Digital cellular telecommunications system (Phase 2+);
Mobile Application Part (MAP) Specification
(3GPP TS 09.02 version 5.18.0 Release 1996)
Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for ETSI members and non-members, and can be found in ETSI SR 000 314: "Intellectual Property Rights (IPRs): Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards", which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

All published ETSI deliverables shall include information which directs the reader to the above source of information.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.
4.1.3.3.2 Before location updating completion.................................................................39
4.1.3.3.3 After location updating completion................................................................39
4.1.3.4 VLR restoration.................................................................................................40
4.1.3.4.1 Inter-VLR information retrieval .................................................................40
4.1.3.4.2 HLR request ..............................................................................................40
4.1.3.5 The Interworking MSC (IWMSC) for Short Message Service .........................40
4.1.3.6 The Equipment Identity Register (EIR) ................................................................40
4.1.3.7 Summary table ...............................................................................................40
4.2 Use of TC ..............................................................................................................41
5 General on MAP services .......................................................................................43
5.1 Terminology and definitions ..................................................................................43
5.2 Modelling principles ............................................................................................43
5.3 Common MAP services .......................................................................................43
5.3.1 MAP-OPEN service .........................................................................................44
5.3.2 MAP-CLOSE service .......................................................................................47
5.3.3 MAP-DELIMITER service ................................................................................47
5.3.4 MAP-U-ABORT service ..................................................................................47
5.3.5 MAP-P-ABORT service ..................................................................................48
5.3.6 MAP-NOTICE service ....................................................................................49
5.4 Sequencing of services .......................................................................................49
5.5 General rules for mapping of services onto TC ....................................................52
5.5.1 Mapping of common services ..........................................................................52
5.5.2 Mapping of user specific services .....................................................................53
5.6 Definition of parameters .....................................................................................54
5.6.1 Common parameters .......................................................................................56
5.6.1.1 Invoke Id .......................................................................................................56
5.6.1.2 Linked Id .....................................................................................................56
5.6.1.3 Provider error ..............................................................................................56
5.6.1.4 User error ...................................................................................................56
5.6.2 Numbering and identification parameter .........................................................59
5.6.2.1 IMSI ..........................................................................................................59
5.6.2.2 TMSI ..........................................................................................................59
5.6.2.3 IMEI ..........................................................................................................59
5.6.2.4 Previous location area Id ............................................................................59
5.6.2.5 Stored location area Id ...............................................................................59
5.6.2.6 Current location area Id .............................................................................59
5.6.2.7 Target location area Id ...............................................................................59
5.6.2.8 Target cell Id ..............................................................................................59
5.6.2.9 [Spare] ..........................................................................................................59
5.6.2.10 Originating entity number .........................................................................59
5.6.2.11 MSC number ............................................................................................59
5.6.2.12 Target MSC number ..................................................................................59
5.6.2.13 HLR number ............................................................................................59
5.6.2.14 VLR number .............................................................................................59
5.6.2.15 HLR Id .....................................................................................................60
5.6.2.16 LMSI .........................................................................................................60
5.6.2.17 MS ISDN ...................................................................................................60
5.6.2.18 OMC Id .....................................................................................................60
5.6.2.19 Roaming number .......................................................................................60
5.6.2.20 [Spare] ......................................................................................................60
5.6.2.21 Handover number ......................................................................................60
5.6.2.22 Forwarded-to number ...............................................................................60
5.6.2.23 Forwarded-to subaddress .........................................................................60
5.6.2.24 Called number ..........................................................................................60
5.6.2.25 Calling number ..........................................................................................60
5.6.2.26 Originally dialled number .........................................................................60
5.6.2.27 Service centre address ..............................................................................60
5.6.2.28 Zone Code .................................................................................................60
5.6.2.29 MSIsdn-Alert ...........................................................................................61
5.6.2.30 Location Information ................................................................................61
5.6.2.31 GMSC Address ................................................................................................................. 61
5.6.2.32 VMSC Address .................................................................................................................... 61
5.6.2.33 Group Id ............................................................................................................................. 61
5.6.3 Subscriber management parameters ...................................................................................... 61
  5.6.3.1 Category ............................................................................................................................. 61
5.6.3.2 Equipment status .................................................................................................................. 61
5.6.3.3 Extensible Bearer service ..................................................................................................... 61
5.6.3.4 Extensible Teleservice ....................................................................................................... 61
5.6.3.5 Extensible Basic Service Group ......................................................................................... 61
5.6.3.6 GSM bearer capability ........................................................................................................ 61
5.6.3.7 Subscriber Status ............................................................................................................... 62
5.6.3.8 CUG Outgoing Access indicator ....................................................................................... 62
5.6.3.9 Operator Determined Barring General Data ...................................................................... 62
5.6.3.10 ODB HPLMN Specific Data .............................................................................................. 62
5.6.3.11 Regional Subscription Data .............................................................................................. 63
5.6.3.12 Regional Subscription Response ....................................................................................... 63
5.6.3.13 Roaming Restriction Due To Unsupported Feature .......................................................... 63
5.6.3.14 Extensible SS-Info .......................................................................................................... 63
5.6.3.15 Extensible Forwarding information .................................................................................. 63
5.6.3.16 Extensible Forwarding feature ........................................................................................... 63
5.6.3.17 Extensible SS-Status ........................................................................................................ 63
5.6.3.18 Extensible Forwarding Options ....................................................................................... 64
5.6.3.19 Extensible No reply condition timer ............................................................................... 64
5.6.3.20 Extensible Call barring information ............................................................................... 64
5.6.3.21 Extensible Call barring feature ....................................................................................... 64
5.6.3.22 CUG info ....................................................................................................................... 64
5.6.3.23 CUG subscription ........................................................................................................... 64
5.6.3.24 CUG interlock .................................................................................................................. 64
5.6.3.25 CUG index ..................................................................................................................... 64
5.6.3.26 CUG feature .................................................................................................................... 65
5.6.3.27 Inter CUG options .......................................................................................................... 65
5.6.3.28 Intra CUG restrictions ..................................................................................................... 65
5.6.3.29 Extensible SS-Data ......................................................................................................... 65
5.6.3.30 Subscriber State .............................................................................................................. 65
5.6.3.31 Requested Info ................................................................................................................ 65
5.6.3.32 Suppression of Announcement ....................................................................................... 66
5.6.3.33 Suppress T-CSI .............................................................................................................. 66
5.6.3.34 GMSC CAMEL Subscription Info .................................................................................. 66
5.6.3.35 VLR CAMEL Subscription Info ..................................................................................... 66
5.6.3.36 Supported CAMEL Phases ............................................................................................ 66
5.6.3.37 CUG Subscription Flag .................................................................................................. 66
5.6.3.38 CAMEL Subscription Info Withdraw ........................................................................... 66
5.6.3.39 Voice Group Call Service (VGCS) Data ........................................................................ 66
5.6.3.40 Voice Broadcast Service (VBS) Data ............................................................................ 66
5.6.4 Supplementary services parameters ...................................................................................... 66
  5.6.4.1 SS-Code .......................................................................................................................... 66
5.6.4.2 SS-Status .......................................................................................................................... 67
5.6.4.3 SS-Data ............................................................................................................................ 67
5.6.4.4 Override Category ............................................................................................................. 67
5.6.4.5 CLI Restriction Option ..................................................................................................... 67
5.6.4.6 Forwarding Options ........................................................................................................... 67
5.6.4.7 No reply condition timer .................................................................................................. 67
5.6.4.8 - 5.6.4.14 [Spare] ............................................................................................................. 68
5.6.4.15 Forwarding information .................................................................................................. 68
5.6.4.16 Forwarding feature .......................................................................................................... 68
5.6.4.17 [Spare] ........................................................................................................................... 68
5.6.4.18 Call barring information .................................................................................................. 68
5.6.4.19 Call barring feature ......................................................................................................... 68
5.6.4.20 New password ................................................................................................................ 68
5.6.4.21 Current password ............................................................................................................ 68
5.6.4.22 Guidance information ...................................................................................................... 69
5.6.4.23 [Spare] ........................................................................................................................... 69
5.6.4.24 SS-Info .............................................................................................................................................. 69
5.6.4.25 - 5.6.4.35 [Spare] .............................................................................................................................. 69
5.6.4.36 USSD Data Coding Scheme ............................................................................................................ 69
5.6.4.37 USSD String ................................................................................................................................ 69
5.6.4.38 Bearer service ................................................................................................................................ 69
5.6.4.39 Teleservice .................................................................................................................................... 69
5.6.4.40 Basic Service Group ...................................................................................................................... 70
5.6.4.41 eMLPP information .......................................................................................................................... 70
5.6.5 Call parameters .................................................................................................................................. 70
5.6.5.1 Call reference number ...................................................................................................................... 70
5.6.5.2 Interrogation type ............................................................................................................................. 70
5.6.5.3 OR interrogation ............................................................................................................................... 70
5.6.5.4 OR capability .................................................................................................................................... 70
5.6.5.5 Forwarding reason ........................................................................................................................... 70
5.6.5.6 Forwarding interrogation required ................................................................................................... 70
5.6.5.7 O-CSI ................................................................................................................................................ 70
5.6.6 Radio parameters ............................................................................................................................... 71
5.6.6.1 - 5.6.6.6 [Spare] ................................................................................................................................. 71
5.6.6.7 HO-Number Not Required ............................................................................................................. 71
5.6.7 Authentication parameters .................................................................................................................. 71
5.6.7.1 Authentication set list ....................................................................................................................... 71
5.6.7.2 Rand .................................................................................................................................................. 71
5.6.7.3 Sres ................................................................................................................................................... 71
5.6.7.4 Kc ....................................................................................................................................................... 71
5.6.7.5 [Spare] .............................................................................................................................................. 71
5.6.7.6 Cksn .................................................................................................................................................. 71
5.6.7.7 Ciphering mode ................................................................................................................................. 71
5.6.8 Short message parameters .................................................................................................................. 71
5.6.8.1 SM-RP-DA ......................................................................................................................................... 71
5.6.8.2 SM-RP-OA ....................................................................................................................................... 72
5.6.8.3 MWD status ...................................................................................................................................... 72
5.6.8.4 SM-RP-UI ......................................................................................................................................... 72
5.6.8.5 SM-RP-PRI ....................................................................................................................................... 72
5.6.8.6 SM Delivery Outcome ...................................................................................................................... 72
5.6.8.7 More Messages To Send ................................................................................................................... 72
5.6.8.8 Alert Reason ...................................................................................................................................... 72
5.6.8.9 Absent Subscriber Diagnostic SM .................................................................................................. 72
5.6.9 Access and signalling system related parameters ................................................................................ 73
5.6.9.1 BSS-apdu ......................................................................................................................................... 73
5.6.9.2 CM service type ............................................................................................................................... 73
5.6.9.3 Access connection status ................................................................................................................ 73
5.6.9.4 External Signal Information ............................................................................................................ 73
5.6.9.5 Access signalling information ......................................................................................................... 73
5.6.9.6 Location update type ....................................................................................................................... 73
5.6.9.7 Protocol ID ....................................................................................................................................... 73
5.6.9.8 Network signal information ............................................................................................................ 74
5.6.10 System operations parameters ......................................................................................................... 74
5.6.10.1 Network resources ......................................................................................................................... 74
5.6.10.2 Trace reference ............................................................................................................................... 75
5.6.10.3 Trace type ....................................................................................................................................... 75
5.7 Representation of a list of a basic parameter in service-primitives .......................................................... 75
6 Mobility services ...................................................................................................................................... 76
6.1 Location management services .............................................................................................................. 76
6.1.1 MAP_UPDATE_LOCATION_AREA service .................................................................................... 76
6.1.1.1 Definition ......................................................................................................................................... 76
6.1.1.2 Service primitives ............................................................................................................................ 76
6.1.1.3 Parameter definitions and use .......................................................................................................... 76
6.1.2 MAP_UPDATE_LOCATION service ................................................................................................. 77
6.1.2.1 Definition ......................................................................................................................................... 77
6.1.2.2 Service primitives ............................................................................................................................ 77
6.1.2.3 Parameter definitions and use .......................................................................................................... 77
6.1.3 MAP CANCEL_LOCATION service ................................................................. 78
6.1.3.1 Definition ................................................................................................. 78
6.1.3.2 Service primitives .................................................................................... 78
6.1.3.3 Parameter definitions and use ................................................................. 79
6.1.4 MAP SEND_IDENTIFICATION service ....................................................... 79
6.1.4.1 Definition ................................................................................................. 79
6.1.4.2 Service primitives .................................................................................... 79
6.1.4.3 Parameter definitions and use ................................................................. 79
6.1.5 MAP DETACH_IMSI service ........................................................................ 80
6.1.5.1 Definition ................................................................................................. 80
6.1.5.2 Service primitives .................................................................................... 80
6.1.5.3 Parameter definitions and use ................................................................. 80
6.1.6 MAP PURGE_MS service ........................................................................... 80
6.1.6.1 Definition ................................................................................................. 80
6.1.6.2 Service primitives .................................................................................... 81
6.1.6.3 Parameter definitions and use ................................................................. 81
6.2 Paging and search ......................................................................................... 81
6.2.1 MAP PAGE service .................................................................................... 81
6.2.1.1 Definition ................................................................................................. 81
6.2.1.2 Service primitives .................................................................................... 81
6.2.1.3 Parameter definitions and use ................................................................. 81
6.2.2 MAP SEARCH_FOR_MS service .............................................................. 82
6.2.2.1 Definition ................................................................................................. 82
6.2.2.2 Service primitives .................................................................................... 82
6.2.2.3 Parameter definitions and use ................................................................. 82
6.3 Access management services ....................................................................... 83
6.3.1 MAP PROCESS_ACCESS_REQUEST service ............................................. 83
6.3.1.1 Definition ................................................................................................. 83
6.3.1.2 Service primitives .................................................................................... 83
6.3.1.3 Parameter definitions and use ................................................................. 83
6.4 Handover services ......................................................................................... 85
6.4.1 MAP PREPARE_HANDBOVER service ...................................................... 85
6.4.1.1 Definition ................................................................................................. 85
6.4.1.2 Service primitives .................................................................................... 85
6.4.1.3 Parameter use .......................................................................................... 85
6.4.2 MAP SEND_END_SIGNAL service ......................................................... 86
6.4.2.1 Definition ................................................................................................. 86
6.4.2.2 Service primitives .................................................................................... 86
6.4.2.3 Parameter use .......................................................................................... 86
6.4.3 MAP PROCESS_ACCESS_SIGNALING service ....................................... 86
6.4.3.1 Definition ................................................................................................. 86
6.4.3.2 Service primitives .................................................................................... 86
6.4.3.3 Parameter use .......................................................................................... 87
6.4.4 MAP FORWARD_ACCESS_SIGNALING service .................................... 87
6.4.4.1 Definition ................................................................................................. 87
6.4.4.2 Service primitives .................................................................................... 87
6.4.4.3 Parameter use .......................................................................................... 87
6.4.5 MAP PREPARE_SUBSEQUENT_HANDBOVER service ....................... 87
6.4.5.1 Definition ................................................................................................. 87
6.4.5.2 Service primitives .................................................................................... 88
6.4.5.3 Parameter use .......................................................................................... 88
6.4.6 MAP_ALLOCATE_HANDBOVER_NUMBER service .............................. 88
6.4.6.1 Definition ................................................................................................. 88
6.4.6.2 Service primitives .................................................................................... 89
6.4.6.3 Parameter use .......................................................................................... 89
6.4.7 MAP SEND_HANDBOVER_REPORT service ....................................... 89
6.4.7.1 Definition ................................................................................................. 89
6.4.7.2 Service primitives .................................................................................... 89
6.4.7.3 Parameter use .......................................................................................... 89
6.5 Authentication management services ......................................................... 90
6.5.1 MAP AUTHENTICATE service ................................................................. 90
6.5.1.1 Definition ................................................................................................. 90
6.5.1.2 Service primitives .......................................................................................................... 90
6.5.1.3 Parameter use ................................................................................................................. 90
6.5.2 MAP_SEND_AUTHENTICATION_INFO service ................................................................. 90
6.5.2.1 Definition ....................................................................................................................... 90
6.5.2.2 Service primitives ............................................................................................................. 91
6.5.2.3 Parameter use ................................................................................................................. 91
6.6 Security management services ............................................................................................. 91
6.6.1 MAP_SET_CIPHERING_MODE service .............................................................................. 91
6.6.1.1 Definitions .................................................................................................................... 91
6.6.1.2 Service primitives ......................................................................................................... 92
6.6.1.3 Parameter use ................................................................................................................. 92
6.7 International mobile equipment identities management services ...................................... 92
6.7.1 MAP_CHECK_IMEI service ............................................................................................. 92
6.7.1.1 Definition ....................................................................................................................... 92
6.7.1.2 Service primitives ......................................................................................................... 92
6.7.1.3 Parameter use ................................................................................................................. 92
6.8 Subscriber management services ......................................................................................... 94
6.8.1 MAP_INSERT-SUBSCRIBER-DATA service .................................................................... 94
6.8.1.1 Definition ....................................................................................................................... 94
6.8.1.2 Service primitives ......................................................................................................... 94
6.8.1.3 Parameter use ................................................................................................................. 95
6.9 Identity management services .............................................................................................. 102
6.9.1 MAP_PROVIDE-IMSI service .......................................................................................... 102
6.9.1.1 Definition ....................................................................................................................... 102
6.9.1.2 Service primitives ......................................................................................................... 102
6.9.1.3 Parameter use ................................................................................................................. 102
6.9.2 MAP_OBTAIN_IMEI service ............................................................................................ 103
6.9.2.1 Definition ....................................................................................................................... 103
6.9.2.2 Service primitives ......................................................................................................... 103
6.9.2.3 Parameter use ................................................................................................................. 103
6.10 Fault recovery services ..................................................................................................... 103
6.10.1 MAP_RESET service ...................................................................................................... 103
6.10.1.1 Definition ....................................................................................................................... 103
6.10.1.2 Service primitives ......................................................................................................... 103
6.10.1.3 Parameter definition and use ....................................................................................... 103
6.10.2 MAP_FORWARD_CHECK_SS_INDICATION service .................................................... 104
6.10.2.1 Definition ....................................................................................................................... 104
6.10.2.2 Service primitives ......................................................................................................... 104
6.10.2.3 Parameter definition and use ....................................................................................... 104
6.10.3 MAP_RESTORE_DATA service ...................................................................................... 104
6.10.3.1 Definition ....................................................................................................................... 104
6.10.3.2 Service primitives ......................................................................................................... 104
6.10.3.3 Parameter definitions and use ....................................................................................... 105
6.11 Subscriber Information services ......................................................................................... 105
6.11.1 MAP_ANY-TIME-INTERROGATION service ................................................................. 105
6.11.1.1 Definition ....................................................................................................................... 105
6.11.1.2 Service primitives ......................................................................................................... 106
6.11.1.3 Parameter definition and use ....................................................................................... 106
6.11.2 MAP_PROVIDE-SUBSCRIBER-Info service ................................................................... 107
6.11.2.1 Definition ....................................................................................................................... 107
6.11.2.2 Service primitives ......................................................................................................... 107
6.11.2.3 Parameter definition and use ....................................................................................... 107
12.1 Dialogue establishment
12.1.1 Handling of unknown operations
12.1.2 Receipt of a MAP-OPEN request primitive
12.1.3 Receipt of a TC-BEGIN indication
12.1.4 Receipt of a MAP-OPEN response
12.1.5 Receipt of the first TC-CONTINUE ind
12.1.6 Receipt of a TC-END ind
12.1.7 Receipt of a TC-U-ABORT ind
12.1.8 Receipt of a TC-P-ABORT ind
12.2 Dialogue continuation
12.2.1 Sending entity
12.2.2 Receiving entity
12.3 Dialogue termination
12.3.1 Receipt of a MAP-CLOSE request
12.3.2 Receipt of a TC-END indication
12.4 User Abort
12.4.1 MAP-U-ABORT request
12.4.2 TC-U-ABORT ind
12.5 Provider Abort
12.5.1 MAP PM error situation
12.5.2 TC-P-ABORT ind
12.5.3 TC-U-ABORT ind
12.6 Procedures for MAP specific services
12.6.1 Service invocation
12.6.2 Service invocation receipt
12.6.3 Service response
12.6.4 Receipt of a response
12.6.4.1 Receipt of a TC-RESULT-NL indication
12.6.4.2 Receipt of a TC-RESULT-L indication
12.6.4.3 Receipt of a TC-U-ERROR indication
12.6.4.4 Receipt of a TC-INVOKE indication
12.6.4.5 Receipt of a TC-U-REJECT indication
12.6.4.6 Receipt of a TC-L-REJECT indication
12.6.4.7 Receipt of a TC-L-CANCEL indication
12.6.4.8 Receipt of a TC-NOTICE indication
12.6.5 Other events
12.6.5.1 Receipt of a TC-U-REJECT
12.6.5.2 Receipt of a TC-R-REJECT indication
12.6.5.3 Receipt of a TC-L-REJECT indication
12.6.6 Parameter checks
12.6.7 Returning state machines to idle
12.6.8 Load control
13 Mapping on to TC services
13.1 Dialogue control
13.1.1 Directly mapped parameters
13.1.2 Use of other parameters of dialogue handling primitives
13.1.2.1 Dialogue Id
13.1.2.2 Application-context-name
13.1.2.3 User information
13.1.2.4 Component present
13.1.2.5 Termination
13.1.2.6 P-Abort-Cause
13.1.2.7 Quality of service
13.2 Service specific procedures
13.2.1 Directly mapped parameters
13.2.2 Use of other parameters of component handling primitives
13.2.2.1 Dialogue Id
13.2.2.2 Class
13.2.2.3 Linked Id
13.2.2.4 Operation
13.2.2.5 Error
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>13.2.2.6</td>
<td>Parameters</td>
<td>157</td>
</tr>
<tr>
<td>13.2.2.7</td>
<td>Time out</td>
<td>157</td>
</tr>
<tr>
<td>13.2.2.8</td>
<td>Last component</td>
<td>157</td>
</tr>
<tr>
<td>13.2.2.9</td>
<td>Problem code</td>
<td>157</td>
</tr>
<tr>
<td>13.2.2.9.1</td>
<td>Mapping to MAP User Error</td>
<td>157</td>
</tr>
<tr>
<td>13.2.2.9.2</td>
<td>Mapping to MAP Provider Error parameter</td>
<td>157</td>
</tr>
<tr>
<td>13.2.2.9.3</td>
<td>Mapping to diagnostic parameter</td>
<td>158</td>
</tr>
</tbody>
</table>

