective: To match irrelevant CHANNEL REQUEST msg and MEASUREMENT REPORT msg and return or match other unexpected events and fail the test case.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_01		6.
2		RETURN			
3		?TIMEOUT T_guard			1.
4		+ltree_SDCCH1_release			
5		+ltree_SDCCH2_release			
6		+ltree_TCH1_release			
7		+ltree_TCH2_release			
8		(TCV_Null := OO_TguardTimeOut())			
9	CS_0144	CANCEL		I	5.
10		L?DL_RacInChRq	ChReq(ChRequest_02)		2.
11		RETURN			
12		L?DL_UdatInMsrRpt	MsrRept(MsrReport_02)		3.
13		RETURN			
14		L?OTHERWISE			4.
15		+ltree_SDCCH1_release			
16		+ltree_SDCCH2_release			
17		+ltree_TCH1_release			
18		+ltree_TCH2_release			
19		L?DL_RelIn	DLRelInd_01		
20	CS_0145	CANCEL		F	5.
21		?TIMEOUT T_guard			
22	CS_0146	CANCEL		F	5.
		ltree_SDCCH1_release			
23		[TCV_ch <> "dummy"]			
24		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
25		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
26		[TCV_ch1 <> "dummy"]			
27		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
28		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
29		[TCV_chTch <> "dummy"]			
30		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
31		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
32		[TCV_chTch1 <> "dummy"]			
33		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
34		[TCV_chTch1 = "dummy"]			

Detailed Comments:
1. The guard timer times out, inconclusive.
2. To throw away any irrelevant channel request.
3. To throw away any measurement report.
4. Other unexpected events, fail.
5. Cancel of all running timers.
6. Cover also DLRelInd_02 for local end termination.

Default Dynamic Behaviour

Default Name: OtherEventsFail_02

Group: GSM_MS_CS_4220/

Objective: To match unexpected events and fail the test case but ignore channel request messages that are sent before the lower tester has sent (and the MS received) the Immediate Assigment message.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_02		
2		RETURN			
3		L?DL_RacInChRq	ChReq(ChRequest_02)		1.
4		RETURN			
5		?TIMEOUT T_guard			2.
6		+ltree_SDCCH1_release			
7		+ltree_SDCCH2_release			
8		+ltree_TCH1_release			
9		+ltree_TCH2_release			
10		(TCV_Null := OO_TguardTimeOut())			
11	CS_0147	CANCEL		I	4.
12		L?OTHERWISE			3.
13		+ltree_SDCCH1_release			
14		+ltree_SDCCH2_release			
15		+ltree_TCH1_release			
16		+ltree_TCH2_release			
17		L?DL_RelIn	DLRelInd_01		
18	CS_0148	CANCEL		F	4.
19		?TIMEOUT T_guard			
20	CS_0149	CANCEL		F	4.
		ltree_SDCCH1_release			
21		[TCV_ch <> "dummy"]			
22		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
23		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
24		[TCV_ch1 <> "dummy"]			
25		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
26		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
27		[TCV_chTch <> "dummy"]			
28		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
29		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
30		[TCV_chTch1 <> "dummy"]			
31		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
32		[TCV_chTch1 = "dummy"]			

Detailed Comments:

1. To throw away any irrelevant channel request.
2. The guard timer times out, inconclusive.
3. Unexpected events, fail.
4. Cancel of all running timers.

Default Dynamic Behaviour					
Default Name:	OtherEvents_01				
Group:	GSM_MS_CS_4220/				
Objective:	To match irrelevant messages and return				
Nr	**Label**	**Behaviour Description**	**CRef**	**V**	**Comments**
1		L?OTHERWISE			1.
2		RETURN			
Detailed Comments:	1. Cover also DLRelInd_02 for local end termination.				

Default Dynamic Behaviour

Default Name: OtherEvents_02

Group: GSM_MS_CS_4220/

Objective: To match unexpected events and sign final verdict for preambles but ignore channel request messages that are sent before the lower tester has sent (and the MS received) the Immediate Assigment message.

Comments: used in preambles.

Nr	Label	Behaviour Description	CRef	V	Comments
1		L?DL_RelIn	DLRelInd_02		
2		RETURN			
3		L?DL_RacInChRq	ChReq(ChRequest_02)		1.
4		RETURN			
5		?TIMEOUT T_guard			2.
6		+ltree_SDCCH1_release			
7		+ltree_SDCCH2_release			
8		+ltree_TCH1_release			
9		+ltree_TCH2_release			
10		(TCV_Null := OO_TguardTimeOut())			
11	CS_0150	CANCEL		I	4.
12		L?OTHERWISE			3.
13		+ltree_SDCCH1_release			
14		+ltree_SDCCH2_release			
15		+ltree_TCH1_release			
16		+ltree_TCH2_release			
17		L?DL_RelIn	DLRelInd_01		
18	CS_0151	CANCEL		I	4.
19		?TIMEOUT T_guard			2.
20	CS_0152	CANCEL		I	4.
		ltree_SDCCH1_release			
21		[TCV_ch <> "dummy"]			
22		L!DL_DatRqChRel	ChRel(TCV_ch, ChRelease_01)		
23		[TCV_ch = "dummy"]			
		ltree_SDCCH2_release			
24		[TCV_ch1 <> "dummy"]			
25		L!DL_DatRqChRel	ChRel(TCV_ch1, ChRelease_01)		
26		[TCV_ch1 = "dummy"]			
		ltree_TCH1_release			
27		[TCV_chTch <> "dummy"]			
28		L!DL_DatRqChRel	ChRel(TCV_chTch, ChRelease_01)		
29		[TCV_chTch = "dummy"]			
		ltree_TCH2_release			
30		[TCV_chTch1 <> "dummy"]			
31		L!DL_DatRqChRel	ChRel(TCV_chTch1, ChRelease_01)		
32		[TCV_chTch1 = "dummy"]			

Detailed Comments:

1. To throw away any irrelevant channel request.
2. The guard timer times out, inconclusive.
3. Unexpected events, inconclusive.
4. Cancel of all running timers.

History

Document history	
September 1996	First Edition
January 1997	Second Edition
May 1997	Third Edition
September 1997	Fourth Edition
January 1997	Fifth Edition
April 1998	Sixth Edition
May 1998	One-step Approval Procedure OAP 9841: 1998-05-20 To 1998-10-16 (Eighth Edition)
August 1998	Seventh Edition
October 1998	Eighth Edition

ISBN 2-7437-2612-1
Dépôt légal : Octobre 1998