**13.3** SDL descriptions | 158 |

**14** Abstract syntax of the MAP protocol | 184 |
| **14.1** General | 184 |
| **14.1.1** Encoding rules | 184 |
| **14.1.2** Use of TC | 184 |
| **14.1.2.1** Use of Global Operation and Error codes defined outside MAP | 185 |
| **14.1.3** Use of information elements defined outside MAP | 185 |
| **14.1.4** Compatibility considerations | 185 |
| **14.1.5** Structure of the Abstract Syntax of MAP | 186 |
| **14.1.6** Application Contexts | 188 |
| **14.2** Operation packages | 189 |
| **14.2.1** General aspects | 189 |
| **14.2.2** Packages specifications | 190 |
| **14.2.2.1** Location updating | 190 |
| **14.2.2.2** Location cancellation | 190 |
| **14.2.2.3** Roaming number enquiry | 190 |
| **14.2.2.4** Information retrieval | 190 |
| **14.2.2.5** Inter-VLR information retrieval | 190 |
| **14.2.2.6** IMSI retrieval | 191 |
| **14.2.2.7** Call control transfer | 191 |
| **14.2.2.8** - **14.2.2.9** [Spare] | 191 |
| **14.2.2.10** Interrogation | 191 |
| **14.2.2.11** [Spare] | 191 |
| **14.2.2.12** Handover Control | 191 |
| **14.2.2.13** Subscriber Data management stand alone | 192 |
| **14.2.2.14** Equipment management | 192 |
| **14.2.2.15** Subscriber data management | 192 |
| **14.2.2.16** Location register restart | 192 |
| **14.2.2.17** Tracing stand-alone | 192 |
| **14.2.2.18** Functional SS handling | 193 |
| **14.2.2.19** Tracing | 193 |
| **14.2.2.20** Binding | 193 |
| **14.2.2.21** Unstructured SS handling | 193 |
| **14.2.2.22** MO Short message relay services | 194 |
| **14.2.2.23** Short message gateway services | 194 |
| **14.2.2.24** MT Short message relay services | 194 |
| **14.2.2.25** [Spare] | 194 |
| **14.2.2.26** Message waiting data management | 194 |
| **14.2.2.27** Alerting | 195 |
| **14.2.2.28** Data restoration | 195 |
| **14.2.2.29** Purging | 195 |
| **14.2.2.30** Subscriber information enquiry | 195 |
| **14.2.2.31** Any time information enquiry | 196 |

**14.3** Application contexts | 197 |
| **14.3.1** General aspects | 197 |
| **14.3.2** Application context definitions | 198 |
| **14.3.2.1** [Spare] | 198 |
| **14.3.2.2** Location Updating | 198 |
| **14.3.2.3** Location Cancellation | 198 |
| **14.3.2.4** Roaming number enquiry | 198 |
| **14.3.2.5** [Spare] | 199 |
| **14.3.2.6** Location Information Retrieval | 199 |
| **14.3.2.7** Call control transfer | 199 |
| **14.3.2.8** - **14.3.2.10** [Spare] | 199 |
14.3.2.11 Location registers restart.................................................................................................................199
14.3.2.12 Handover control.................................................................................................................................199
14.3.2.13 IMSI Retrieval.........................................................................................................................................200
14.3.2.14 Equipment Management....................................................................................................................200
14.3.2.15 Information retrieval ..........................................................................................................................200
14.3.2.16 Inter-VLR information retrieval........................................................................................................200
14.3.2.17 Stand Alone Subscriber Data Management.......................................................................................200
14.3.2.18 Tracing ..................................................................................................................................................201
14.3.2.19 Network functional SS handling.........................................................................................................201
14.3.2.20 Network unstructured SS handling.....................................................................................................201
14.3.2.21 Short Message Gateway....................................................................................................................201
14.3.2.22 Mobile originating Short Message Relay..........................................................................................202
14.3.2.23 [Spare]................................................................................................................................................202
14.3.2.24 Short message alert............................................................................................................................202
14.3.2.25 Short message waiting data management.........................................................................................202
14.3.2.26 Mobile terminating Short Message Relay..........................................................................................202
14.3.2.27 MS purging .......................................................................................................................................203
14.3.2.28 Subscriber information enquiry........................................................................................................203
14.3.2.29 Any time information enquiry...........................................................................................................203
14.3.3 ASN.1 Module for application-context-names..........................................................................................204
14.4 MAP Dialogue Information........................................................................................................................205
14.5 MAP operation and error codes..................................................................................................................208
14.6 MAP operation and error types ..................................................................................................................213
14.6.1 Mobile Service Operations ....................................................................................................................213
14.6.2 Operation and Maintenance Operations..................................................................................................217
14.6.3 Call Handling Operations.........................................................................................................................219
14.6.4 Supplementary service operations........................................................................................................221
14.6.5 Short message service operations........................................................................................................225
14.6.6 Errors .....................................................................................................................................................228
14.7 MAP constants and data types....................................................................................................................233
14.7.1 Mobile Service data types........................................................................................................................233
14.7.2 Operation and maintenance data types....................................................................................................243
14.7.3 Call handling data types..........................................................................................................................244
14.7.4 Supplementary service data types...........................................................................................................247
14.7.5 Supplementary service codes................................................................................................................250
14.7.6 Short message data types.........................................................................................................................252
14.7.7 Error data types.......................................................................................................................................254
14.7.8 Common data types.................................................................................................................................258
14.7.9 Teleservice Codes...................................................................................................................................262
14.7.10 Bearer Service Codes............................................................................................................................264
14.7.11 Extension data types...............................................................................................................................266
15 General on MAP user procedures ..................................................................................................................267
15.1 Introduction..................................................................................................................................................267
15.2 Common aspects of user procedure descriptions......................................................................................267
15.2.1 General conventions................................................................................................................................267
15.2.2 Naming conventions................................................................................................................................267
15.2.3 Convention on primitives parameters....................................................................................................269
15.2.3.1 Open service......................................................................................................................................269
15.2.3.2 Close service......................................................................................................................................269
15.2.4 Version handling at dialogue establishment............................................................................................269
15.2.4.1 Behaviour at the initiating side........................................................................................................269
15.2.4.2 Behaviour at the responding side.......................................................................................................269
15.2.5 Abort Handling.......................................................................................................................................270
15.2.6 SDL conventions.....................................................................................................................................270
15.3 Interaction between MAP Provider and MAP Users..................................................................................270
16 Mobility procedures.........................................................................................................................................271
16.1 Location management Procedures...............................................................................................................271
16.1.1 Location updating....................................................................................................................................271
16.1.1.1 General................................................................................................................................................276
16.1.1.2 Detailed procedure in the MSC........................................................................................................280
## 19 Supplementary services procedures

### 19.0 Functional supplementary service processes

<table>
<thead>
<tr>
<th>19.0.1</th>
<th>Functional supplementary service process co-ordinator for MSC</th>
<th>417</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.0.2</td>
<td>Functional supplementary service process co-ordinator for VLR</td>
<td>419</td>
</tr>
<tr>
<td>19.0.3</td>
<td>Functional supplementary service process co-ordinator for HLR</td>
<td>422</td>
</tr>
</tbody>
</table>

### 19.1 Registration procedure

<table>
<thead>
<tr>
<th>19.1.1</th>
<th>General</th>
<th>425</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.1.2</td>
<td>Procedures in the MSC</td>
<td>426</td>
</tr>
<tr>
<td>19.1.3</td>
<td>Procedures in the VLR</td>
<td>428</td>
</tr>
<tr>
<td>19.1.4</td>
<td>Procedures in the HLR</td>
<td>431</td>
</tr>
</tbody>
</table>

### 19.2 Erasure procedure

<table>
<thead>
<tr>
<th>19.2.1</th>
<th>General</th>
<th>434</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.2.2</td>
<td>Procedures in the MSC</td>
<td>434</td>
</tr>
<tr>
<td>19.2.3</td>
<td>Procedures in the VLR</td>
<td>435</td>
</tr>
<tr>
<td>19.2.4</td>
<td>Procedures in the HLR</td>
<td>435</td>
</tr>
</tbody>
</table>

### 19.3 Activation procedure

<table>
<thead>
<tr>
<th>19.3.1</th>
<th>General</th>
<th>435</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.3.2</td>
<td>Procedures in the MSC</td>
<td>437</td>
</tr>
<tr>
<td>19.3.3</td>
<td>Procedures in the VLR</td>
<td>439</td>
</tr>
<tr>
<td>19.3.4</td>
<td>Procedures in the HLR</td>
<td>442</td>
</tr>
</tbody>
</table>

### 19.4 Deactivation procedure

<table>
<thead>
<tr>
<th>19.4.1</th>
<th>General</th>
<th>445</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.4.2</td>
<td>Procedures in the MSC</td>
<td>446</td>
</tr>
<tr>
<td>19.4.3</td>
<td>Procedures in the VLR</td>
<td>446</td>
</tr>
<tr>
<td>19.4.4</td>
<td>Procedures in the HLR</td>
<td>446</td>
</tr>
</tbody>
</table>

### 19.5 Interrogation procedure

<table>
<thead>
<tr>
<th>19.5.1</th>
<th>General</th>
<th>446</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.5.2</td>
<td>Procedures in the MSC</td>
<td>447</td>
</tr>
<tr>
<td>19.5.3</td>
<td>Procedures in the VLR</td>
<td>447</td>
</tr>
<tr>
<td>19.5.4</td>
<td>Procedures in the HLR</td>
<td>451</td>
</tr>
</tbody>
</table>

### 19.6 Invocation procedure

<table>
<thead>
<tr>
<th>19.6.1</th>
<th>General</th>
<th>453</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.6.2</td>
<td>Procedures in the MSC</td>
<td>453</td>
</tr>
<tr>
<td>19.6.3</td>
<td>Procedures in the VLR</td>
<td>454</td>
</tr>
</tbody>
</table>

### 19.7 Password registration procedure

<table>
<thead>
<tr>
<th>19.7.1</th>
<th>General</th>
<th>459</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.7.2</td>
<td>Procedures in the MSC</td>
<td>460</td>
</tr>
</tbody>
</table>
## Short message service procedures

<table>
<thead>
<tr>
<th>Clause</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.7.3</td>
<td>Procedures in the VLR</td>
</tr>
<tr>
<td>19.7.4</td>
<td>Procedures in the HLR</td>
</tr>
<tr>
<td>19.8</td>
<td>Mobile Initiated USSD procedure</td>
</tr>
<tr>
<td>19.8.1</td>
<td>General</td>
</tr>
<tr>
<td>19.8.2</td>
<td>Procedures in the MSC</td>
</tr>
<tr>
<td>19.8.3</td>
<td>Procedures in the VLR</td>
</tr>
<tr>
<td>19.8.4</td>
<td>Procedures in the HLR</td>
</tr>
<tr>
<td>19.9</td>
<td>Network initiated USSD procedure</td>
</tr>
<tr>
<td>19.9.1</td>
<td>General</td>
</tr>
<tr>
<td>19.9.2</td>
<td>Procedure in the MSC</td>
</tr>
<tr>
<td>19.9.3</td>
<td>Procedure in the VLR</td>
</tr>
<tr>
<td>19.9.4</td>
<td>Procedure in the HLR</td>
</tr>
<tr>
<td>19.10</td>
<td>Common macros for clause 19</td>
</tr>
<tr>
<td>19.10.1</td>
<td>SS Password handling macros</td>
</tr>
<tr>
<td>19.10.2</td>
<td>SS Error handling macros</td>
</tr>
</tbody>
</table>

## Short message service procedures

<table>
<thead>
<tr>
<th>Clause</th>
<th>Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>General</td>
</tr>
<tr>
<td>20.1.1</td>
<td>Mobile originated short message service Co-ordinator for the MSC</td>
</tr>
<tr>
<td>20.1.2</td>
<td>Short message Gateway Co-ordinator for the HLR</td>
</tr>
<tr>
<td>20.2</td>
<td>The mobile originated short message transfer procedure</td>
</tr>
<tr>
<td>20.2.1</td>
<td>Procedure in the servicing MSC</td>
</tr>
<tr>
<td>20.2.2</td>
<td>Procedure in the VLR</td>
</tr>
<tr>
<td>20.2.3</td>
<td>Procedure in the interworking MSC</td>
</tr>
<tr>
<td>20.3</td>
<td>The mobile terminated short message transfer procedure</td>
</tr>
<tr>
<td>20.3.1</td>
<td>Procedure in the Servicing MSC</td>
</tr>
<tr>
<td>20.3.2</td>
<td>Procedures in the VLR</td>
</tr>
<tr>
<td>20.3.3</td>
<td>Procedures in the HLR</td>
</tr>
<tr>
<td>20.3.4</td>
<td>Procedures in the gateway MSC</td>
</tr>
<tr>
<td>20.4</td>
<td>The Short Message Alert procedure</td>
</tr>
<tr>
<td>20.4.1</td>
<td>Procedures in the Servicing MSC</td>
</tr>
<tr>
<td>20.4.2</td>
<td>Procedures in the VLR</td>
</tr>
<tr>
<td>20.4.2.1</td>
<td>The Mobile Subscriber is present</td>
</tr>
<tr>
<td>20.4.2.2</td>
<td>The Mobile Equipment has memory available</td>
</tr>
<tr>
<td>20.4.2.3</td>
<td>Procedures in the HLR</td>
</tr>
<tr>
<td>20.4.2.4</td>
<td>Procedures in the Interworking MSC</td>
</tr>
<tr>
<td>20.5</td>
<td>The SM delivery status report procedure</td>
</tr>
<tr>
<td>20.5.1</td>
<td>Procedures in the HLR</td>
</tr>
<tr>
<td>20.5.2</td>
<td>Procedures in the gateway MSC</td>
</tr>
<tr>
<td>20.6</td>
<td>Common procedures for the short message clause</td>
</tr>
<tr>
<td>20.6.1</td>
<td>The macro Report_SM_Delivery_Stat_HLR</td>
</tr>
</tbody>
</table>

## General macro description

<table>
<thead>
<tr>
<th>Clause</th>
<th>Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>MAP open macros</td>
</tr>
<tr>
<td>21.1.1</td>
<td>Macro Receive_Open_Ind</td>
</tr>
<tr>
<td>21.1.2</td>
<td>Macro Receive_Open_Cnf</td>
</tr>
<tr>
<td>21.2</td>
<td>Macros to check the content of indication and confirmation primitives</td>
</tr>
<tr>
<td>21.2.1</td>
<td>Macro Check_Indication</td>
</tr>
<tr>
<td>21.2.2</td>
<td>Macro Check_Confirmation</td>
</tr>
<tr>
<td>21.3</td>
<td>The page and search macros</td>
</tr>
<tr>
<td>21.3.1</td>
<td>Macro PAGE_MSC</td>
</tr>
<tr>
<td>21.3.2</td>
<td>Macro Search_For_MS_MSC</td>
</tr>
<tr>
<td>21.4</td>
<td>Macros for handling an Access Request</td>
</tr>
<tr>
<td>21.4.1</td>
<td>Macro Process_Access_Request_MSC</td>
</tr>
<tr>
<td>21.4.2</td>
<td>Macro Process_Access_Request_VLR</td>
</tr>
<tr>
<td>21.4.3</td>
<td>Macro Identification Procedure</td>
</tr>
<tr>
<td>21.5</td>
<td>Authentication macros and processes</td>
</tr>
<tr>
<td>21.5.1</td>
<td>Macro Authenticate_MSC</td>
</tr>
<tr>
<td>21.5.2</td>
<td>Macro Authenticate_VLR</td>
</tr>
<tr>
<td>21.5.3</td>
<td>Process Obtain_Authentication_Sets_VLR</td>
</tr>
<tr>
<td>21.5.4</td>
<td>Macro Obtain_AuthentPara_VLR</td>
</tr>
<tr>
<td>21.5.5</td>
<td>Process Obtain_Auth_Sets_HLR</td>
</tr>
</tbody>
</table>
21.6 IMEI Handling Macros ......................................................................................................... 587
21.6.1 Macro Check_IMEI_MSC .............................................................................................. 587
21.6.2 Macro Check_IMEI_VLR .............................................................................................. 587
21.6.3 Process Check_IMEI_EIR .............................................................................................. 588
21.6.4 Macro Obtain_IMEI_MSC .............................................................................................. 588
21.6.5 Macro Obtain_IMEI_VLR .............................................................................................. 589
21.7 Insert Subscriber Data Macros .......................................................................................... 596
21.7.1 Macro Insert_Subs_Data_VLR ...................................................................................... 596
21.7.2 Process Insert_Subs_Data_Standalone_HLR ................................................................. 598
21.7.3 Macro Wait_for_Insert_Subs_Data_Cnf ........................................................................ 601
21.7.4 Process Send_Insert_Subs_Data .................................................................................... 603
21.8 Request IMSI Macros ....................................................................................................... 604
21.8.1 Macro Obtain_IMSI_MSC .............................................................................................. 604
21.8.2 Macro Obtain_IMSI_VLR .............................................................................................. 605
21.9 Tracing macros ............................................................................................................... 606
21.9.1 Macro Trace_Subscriber_Activity_MSC ....................................................................... 606
21.9.2 Macro Trace_Subscriber_Activity_VLR ....................................................................... 607
21.9.3 Macro Activate_Tracing_VLR ....................................................................................... 608
21.9.4 Macro Control_Tracing_HLR ....................................................................................... 609
21.10 Short Message Alert procedures .................................................................................... 611
21.10.1 Subscriber_Present_VLR process............................................................................... 611
21.10.2 Macro Alert_Service_Centre_HLR ............................................................................. 612

Annex A (informative): Cross-reference for abstract syntaxes of MAP ........................................ 614
Annex B (informative): Fully expanded ASN.1 sources for abstract syntaxes of MAP ............... 672
B.1 Fully Expanded ASN.1 Source of MAP-Protocol/TCAPMessages ....................................... 672
B.2 Fully Expanded ASN.1 Source of MAP-DialogueInformation ........................................... 710
Annex C: Void ........................................................................................................................ 714
Annex D (informative): Change Request History ................................................................. 715
History .......................................................................................................................................... 717
Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x  the first digit:
    1  presented to TSG for information;
    2  presented to TSG for approval;
    3  or greater indicates TSG approved document under change control.

y  the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z  the third digit is incremented when editorial only changes have been incorporated in the document.
1 Scope

It is necessary to transfer between entities of a Public Land Mobile Network (PLMN) information specific to the PLMN in order to deal with the specific behaviour of roaming Mobile Stations (MS)s. The Signalling System No. 7 specified by CCITT is used to transfer this information.

This European Telecommunication Standard (ETS) describes the requirements for the signalling system and the procedures needed at the application level in order to fulfil these signalling needs.

Clauses 1 to 4 are related to general aspects such as terminology, mobile network configuration and other protocols required by MAP.

MAP consists of a set of MAP services which are provided to MAP service-users by a MAP service-provider.

![Figure 1.1/1: Modelling principles](image)

Clauses 5 to 10 of this ETS describe the MAP services.

Clauses 11 to 14 define the MAP protocol specification and the behaviour of service provider (protocol elements to be used to provide MAP services, mapping on to TC service primitives, abstract syntaxes...).

Clauses 15 to 21 describe the MAP user procedures which make use of MAP services.

1.1 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] GSM 01.04: "Digital cellular telecommunications system (Phase 2+); Abbreviations and acronyms".

[2] GSM 02.01: "Digital cellular telecommunications system (Phase 2+); Principles of telecommunication services supported by a GSM Public Land Mobile Network (PLMN)".

[3] GSM 02.02: "Digital cellular telecommunications system (Phase 2+); Bearer Services (BS) Supported by a GSM Public Land Mobile Network (PLMN)".

[4] GSM 02.03: "Digital cellular telecommunications system (Phase 2+); Teleservices Supported by a GSM Public Land Mobile Network (PLMN)".

[5] GSM 02.04: "Digital cellular telecommunications system (Phase 2+); General on supplementary services".

[6] GSM 02.09: "Digital cellular telecommunications system; Security aspects".
[7] GSM 02.16: "Digital cellular telecommunications system; International Mobile station Equipment Identities (IMEI)".

[8] GSM 02.41: "Digital cellular telecommunications system (Phase 2+); Operator determined barring".

[9] GSM 02.81: "Digital cellular telecommunications system; Line identification supplementary services - Stage 1".

[10] GSM 02.82: "Digital cellular telecommunications system; Call Forwarding (CF) supplementary services - Stage 1".

[11] GSM 02.83: "Digital cellular telecommunications system; Call Waiting (CW) and Call Hold (HOLD) supplementary services - Stage 1".

[12] GSM 02.84: "Digital cellular telecommunications system; Multi Party (MPTY) supplementary services - Stage 1".

[13] GSM 02.85: "Digital cellular telecommunications system; Closed User Group (CUG) supplementary services - Stage 1".

[14] GSM 02.86: "Digital cellular telecommunications system; Advice of charge (AoC) supplementary services - Stage 1".

[15] GSM 02.88: "Digital cellular telecommunications system; Call Barring (CB) supplementary services - Stage 1".

[16] GSM 02.90: "Digital cellular telecommunication system (Phase 2+); Unstructured supplementary services operation - Stage 1".

[17] GSM 03.03 (ETS 300 927): "Digital cellular telecommunications system (Phase 2+); Numbering, addressing and identification".

[18] GSM 03.04: "Digital cellular telecommunications system; Signalling requirements relating to routing of calls to mobile subscribers".

[19] GSM 03.07: "Digital cellular telecommunications system (Phase 2+); Restoration procedures".

[20] GSM 03.08: "Digital cellular telecommunications system (Phase 2+); Organisation of subscriber data".

[21] GSM 03.09: "Digital cellular telecommunications system (Phase 2+; Handover procedures".

[22] GSM 03.11: "Digital cellular telecommunications system; Technical realization of supplementary services".

[23] GSM 03.12: "Digital cellular telecommunications system; Location registration procedures".

[24] GSM 03.20: "Digital cellular telecommunications system; Security related network functions".

[25] GSM 03.38: "Digital cellular telecommunications system (Phase 2+); Alphabets and language specific information for GSM".
[26] GSM 03.40: "Digital cellular telecommunications system (Phase 2+); Technical realization of the Short Message Service (SMS) Point to Point (PP)".

[27] GSM 03.81: "Digital cellular telecommunications system; Line identification supplementary services - Stage 2".

[28] GSM 03.82: "Digital cellular telecommunications system; Call Forwarding (CF) supplementary services - Stage 2".

[29] GSM 03.83: "Digital cellular telecommunications system; Call Waiting (CW) and Call Hold (HOLD) supplementary services - Stage 2".

[30] GSM 03.84: "Digital cellular telecommunications system; Multi Party (MPTY) supplementary services - Stage 2".

[31] GSM 03.85: "Digital cellular telecommunications system; Closed User Group (CUG) supplementary services - Stage 2".

[32] GSM 03.86: "Digital cellular telecommunications system; Advice of Charge (AoC) supplementary services - Stage 2".

[33] GSM 03.88: "Digital cellular telecommunications system; Call Barring (CB) supplementary services - Stage 2".

[34] GSM 03.90: "Digital cellular telecommunications system; Unstructured supplementary services operation - Stage 2".

[35] GSM 04.08: "Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 specification".

[36] GSM 04.10: "Digital cellular telecommunications system; Mobile radio interface layer 3 Supplementary services specification General aspects".

[37] GSM 04.11: "Digital cellular telecommunications system (Phase 2+); Point-to-Point (PP) Short Message Service (SMS) support on mobile radio interface".

[38] GSM 04.80: "Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 supplementary services specification Formats and coding".

[39] GSM 04.81: "Digital cellular telecommunications system; Line identification supplementary services - Stage 3".

[40] GSM 04.82: "Digital cellular telecommunications system; Call Forwarding (CF) supplementary services - Stage 3".

[41] GSM 04.83: "Digital cellular telecommunications system; Call Waiting (CW) and Call Hold (HOLD) supplementary services - Stage 3".

[42] GSM 04.84: "Digital cellular telecommunications system; Multi Party (MPTY) supplementary services - Stage 3".

[43] GSM 04.85: "Digital cellular telecommunications system; Closed User Group (CUG) supplementary services - Stage 3".
[44] GSM 04.86: "Digital cellular telecommunications system; Advice of Charge (AoC) supplementary services - Stage 3".

[45] GSM 04.88: "Digital cellular telecommunications system; Call Barring (CB) supplementary services - Stage 3".

[46] GSM 04.90: "Digital cellular telecommunications system; Unstructured supplementary services operation - Stage 3".

[47] GSM 08.02: "Digital cellular telecommunications system; Base Station System - Mobile-services Switching Centre (BSS - MSC) interface Interface principles".

[48] GSM 08.06: "Digital cellular telecommunications system; Signalling transport mechanism specification for the Base Station System - Mobile-services Switching Centre (BSS - MSC) interface".

[49] GSM 08.08: "Digital cellular telecommunications system (Phase 2+); Mobile Switching Centre - Base Station System (MSC - BSS) interface Layer 3 specification".

[49a] GSM 08.08: "Digital cellular telecommunications system (Phase 1); Mobile Switching Centre - Base Station System (MSC - BSS) interface Layer 3 specification".

[50] GSM 09.01: "Digital cellular telecommunications system; General network interworking scenarios".

[51] GSM 09.02: "Digital cellular telecommunications system (Phase 1); Mobile Application Part (MAP) specification".

[52] GSM 09.03: "Digital cellular telecommunications system; Signalling requirements on interworking between the Integrated Services Digital Network (ISDN) or Public Switched Telephone Network (PSTN) and the Public Land Mobile Network (PLMN)".

[53] GSM 09.04: "Digital cellular telecommunications system; Interworking between the Public Land Mobile Network (PLMN) and the Circuit Switched Public Data Network (CSPDN)".

[54] GSM 09.05: "Digital cellular telecommunications system; Interworking between the Public Land Mobile Network (PLMN) and the Packet Switched Public Data Network (PSPDN) for Packet Assembly/Disassembly facility (PAD) access".

[55] GSM 09.06: "Digital cellular telecommunications system (Phase 2+); Interworking between a Public Land Mobile Network (PLMN) and a Packet Switched Public Data Network/Integrated Services Digital Network (PSPDN/ISDN) for the support of packet switched data transmission services".

[56] GSM 09.07: "Digital cellular telecommunications system (Phase 2+); General requirements on interworking between the Public Land Mobile Network (PLMN) and the Integrated Services Digital Network (ISDN) or Public Switched Telephone Network (PSTN)".

[57] GSM 09.08: "Digital cellular telecommunications system; Application of the Base Station System Application Part (BSSAP) on the E-interface".

[58] GSM 09.10: "Digital cellular telecommunications system (Phase 2+); Information element mapping between Mobile Station - Base Station System and BSS - Mobile-services Switching Centre (MS - BSS - MSC) Signalling procedures and the Mobile Application Part (MAP)".
[59] GSM 09.11: "Digital cellular telecommunications system (Phase 2+); Signalling interworking for supplementary services".

[60] GSM 09.90: "Digital cellular telecommunications system (Phase 2+); Interworking between Phase 1 infrastructure and Phase 2 Mobile Stations (MS)".

[61] GSM 12.08: "Digital cellular telecommunications system (Phase 2); Subscriber and Equipment Trace".

[62] ETS 300 102-1 (1990): "Integrated Services Digital Network (ISDN); User-network interface layer 3 specifications for basic call control".

[63] ETS 300 136 (1992): "Integrated Services Digital Network (ISDN); Closed User Group (CUG) supplementary service description".

[64] ETS 300 138 (1992): "Integrated Services Digital Network (ISDN); Closed User Group (CUG) supplementary service Digital Subscriber Signalling System No.1 (DSS1) protocol".

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

[66] ETR 060: "Signalling Protocols and Switching (SPS); Guide-lines for using Abstract Syntax Notation One (ASN.1) in telecommunication application protocols".

[67] CCITT Recommendation E.164: "Numbering plan for the ISDN era".

[68] CCITT Recommendation E.212: "Identification plan for land mobile stations".

[69] CCITT Recommendation E.213: "Telephone and ISDN numbering plan for land mobile stations".

[70] CCITT Recommendation E.214: "Structuring of the land mobile global title for the signalling connection control part".

[71] CCITT Recommendation Q.669: "Interworking between the Digital Subscriber Signalling System Layer 3 protocol and the Signalling System No.7 ISDN User part".

[72] CCITT Recommendation Q.711: "Specifications of Signalling System No.7; Functional description of the signalling connection control part".

[73] CCITT Recommendation Q.712: "Definition and function of SCCP messages".

[74] CCITT Recommendation Q.713: "Specifications of Signalling System No.7; SCCP formats and codes".

[75] CCITT Recommendation Q.714: "Specifications of Signalling System No.7; Signalling connection control part procedures".

[76] CCITT Recommendation Q.716: "Specifications of Signalling System No.7; Signalling connection control part (SCCP) performances".

[77] CCITT Recommendation Q.721 (1988): "Specifications of Signalling System No.7; Functional description of the Signalling System No.7 Telephone user part".
CCITT Recommendation Q.722 (1988): "Specifications of Signalling System No.7; General function of Telephone messages and signals".

CCITT Recommendation Q.723 (1988): "Specifications of Signalling System No.7; Formats and codes".

CCITT Recommendation Q.724 (1988): "Specifications of Signalling System No.7; Signalling procedures".

CCITT Recommendation Q.725 (1988): "Specifications of Signalling System No.7; Signalling performance in the telephone application".

CCITT Recommendation Q.761 (1988): "Specifications of Signalling System No.7; Functional description of the ISDN user part of Signalling System No.7".

CCITT Recommendation Q.762 (1988): "Specifications of Signalling System No.7; General function of messages and signals".

CCITT Recommendation Q.763 (1988): "Specifications of Signalling System No.7; Formats and codes".

CCITT Recommendation Q.764 (1988): "Specifications of Signalling System No.7; Signalling procedures".

CCITT Recommendation Q.767: "Specifications of Signalling System No.7; Application of the ISDN user part of CCITT signalling System No.7 for international ISDN interconnections".

CCITT Recommendation Q.771: "Specifications of Signalling System No.7; Functional description of transaction capabilities".

CCITT Recommendation Q.772: "Specifications of Signalling System No.7; Transaction capabilities information element definitions".

CCITT Recommendation Q.773: "Specifications of Signalling System No.7; Transaction capabilities formats and encoding".

CCITT Recommendation Q.774: "Specifications of Signalling System No.7; Transaction capabilities procedures".

CCITT Recommendation Q.775: "Specifications of Signalling System No.7; Guide-lines for using transaction capabilities".

CCITT Recommendation X.200: "Reference Model of Open systems interconnection for CCITT Applications".


CCITT Recommendation X.210: "Open systems interconnection layer service definition conventions".
1.2 Abbreviations

Abbreviations used in this ETS are listed in GSM 01.04.

2 Configuration of the mobile network

2.1 The entities of the mobile system

To provide the mobile service as it is defined, it is necessary to introduce some specific functions. These functional entities can be implemented in different equipments or integrated. In any case, exchanges of data occur between these entities.

2.1.1 The Home Location Register (HLR)

This functional entity is a data base in charge of the management of mobile subscribers. A PLMN may contain one or several HLRs; it depends on the number of mobile subscribers, on the capacity of the equipment and on the organization of the network. All subscription data are stored there. The main information stored there concerns the location of each MS in order to be able to route calls to the mobile subscribers managed by each HLR. All management interventions occur on this data base. The HLRs have no direct control of MSCs.

Two numbers attached to each mobile subscription are stored in the HLR:

- IMSI;
- MSISDN.

The data base contains other information such as:

- location information (VLR number);
- basic telecommunication services subscription information;
- service restrictions (e.g. roaming limitation);
- supplementary services; the tables contain the parameters attached to these services.

The organization of the subscriber data is detailed in GSM 03.08.

2.1.2 The Visitor Location Register (VLR)

An MS roaming in an MSC area is controlled by the Visitor Location Register in charge of this area. When an MS appears in a location area it starts a location updating procedure. The MSC in charge of that area notices this registration and transfers to the Visitor Location Register the identity of the location area where the MS is situated. A VLR may be in charge of one or several MSC areas.
The VLR also contains the information needed to handle the calls set up or received by the MSs registered in its data base (in some cases the VLR may have to obtain additional information from the HLR); the following elements can be found in its tables:

- the IMSI;
- the MSISDN;
- the TMSI, if applicable;
- the location area where the MS has been registered. This will be used to call the station;
- supplementary service parameters.

The information is passed between VLR and HLR by the procedures described in GSM 03.12.

The organization of the subscriber data is detailed in GSM 03.08.

2.1.3 The Mobile-services Switching Centre (MSC)

The Mobile-services Switching Centre is an exchange which performs all the switching functions for MSs located in a geographical area designated as the MSC area. The main difference between an MSC and an exchange in a fixed network is that the MSC has to take into account the impact of the allocation of radio resources and the mobile nature of the subscribers and has to perform, for example, the following procedures:

- procedures required for the location registration (see GSM 03.12);
- procedures required for hand-over (see GSM 03.09).

2.1.4 The Base Station System (BSS)

The BSS is the sub-system of Base Station equipment (transceivers, controllers, etc...) which is viewed by the MSC through a single interface (A-interface) with the functionality described in GSM 08.02.

2.1.5 The Gateway MSC (GMSC)

In the case of incoming calls to the PLMN, if the fixed network is unable to interrogate the HLR, the call is routed to an MSC. This MSC will interrogate the appropriate HLR and then route the call to the MSC where the MS is located. The MSC which then performs the routing function to the actual location of the mobile is called the Gateway MSC.

The choice of which MSCs can act as Gateway MSCs is a network operator matter (e.g. all MSCs or some designated MSCs).

If the call is a voice group/broadcast call it is routed directly from the GMSC to the VBS/VGCS Anchor MSC, based on information (VBS/VGCS call reference) contained in the dialled number. See also GTSs 03.68 and 03.69.

See also GSM 03.04.

2.1.6 The SMS Gateway MSC

The SMS GMSC is the interface between the Mobile Network and the network which provides access to the Short Message Service Centre, for short messages to be delivered to MSs.

The choice of which MSCs can act as SMS Gateway MSCs is a network operator matter (e.g. all MSCs or some designated MSCs).

2.1.7 The SMS Interworking MSC

The SMS IWMSC is the interface between the Mobile Network and the network which provides access to the Short Message Service Centre, for short messages submitted by MSs.
The choice of which MSCs can act as SMS Interworking MSCs is a network operator matter (e.g. all MSCs or some designated MSCs).

2.1.8 The VBS/VGCS Anchor MSC

The voice broadcast/group call anchor MSC obtains from the associated GCR all relevant attributes and controls in turn all cells and dispatchers belonging to a given group call.

NOTE: Interactions between MSCs are not scope of this release of the specification; i.e. the "relay MSC function" which is needed to implement inter-MSC group calls will be specified in the next release of this standard.

2.1.9 The Equipment Identity Register (EIR)

This functional unit is a data base in charge of the management of the equipment identities of the MSs; see also GSM 02.16.

2.1.10 The GSM Service Control Function (gsmSCF)

This functional entity contains the CAMEL service logic to implement OSS. It interfaces with the gsmSSF and the HLR; see also TS GSM 03.78.

2.1.11 The Group Call Register (GCR)

This functional unit is a data base in charge of the management of attributes related to the establishment of Voice Broadcast Calls and Voice Group Calls.

2.2 Configuration of a Public Land Mobile Network (PLMN)

The basic configuration of a Public Land Mobile Network is presented in figure 2.2/1. In this figure the most general solution is described in order to define all the possible interfaces which can be found in any PLMN. The specific implementation in each network may be different: some particular functions may be implemented in the same equipment and then some interfaces may become internal interfaces. In any case the configuration of a PLMN must have no impact on the relationship with the other PLMNs.

In this configuration, all the functions are considered implemented in different equipments. Therefore, all the interfaces are external and need the support of the Mobile Application Part of the Signalling System No. 7 to exchange the data necessary to support the mobile service. From this configuration, all the possible PLMN organizations can be deduced.

2.3 Interconnection between PLMNs

Since the configuration of a PLMN does not have any impact on other PLMNs, the signalling interfaces specified can be implemented both between the entities within a PLMN and between different PLMNs.

2.4 The interfaces within the mobile service

2.4.1 Interface between the HLR and the VLR (D-interface)

This interface is used to exchange the data related to the location of the MS and to the management of the subscriber. The main service provided to the mobile subscriber is the capability to set up or to receive calls within the whole service area. To support that purpose the location registers have to exchange data. The VLR informs the HLR on the registration of a MS managed by the latter and provides it with the relevant location information. The HLR sends to the VLR all the data needed to support the service to the MS. The HLR then calls the previous VLR to inform it that it can cancel the location registration of this station because of the roaming of the mobile.
Exchanges of data may also occur when the mobile subscriber requires a particular service, when he wants to change some data attached to his subscription or when some parameters of the subscription are modified by administrative means.

2.4.2 Interface between the HLR and the gsmSCF (J-interface)

This interface is used by the gsmSCF to request information from the HLR. Support of the gsmSCF-HLR interface is a network operator option. As a network operator option, the HLR may refuse to provide the information requested by the gsmSCF.

2.4.3 Interface between the VLR and its associated MSC(s) (B-interface)

The VLR is the location and management data base for the MSs roaming in the area controlled by the associated MSC(s). Whenever the MSC needs data related to a given MS currently located in its area, it interrogates the VLR. When a MS initiates a location updating procedure with an MSC, the MSC informs its VLR which stores the relevant information in its tables. This procedure occurs whenever a mobile roams to another location area. Also, for instance when a subscriber activates a specific supplementary service or modifies some data attached to a service, the MSC transfers (via the VLR) the request to the HLR, which stores these modifications and updates the VLR if required.

However, this interface is not fully operational specified. It is strongly recommended not to implement the B-interface as an external interface.

2.4.4 Interface between VLRs (G-interface)

When an MS initiates a location updating using TMSI, the VLR can fetch the IMSI and authentication set from the previous VLR.

2.4.5 Interface between the HLR and the MSC (C-interface)

When the fixed network is not able to perform the interrogation procedure needed to set up a call to a mobile subscriber, the Gateway MSC has to interrogate the HLR of the called subscriber to obtain the roaming number of the called MS (see GSM 03.04).

To forward a short message to a mobile subscriber, the SMS Gateway MSC has to interrogate the HLR to obtain the MSC number where the MS is located.

2.4.6 Interface between MSCs (E-interface)

When a MS moves from one MSC area to another during a call, a handover procedure has to be performed in order to continue the communication. For that purpose the MSCs involved have to exchange data to initiate and then to realize the operation.

This interface is also used to forward short messages.

The application of this interface for inter-MSC VBS/VGCS calls will be specified in the next release of this standard.

2.4.7 Interface between the MSC and Base Station Systems (A-interface)

The description of this interface is contained in the GSM 08-series of MSs.

The BSS-MSC interface carries information concerning:

- BSS management;
- call handling;
- location management.
2.4.8 Interface between MSC and EIR (F-interface)

This interface is used when an MSC wants to check an IMEI.

2.4.9 Interface between VBS/VGCS Anchor MSC and GCR (I-interface)

This is an internal interface.

2.5 Splitting of the data storage

The data attached to each MS management, operation and location are stored in the Location Registers. Some data are duplicated in the HLR and in the VLR, but others may be stored only in one place.

A detailed description of the data organization can be found in GSM 03.08.
Figure 2.2/1: Configuration of a PLMN
3 Overload and compatibility overview

3.1 Overload control

There is a requirement for an overload/congestion control for all entities of the Public Land Mobile Network and the underlying Signalling System No. 7.

3.1.1 Overload control for MSC (outside MAP)

For the entity MSC the following two procedures (outside MAP) may be applied to control the processor load:

- ISDN
  CCIIT Recommendation Q.764 (Automatic Congestion Control), applicable to reduce the mobile terminating traffic;

- BSSAP
  GSM 08.08 (A-interface Flow Control), applicable to reduce the mobile originating traffic.

3.1.2 Overload control for MAP entities

For all MAP entities, especially the HLR, the following overload control method is applied:

If overload of a MAP entity is detected requests for certain MAP operations (see tables 3.1/1 and 3.1/2) may be ignored by the responder. The decision as to which MAP Operations may be ignored is made by the MAP service provider and is based upon the priority of the application context.

Since most of the affected MAP operations are supervised in the originating entity by TC timers (medium) an additional delay effect is achieved for the incoming traffic.

If overload levels are applicable in the Location Registers the MAP operations should be discarded taking into account the priority of their application context (see table 3.1/1 for HLR and table 3.1/2 for MSC/VLR; the lowest priority is discarded first).

The ranking of priorities given in the tables 3.1/1 and 3.1/2 is not normative. The tables can only be seen as a proposal which might be changed due to network operator/implementation matters.
<table>
<thead>
<tr>
<th>Priority high</th>
<th>Responder = HLR</th>
<th>Initiating Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mobility Management</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>networkLocUp</td>
<td>(updateLocation),</td>
<td>VLR</td>
</tr>
<tr>
<td></td>
<td>(restoreData/v2),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(sendParameters/v1)</td>
<td></td>
</tr>
<tr>
<td>infoRetrieval</td>
<td>(sendAuthenticationInfo/v2),</td>
<td>VLR</td>
</tr>
<tr>
<td></td>
<td>(sendParameters/v1)</td>
<td></td>
</tr>
<tr>
<td>msPurging</td>
<td>(purgeMS/v2)</td>
<td>VLR</td>
</tr>
<tr>
<td><strong>Short Message Service</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>shortMsgGateway</td>
<td>(sendRoutingInfoForSM),</td>
<td>GMSC</td>
</tr>
<tr>
<td></td>
<td>(reportSM-DeliveryStatus)</td>
<td></td>
</tr>
<tr>
<td>mwdMngt VLR</td>
<td>(readyForSM/v2),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(noteSubscriberPresent/v1)</td>
<td></td>
</tr>
<tr>
<td><strong>Mobile Terminating Traffic</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>locInfoRetrieval</td>
<td>(sendRoutingInfo)</td>
<td>GMSC</td>
</tr>
<tr>
<td>anyTimeEnquiry</td>
<td>(anyTimeInterrogation)</td>
<td>gsmSCF</td>
</tr>
<tr>
<td><strong>Subscriber Controlled Inputs (Supplementary Services)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>networkFunctionalSs</td>
<td>(registerSS),</td>
<td>VLR</td>
</tr>
<tr>
<td></td>
<td>(eraseSS),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(activateSS),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(deactivateSS),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(interrogateSS),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(registerPassword),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(processUnstructuredSS-Data/v1),</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(beginSubscriberActivity/v1)</td>
<td></td>
</tr>
<tr>
<td>networkUnstructuredSs</td>
<td>(processUnstructuredSS-Request/v2)</td>
<td>VLR</td>
</tr>
<tr>
<td>imsiRetrieval</td>
<td>(sendIMSI/v2)</td>
<td>VLR</td>
</tr>
</tbody>
</table>

**Priority low**

- **NOTE:** The application context name is the last component but one of the object identifier.
- Operation names are given in brackets for information with "/vn" appended to vn only operations.
Table 3.1/2: Priorities of Application Contexts for MSC/VLR as Responder

<table>
<thead>
<tr>
<th>Responder = MSC/VLR</th>
<th>Initiating Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Priority high</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Handover</strong></td>
<td></td>
</tr>
<tr>
<td>handoverControl</td>
<td>MSC</td>
</tr>
<tr>
<td>(prepareHandover/v2),</td>
<td></td>
</tr>
<tr>
<td>(performHandover/v1)</td>
<td></td>
</tr>
<tr>
<td><strong>Mobility and Location Register Management</strong></td>
<td></td>
</tr>
<tr>
<td>locationCancel</td>
<td>HLR</td>
</tr>
<tr>
<td>(cancelLocation)</td>
<td></td>
</tr>
<tr>
<td>reset</td>
<td>HLR</td>
</tr>
<tr>
<td>(reset)</td>
<td></td>
</tr>
<tr>
<td>interVlrInfoRetrieval</td>
<td>VLR</td>
</tr>
<tr>
<td>(sendIdentification/v2),</td>
<td></td>
</tr>
<tr>
<td>(sendParameters/v1)</td>
<td></td>
</tr>
<tr>
<td>subscriberDataMngt</td>
<td>HLR</td>
</tr>
<tr>
<td>(insertSubscriberData),</td>
<td></td>
</tr>
<tr>
<td>(deleteSubscriberData)</td>
<td></td>
</tr>
<tr>
<td>tracing</td>
<td>HLR</td>
</tr>
<tr>
<td>(activateTraceMode),</td>
<td></td>
</tr>
<tr>
<td>(deactivateTraceMode)</td>
<td></td>
</tr>
<tr>
<td><strong>Short Message Service</strong></td>
<td></td>
</tr>
<tr>
<td>shortMsgMO-Relay</td>
<td>MSC</td>
</tr>
<tr>
<td>(MO-ForwardSM v3)</td>
<td></td>
</tr>
<tr>
<td>(forwardSM v1/v2)</td>
<td>shortMsgMT-</td>
</tr>
<tr>
<td>(v1/v2)</td>
<td>(forwardSM</td>
</tr>
<tr>
<td></td>
<td>v1/v2)</td>
</tr>
<tr>
<td>shortMsgAlert</td>
<td>HLR</td>
</tr>
<tr>
<td>(alertServiceCentre/v2),</td>
<td></td>
</tr>
<tr>
<td>(alertServiceCentreWithoutResult/v1)</td>
<td></td>
</tr>
<tr>
<td><strong>Mobile Terminating Traffic</strong></td>
<td></td>
</tr>
<tr>
<td>roamingNbEnquiry</td>
<td>HLR</td>
</tr>
<tr>
<td>(provideRoamingNumber)</td>
<td></td>
</tr>
<tr>
<td>callControlTransfer</td>
<td>MSC</td>
</tr>
<tr>
<td>(resumeCallHandling)</td>
<td></td>
</tr>
<tr>
<td>subscriberInfoEnquiry</td>
<td>HLR</td>
</tr>
<tr>
<td>(provideSubscriberInformation)</td>
<td></td>
</tr>
<tr>
<td>networkUnstructuredSs</td>
<td>HLR</td>
</tr>
<tr>
<td>(unstructuredSS-Request/v2),</td>
<td></td>
</tr>
<tr>
<td>(unstructuredSS-Notify/v2)</td>
<td></td>
</tr>
<tr>
<td><strong>Priority low</strong></td>
<td></td>
</tr>
</tbody>
</table>

NOTE: The application context name is the last component but one of the object identifier. Operation names are given in brackets for information with "/vn" appended to vn only operations.
3.1.3 Congestion control for Signalling System No. 7

The requirements of SS7 Congestion control have to be taken into account as far as possible.

Means which could be applied to achieve the required traffic reductions are described in subclauses 3.1.1 and 3.1.2.

3.2 Compatibility

3.2.1 General

This ETS of the Mobile Application Part is designed in such a way that an implementation which conforms to it can also conform to the Mobile Application Part operational version 1 specifications, except on the MSC-VLR interface.

A version negotiation mechanism based on the use of an application-context-name is used to negotiate the protocol version used between two entities for supporting a MAP-user signalling procedure.

When starting a signalling procedure, the MAP-user supplies an application-context-name to the MAP-provider. This name refers to the set of application layer communication capabilities required for this dialogue. This refers to the required TC facilities (e.g. version 1 or 2) and the list of operation packages (i.e. set of operations) from which operations can be invoked during the dialogue.

A version one application-context-name may only be transferred to the peer user in a MAP-U-ABORT to an entity of version two or higher (i.e. to trigger a dialogue which involves only communication capabilities defined for MAP operational version 1).

If the proposed application-context-name can be supported by the responding entity the dialogue continues on this basis otherwise the dialogue is refused and the initiating user needs to start a new dialogue, which involves another application-context-name which requires less communication capabilities but provides similar functionalities (if possible).

When a signalling procedure can be supported by several application contexts which differ by their version number, the MAP-User needs to select a name. It can either select the name which corresponds to the highest version it supports or follow a more specific strategy so that the number of protocol fallbacks due to version compatibility problems be minimized.

3.2.2 Strategy for selecting the Application Context (AC) version

A method should be used to minimize the number of protocol fall-backs which would occur sometimes if the highest supported AC-Name were always the one selected by GSM entities when initiating a dialogue. The following method is an example which can be used mainly at transitory phase stage when the network is one of mixed phase entities.

3.2.2.1 Proposed method

A table (table 1) may be set up by administrative action to define the highest application context (AC) version supported by each destination; a destination may be another node within the same or a different PLMN, or another PLMN considered as a single entity. The destination may be defined by an E.164 number or an E.214 number derived from an IMSI. The table also includes the date when each destination is expected to be able to handle at least one AC of the latest version of the MAP protocol. When this date is reached, the application context supported by the node is marked as “unknown”, which will trigger the use of table 2.

A second table (table 2) contains an entry for each destination which has an entry in table 1. For a given entity, the entry in table 2 may be a single application context version or a vector of different versions applying to different application contexts for that entity. Table 2 is managed as described in subclause 3.2.2.2.

The data for each destination will go through the following states:

a) the version shown in table 1 is "version n-1", where ‘n’ is the highest version existing in this specification; table 2 is not used;

b) the version shown in table 1 is "unknown"; table 2 is used, and maintained as described in subclause 3.2.2.2;
c) when the PLMN operator declares that an entity (single node or entire PLMN) has been upgraded to support all the MAP version n ACs defined for the relevant interface, the version shown in table 1 is set to "version n" by administrative action; table 2 is no longer used, and the storage space may be recovered.

3.2.2.2 Managing the version look-up table

WHEN it receives a MAP-OPEN ind the MAP-User determines the originating entity number either using the originating address parameter or the originating reference parameter or retrieving it from the subscriber data using the IMSI or the MSISDN.

IF the entity number is known

THEN

It updates (if required) the associated list of highest supported ACs

ELSE

It creates an entry for this entity and includes the received AC-name in the list of highest supported ACs.

WHEN starting a procedure, the originating MAP-user looks up its version control table.

IF the destination address is known and not timed-out

THEN

It retrieves the appropriate AC-name and uses it

IF the dialogue is accepted by the peer

THEN

It does not modify the version control table

ELSE (this should never occur)

It starts a new dialogue with the common highest version supported (based on information implicitly or explicitly provided by the peer).

It replace the old AC-name by the new one in the list of associated highest AC supported.

ELSE

It uses the AC-name which corresponds to the highest version it supports.

IF the dialogue is accepted by the peer

THEN

It adds the destination node in its version control table and includes the AC-Name in the list of associated highest AC supported.

ELSE

It starts a new dialogue with the common highest version supported (based on information implicitly or explicitly provided by the peer).

IF the destination node was not known

THEN

It adds the destination node in its version control table and includes the new AC-Name in the list of associated highest AC supported.

ELSE

It replaces the old AC-name by the new one in the list of highest supported AC and reset the timer.
3.2.2.3 Optimizing the method

A table look-up may be avoided in some cases if both the HLR and the VLR stores for each subscriber the version of the AC-name used at location updating. Then:

- for procedures which make use of the same application-context, the same AC-name (thus the same version) can be selected (without any table look-up) when the procedure is triggered;

- for procedures which make use of a different application-context but which includes one of the packages used by the location updating AC, the same version can be selected (without any table look-up) when the procedure is triggered;

for HLR:

- Subscriber data modification (stand alone);

for VLR:

- Data Restoration.
4 Requirements concerning the use of SCCP and TC

4.1 Use of SCCP

The Mobile Application Part makes use of the services offered by the Signalling Connection Control Part of signalling System No. 7. CCITT Blue Book or ITU-T (03/93) Recommendations Q.711 to Q.716 should be consulted for the full specification of SCCP.

4.1.1 SCCP Class

MAP will only make use of the connectionless classes (0 or 1) of the SCCP.

4.1.2 Sub-System Number (SSN)

The Application Entities (AEs) defined for MAP consist of several Application Service Elements (ASEs) and are addressed by sub-system numbers (SSNs). The SSN for MAP are specified in GSM 03.03 [17].

4.1.3 SCCP addressing

4.1.3.1 Introduction

Within the GSM System there will be a need to communicate between entities within the same PLMN and in different PLMNs. Using the Mobile Application Part (MAP) for this function implies the use of Transaction Capabilities (TC) and the Signalling Connection Control Part (SCCP) of CCITT Signalling System No. 7.

Only the entities which should be addressed are described below. The format and coding of address parameters carried by the SCCP for that purpose shall comply with CCITT Recommendation Q.713 with the following restrictions:

1) Intra-PLMN addressing

For communication between entities within the same PLMN, a MAP SSN shall always be included in the called and calling party addresses. All other aspects of SCCP addressing are network specific.

2) Inter-PLMN addressing

a) Called Party Address

- SSN indicator = 1 (MAP SSN always included);
- Global title indicator = 0100 (Global title includes translation type, numbering plan, encoding scheme and nature of address indicator);
- the translation type field will be coded "00000000" (Not used);
- Routing indicator = 0 (Routing on global title);

b) Calling Party Address

- SSN indicator = 1 (MAP SSNs always included);
- Point code indicator = 0;
- Global title indicator = 0100 (Global title includes translation type, numbering plan, encoding scheme and nature of address indicator);
- the translation type field will be coded "00000000" (Not used);
- Routing indicator = 0 (Routing on Global Title).
If a Global Title translation is required for obtaining routing information, one of the numbering plans E.164, E.212 and E.214 is applicable.

- **E.212 numbering plan**

   An E.212 number must not be included as Global Title in an SCCP UNITDATA message. The translation of an E.212 number into a Mobile Global Title is applicable in a dialogue initiating VLR if the routing information towards the HLR is derived from the subscriber's IMSI. When an MS moves from one VLR service area to another, the new VLR may derive the address of the previous VLR from the Location Area Identification provided by the MS in the location registration request. The PLMN where the previous VLR is located is identified by the E.212 numbering plan elements of the Location Area Identification, i.e. the Mobile Country Code (MCC) and the Mobile Network Code (MNC).

- **E.214 and E.164 numbering plans**

   Only address information belonging to either E.214 or E.164 numbering plan is allowed to be included as Global Title in the Called and Calling Party Address.

   If the Calling Party Address associated with the dialogue initiating message contains a Global Title, the sending network entity shall include its E.164 entity number.

   When receiving an SCCP UNITDATA message, SCCP shall accept either of the valid numbering plans in the Called Party Address and in the Calling Party Address.

   When receiving an N-UNITDATA-REQUEST primitive from TC, SCCP shall accept an E.164 number or an E.214 number in the Called Address and in the Calling Address.

The following subclauses describe the method of SCCP addressing appropriate for each entity both for the simple intra-PLMN case and where an inter-PLMN communication is required. The following entities are considered:

- the Mobile-services Switching Centre (MSC);  
- the Home location Register (HLR);  
- the Visitor Location Register (VLR);  
- the Gateway Mobile-services Switching Centre (GMSC);  
- the Interworking Mobile-services Switching Centre (IWMSC).

### 4.1.3.2 The Mobile-services Switching Centre (MSC)

There are several cases where it is necessary to address the MSC.

#### 4.1.3.2.1 MSC interaction during handover

The address is derived from the target Cellid.

#### 4.1.3.2.2 MSC for short message routing

When a short message has to be routed to a MS, the GMSC addresses the VMSC by an MSC identity received from the HLR which complies with E.164 rules.

For MS originating short message, the IWMSC address is derived from the Service Centre address.

### 4.1.3.3 The Home Location Register (HLR)

There are several cases where the HLR has to be addressed:

#### 4.1.3.3.1 During call set-up

When a call is initiated the HLR of the called mobile subscriber will be interrogated to discover the whereabouts of the MS. The addressing required by the SCCP will be derived from the MSISDN dialled by the calling subscriber. The
dialled number will be translated into either an SPC, in the case of communications within a PLMN, or a Global Title if other networks are involved (i.e. if the communication is across a PLMN boundary).

If the calling subscriber is a fixed network subscriber, the interrogation can be initiated from the Gateway MSC of the home PLMN in the general case. If the topology of the network allows it, the interrogation could be initiated from any Signalling Point which has MAP capabilities, e.g. local exchange, outgoing International Switching Centre (ISC), etc.

4.1.3.3.2 Before location updating completion

When a MS registers for the first time in a VLR, the VLR has to initiate the update location dialogue with the MS's HLR and a preceding dialogue for authentication information retrieval if the authentication information must be retrieved from the HLR. When initiating either of these dialogues, the only data for addressing the HLR that the VLR has available is contained in the IMSI, and addressing information for SCCP must be derived from it. When continuing the established update location dialogue (as with any other dialogue), the VLR must derive the routing information towards the HLR from the Calling Party Address received with the first responding CONTINUE message until the dialogue terminating message is received. This means that the VLR must be able to address the HLR based:

- on an E.214 Mobile Global Title originally derived by the VLR from the IMSI; or
- on an E.164 HLR address; or
- in the case of intra-PLMN signalling, on an SPC.

When answering to the VLR the HLR should insert its E.164 address in the Calling Party Address of the SCCP message containing the first responding CONTINUE message.

NOTE: As PCS 1900 networks do not use E.214 addresses, the use of an E.214 address may lead to interworking problems between GSM and PCS 1900 networks.

If the HLR is in the same PLMN as the VLR, local translation tables may exist to derive an SPC. For authentication information retrieval and location updating via the international PSTN/ISDN signalling network, the Global title must be derived from the IMSI, using the principles contained in CCITT Recommendation E.214 and the Numbering Plan Indicator (NPI) value referenced by the SCCP Specifications. A summary of the translation from the IMSI (CCITT Recommendation E.212) to Mobile Global Title (described in CCITT Recommendation E.214) is shown below:

- E.212 Mobile Country Code translates to E.164 Country Code;
- E.212 Mobile Network Code translates to E.164 National Destination Code;
- E.212 Mobile Subscriber Identification Number (MSIN) is carried unchanged if within the E.164 number maximum length and terminated by the ST signal (15 digits + ST). If the Mobile Global Title is more than 15 digits the number is truncated to 15 by deleting the least significant digits.

This translation will be done either at the application or at SCCP level in the VLR. The Mobile Global Title thus derived will be used to address the HLR.

If location updating is triggered by an MS that roams from one MSC Area into a different MSC Area served by the same VLR, the VLR shall address the HLR in the same way as if the MS registers for the first time in the VLR.

4.1.3.3.3 After location updating completion

In this case, the subscriber's basic MSISDN has been received from the HLR during the subscriber data retrieval procedure as well as the HLR number constituting a parameter of the MAP message indicating successful completion of the update location dialogue. From either of these E.164 numbers the address information for initiating dialogues with the roaming subscriber's HLR can be derived. Also the subscriber's IMSI may be used for establishing the routing information towards the HLR. This may apply in particular if the dialogue with the HLR is triggered by subscriber controlled input.

Thus the SCCP address of the roaming subscriber's HLR may be an SPC, or it may be a Global title consisting of the E.164 MSISDN or the E.164 number allocated to the HLR or the E.214 Mobile Global Title derived from the IMSI.
4.1.3.3.4 VLR restoration

If a roaming number is requested by the HLR for an IMSI that has no data record in the interrogated VLR, the VLR provides the roaming number in the dialogue terminating message. Subsequently the VLR must retrieve the authentication data from the MS’s HLR, if required, and must then trigger the restore data procedure. For this purpose, the VLR has to initiate in succession two independent dialogues with the MS’s HLR. The MTP and SCCP address information needed for routing towards the HLR can be derived from the IMSI received as a parameter of the MAP message requesting the roaming number. In this case, the IMSI received from the HLR in the roaming number request shall be processed in the same way as the IMSI that is received from an MS that registers for the first time within a VLR. Alternatively to the IMSI, the Calling Party Address associated with the roaming number request may be used to obtain the routing information towards the HLR.

4.1.3.4 The Visitor Location Register (VLR)

There are several cases when the VLR needs to be addressed.

4.1.3.4.1 Inter-VLR information retrieval

When an MS moves from one VLR service area to another, the new VLR may request the IMSI and authentication sets from the previous VLR. The new VLR derives the address of the previous VLR from the Location Area Identification provided by the MS in the location registration request.

4.1.3.4.2 HLR request

The HLR will only request information from a VLR if it is aware that one of its subscribers is in the VLR’s service area. This means that a location updating dialogue initiated by the VLR has been successfully completed, i.e. the HLR has indicated successful completion of the update location procedure to the VLR.

When initiating dialogues towards the VLR after successful completion of location updating, the routing information used by the HLR is derived from the E.164 VLR number received as a parameter of the MAP message initiating the update location dialogue. If the VLR is in the same PLMN as the HLR, the VLR may be addressed directly by an SPC derived from the E.164 VLR number. For dialogues via the international PSTN/ISDN signaling network, presence of the E.164 VLR number in the Called Party Address is required.

4.1.3.5 The Interworking MSC (IWMSC) for Short Message Service

The IWMSC is the interface between the mobile network and the network to access to the Short Message Service Centre. This exchange has an E.164 address known in the HLR or in the MSC.

4.1.3.6 The Equipment Identity Register (EIR)

The EIR address is either unique or could be derived from the IMEI. The type of address is not defined.

4.1.3.7 Summary table

The following table summarizes the SCCP address used for invoke operations. As a principle, within a PLMN either an SPC or a GT may be used (network operation option), whereas when addressing an entity outside the PLMN the GT must be used. The address type mentioned in the table (e.g. MSISDN) is used as GT or to derive the SPC.

For a response, the originating address passed in the invoke is used. For extra-PLMN addressing the entity number is used as GT; for intra-PLMN addressing an SPC derived from the entity number may be used instead. When using an SPC, the SPC may be taken directly from MTP.
### Table 4.1/1

<table>
<thead>
<tr>
<th>to from</th>
<th>fixed network</th>
<th>HLR</th>
<th>VLR</th>
<th>MSC</th>
<th>EIR</th>
<th>gsmSCF</th>
</tr>
</thead>
<tbody>
<tr>
<td>fixed network</td>
<td>---</td>
<td>E:GT</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>---</td>
<td>T:MSISDN</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>home location register</td>
<td>---</td>
<td>---</td>
<td>I:SPC/MSISDN/HLR NUMBER (note)</td>
<td>---</td>
<td>---</td>
<td>I:SPC/MSISDN</td>
</tr>
<tr>
<td></td>
<td>---</td>
<td>---</td>
<td>I:SPC/MSISDN/HLR NUMBER (note)</td>
<td>---</td>
<td>---</td>
<td>I:SPC/MSISDN</td>
</tr>
<tr>
<td>visitor location register</td>
<td>---</td>
<td>I:SPC/MSISDN/HLR NUMBER (note)</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>mobile services switching centre</td>
<td>---</td>
<td>I:SPC/MSISDN/HLR NUMBER (note)</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>---</td>
<td>I:SPC/MSISDN/HLR NUMBER (note)</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>gsm Service Control Function</td>
<td>---</td>
<td>I:SPC/MSISDN/HLR NUMBER (note)</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>

I: Intra-PLMN.  
E: Extra(Inter)-PLMN.  
T: Address Type.  
GT: Global Title.  
MGT: E.214 Mobile Global Title.  
SPC: Signalling Point Code.

**NOTE:** For initiating the location updating procedure and an authentication information retrieval from the HLR preceding it, the VLR has to derive the HLR address from the IMSI of the MS. The result can be an SPC or an E.214 Mobile Global Title. When continuing the established update location dialogue (as with any other dialogue) the VLR must derive the routing information towards the HLR from the Calling Party Address received with the first responding CONTINUE message until the dialogue terminating message is received.

For transactions invoked by the VLR after update location completion, the VLR may derive the information for addressing the HLR from addresses received in the course of the update location procedure (MSISDN or HLR number) or from the IMSI.

When invoking the Restore Data procedure and an authentication information retrieval from the HLR preceding it, the VLR must derive the information for addressing the HLR from the address information received in association with the roaming number request. This may be either the IMSI received as a parameter of the MAP message requesting the Roaming Number or the Calling Party Address associated with the MAP message requesting the Roaming Number.

### 4.2 Use of TC

The Mobile Application part makes use of the services offered by the Transaction Capabilities (TC) of signalling system No. 7. ETS 300 287, which is based on CCITT White Book Recommendations Q.771 to Q.775, should be consulted for the full specification of TC.

The MAP uses all the services provided by TC except the ones related to the unstructured dialogue facility.

From a modelling perspective, the MAP is viewed as a single Application Service Element. Further structuring of it is for further study.

Transaction Capabilities refers to a protocol structure above the network layer interface (i.e, the SCCP service interface) up to the application layer including common application service elements but not the specific application service elements using them.

TC is structured as a Component sub-layer above a Transaction sub-layer.
The Component sub-layer provides two types of application services: services for the control of end-to-end dialogues and services for Remote Operation handling. These services are accessed using the TC-Dialogue handling primitives and TC-Component handling primitives respectively.

Services for dialogue control include the ability to exchange information related to application-context negotiation as well as initialization data.

Services for Remote Operation handling provide for the exchange of protocol data units invoking tasks (operations), and reporting their outcomes (results or errors) plus any non-application-specific protocol errors detected by the component sub-layer. The reporting of application-specific protocol errors by the TC user, as distinct from application process errors, is also provided. The Transaction sub-layer provides a simple end-to-end connection association service over which several related protocol data units (i.e. built by the Component Sub-Layer) can be exchanged. A Transaction termination can be prearranged (no indication provided to the TC user) or basic (indication provided).

Figure 4.2/1: Facilities for supporting the Mobile Application Part in Signalling System No.7
5 General on MAP services

5.1 Terminology and definitions

The term service is used in clauses 5 to 10 as defined in CCITT Recommendation X.200. The service definition conventions of CCITT Recommendation X.210 are also used.

5.2 Modelling principles

MAP provides its users with a specified set of services and can be viewed by its users as a "black box" or abstract machine representing the MAP service-provider. The service interface can then be depicted as shown in figure 5.2/1.

![Figure 5.2/1: Modelling principles](image)

The MAP service-users interact with the MAP service-provider by issuing or receiving MAP service-primitives at the service interface.

A MAP service-user may receive services from several instances of the MAP service-provider at the same time. In such cases the overall procedure is synchronised by the service-user.

The MAP service-primitives are named using the following notation:

```
MAP-ServicePrimitiveName type
```

where type can be any of: request (req), indication (ind), response (rsp) or confirm (cnf) (In the user arrow diagrams type is not indicated in the case of req/ind and indicated as "ack" in the case of rsp/cnf).

The services are further classified as unconfirmed-service, confirmed-service and provider-initiated-service where the first two categories refer to whether or not the service is confirmed by the service-provider. The confirmation may or may not correspond to a response provided by the other service-user.

MAP services are also classified as common MAP services which are available to all MAP service-users, and MAP service-user specific services which are services available to one or several, but not all, MAP service-users.

A MAP dialogue is defined as an exchange of information between two MAP users in order to perform a common task. A MAP dialogue will consist of one or several MAP services.

5.3 Common MAP services

All MAP service-users require access to services for performing basic application layer functions:

- for establishing and clearing MAP dialogues between peer MAP service-users;
- for accessing functions supported by layers below the applications layer;
- for reporting abnormal situations;
- for handling of different MAP versions;

```
MAP-ServicePrimitiveName type
```
For these purposes the following common services are defined:

- MAP-OPEN service;
- MAP-CLOSE service;
- MAP-DELIMITER service;
- MAP-U-ABORT service;
- MAP-P-ABORT service;
- MAP-NOTICE service.

In defining the service-primitives the following convention is used for categorising parameters:

M the inclusion of the parameter is mandatory. The M category can be used for any primitive type and specifies that the corresponding parameter must be present in the indicated primitive type;

O the inclusion of the parameter is a service-provider option. The O category can be used in indication and confirm type primitives and is used for parameters that may optionally be included by the service-provider;

U the inclusion of the parameter is a service-user option. The U category can be used in request and response type primitives. The inclusion of the corresponding parameter is the choice of the service-user;

C the inclusion of the parameter is conditional. The C category can be used for the following purposes:

- to indicate that if the parameter is received from another entity it must be included for the service being considered;
- to indicate that the service user must decide whether to include the parameter, based on the context on which the service is used;
- to indicate that one of a number of mutually exclusive parameters must be included (e.g. parameters indicating a positive result versus parameters indicating a negative result);
- to indicate that a service user optional parameter (marked "U") or a conditional parameter (marked "C") presented by the service user in a request or response type primitive is to be presented to the service user in the corresponding indication or confirm type primitive;

(=) when appended to one of the above, this symbol means that the parameter takes the same value as the parameter appearing immediately to its left;

blank the parameter is not present.

A primitive type may also be without parameters, i.e. no parameter is required with the primitive type; in this case the corresponding column of the table is empty.

### 5.3.1 MAP-OPEN service

This service is used for establishing a MAP dialogue between two MAP service-users. The service is a confirmed service with service primitives as shown in table 5.3.1.
Table 5.3/1: Service-primitives for the MAP-OPEN service

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application context name</td>
<td>M</td>
<td>M(=)</td>
<td>U</td>
<td>C(=)</td>
</tr>
<tr>
<td>Destination address</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination reference</td>
<td>U</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Originating address</td>
<td>U</td>
<td>O</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Originating reference</td>
<td>U</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific information</td>
<td>U</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responding address</td>
<td></td>
<td></td>
<td>U</td>
<td>C(=)</td>
</tr>
<tr>
<td>Result</td>
<td></td>
<td></td>
<td>U</td>
<td>C(=)</td>
</tr>
<tr>
<td>Refuse-reason</td>
<td></td>
<td></td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
</tbody>
</table>

Application context name:

This parameter identifies the type of application context being established. If the dialogue is accepted the received application context name shall be echoed. In case of refusal of dialogue this parameter shall indicate the highest version supported.

Destination address:

A valid SCCP address identifying the destination peer entity (see also clause 4). As an implementation option, this parameter may also, in the indication, be implicitly associated with the service access point at which the primitive is issued.

Destination-reference:

This parameter is a reference which refines the identification of the called process. It may be identical to Destination address but its value is to be carried at MAP level. Table 5.3/2 describes the MAP services using this parameter. Only these services are allowed to use it.

Table 5.3/2: Use of the destination reference

<table>
<thead>
<tr>
<th>MAP service</th>
<th>Reference type</th>
<th>Use of the parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP-REGISTER-SS</td>
<td>IMSI</td>
<td>Subscriber identity</td>
</tr>
<tr>
<td>MAP-ERASE-SS</td>
<td>IMSI</td>
<td>Subscriber identity</td>
</tr>
<tr>
<td>MAP-ACTIVATE-SS</td>
<td>IMSI</td>
<td>Subscriber identity</td>
</tr>
<tr>
<td>MAP-DEACTIVATE-SS</td>
<td>IMSI</td>
<td>Subscriber identity</td>
</tr>
<tr>
<td>MAP-INTERROGATE-SS</td>
<td>IMSI</td>
<td>Subscriber identity</td>
</tr>
<tr>
<td>MAP-REGISTER-PASSWORD</td>
<td>IMSI</td>
<td>Subscriber identity</td>
</tr>
<tr>
<td>MAP-PROCESS-UNSTRUCTURED-SS-REQUEST</td>
<td>IMSI</td>
<td>Subscriber identity</td>
</tr>
<tr>
<td>MAP-UNSTRUCTURED-SS-REQUEST</td>
<td>IMSI</td>
<td>Subscriber identity</td>
</tr>
<tr>
<td>MAP-UNSTRUCTURED-SS-NOTIFY</td>
<td>IMSI</td>
<td>Subscriber identity</td>
</tr>
<tr>
<td>MAP-FORWARD-SHORT-MESSAGE</td>
<td>IMSI (note)</td>
<td>Subscriber identity</td>
</tr>
</tbody>
</table>

NOTE: Only when the IMSI and the LMSI are received together from the HLR in the mobile terminated short message transfer.

Originating address:
A valid SCCP address identifying the requestor of a MAP dialogue (see also clause 4). As an implementation option, this parameter may also, in the request, be implicitly associated with the service access point at which the primitive is issued.

**Originating-reference:**

This parameter is a reference which refines the identification of the calling process. It may be identical to the Originating address but its value is to be carried at MAP level. Table 5.3/3 describes the MAP services using the parameter. Only these services are allowed to use it. Processing of the Originating-reference shall be performed according to the supplementary service descriptions and other service descriptions, e.g. operator determined barring.

**Table 5.3/3: Use of the originating reference**

<table>
<thead>
<tr>
<th>MAP service</th>
<th>Reference type</th>
<th>Use of the parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP-REGISTER-SS</td>
<td>ISDN-Address-String</td>
<td>Originated entity address</td>
</tr>
<tr>
<td>MAP-ERASE-SS</td>
<td>ISDN-Address-String</td>
<td>Originated entity address</td>
</tr>
<tr>
<td>MAP-ACTIVATE-SS</td>
<td>ISDN-Address-String</td>
<td>Originated entity address</td>
</tr>
<tr>
<td>MAP-DEACTIVATE-SS</td>
<td>ISDN-Address-String</td>
<td>Originated entity address</td>
</tr>
<tr>
<td>MAP-INTERROGATE-SS</td>
<td>ISDN-Address-String</td>
<td>Originated entity address</td>
</tr>
<tr>
<td>MAP-REGISTER-PASSWORD</td>
<td>ISDN-Address-String</td>
<td>Originated entity address</td>
</tr>
<tr>
<td>MAP-PROCESS-UNSTRUCTURED-SS-REQUEST</td>
<td>ISDN-Address-String</td>
<td>Originated entity address</td>
</tr>
</tbody>
</table>

**Specific information:**

This parameter may be used for passing any user specific information. Establishment and processing of the Specific information is not specified by GSM and shall be performed according to operator specific requirements.

**Responding address:**

An address identifying the responding entity. The responding address is included if required by the context (e.g. if it is different from the destination address).

**Result:**

This parameter indicates whether the dialogue is accepted by the peer.

**Refuse reason:**

This parameter is only present if the Result parameter indicates that the dialogue is refused. It takes one of the following values:

- Application-context-not-supported;
- Invalid-destination-reference;
- Invalid-originating-reference;
- No-reason-given;
- Remote node not reachable;
- Potential version incompatibility.
5.3.2 MAP-CLOSE service

This service is used for releasing a previously established MAP dialogue. The service may be invoked by either MAP service-user depending on rules defined within the service-user. The service is an unconfirmed service with parameters as shown in table 5.3/4.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Request</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release method</td>
<td>M</td>
<td>U</td>
</tr>
<tr>
<td>Specific Information</td>
<td></td>
<td>C(=)</td>
</tr>
</tbody>
</table>

Release method:
This parameter can take the following two values:
- normal release; in this case the primitive is mapped onto the protocol and sent to the peer;
- prearranged end; in this case the primitive is not mapped onto the protocol. Prearranged end is managed independently by the two users, i.e. only the request type primitive is required in this case.

Specific information:
This parameter may be used for passing any user specific information. Establishment and processing of the Specific information is not specified by GSM and shall be performed according to operator specific requirements.

5.3.3 MAP-DELIMITER service

This service is used to explicitly request the transfer of the MAP protocol data units to the peer entities.

See also subclause 5.4 and 5.5 for the detailed use of the MAP-DELIMITER service.

The service is an unconfirmed service with service-primitives as shown in table 5.3/5.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Request</th>
<th>Indication</th>
</tr>
</thead>
</table>

5.3.4 MAP-U-ABORT service

This service enables the service-user to request the MAP dialogue to be aborted. The service is an unconfirmed service with service-primitives as shown in table 5.3/6.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Request</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>User reason</td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>Diagnostic information</td>
<td>U</td>
<td>C(=)</td>
</tr>
<tr>
<td>Specific information</td>
<td>U</td>
<td>C(=)</td>
</tr>
</tbody>
</table>

User reason:
This parameter can take the following values:
- resource limitation (congestion);
  the requested user resource is unavailable due to congestion;
- resource unavailable;
the requested user resource is unavailable for reasons other than congestion;
- application procedure cancellation;
the procedure is cancelled for reason detailed in the diagnostic information parameter;
- procedure error;
processing of the procedure is terminated for procedural reasons.

Diagnostic information:
This parameter may be used to give additional information for some of the values of the user-reason parameter:

<table>
<thead>
<tr>
<th>User reason</th>
<th>Diagnostic information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource limitation (congestion)</td>
<td>- Short term/long term problem</td>
</tr>
<tr>
<td>Resource unavailable</td>
<td>Handover cancellation/</td>
</tr>
<tr>
<td>Application procedure cancellation</td>
<td>Radio Channel release/</td>
</tr>
<tr>
<td></td>
<td>Network path release/</td>
</tr>
<tr>
<td></td>
<td>Call release/</td>
</tr>
<tr>
<td></td>
<td>Associated procedure failure/</td>
</tr>
<tr>
<td></td>
<td>Tandem dialogue released/</td>
</tr>
<tr>
<td></td>
<td>Remote operations failure</td>
</tr>
<tr>
<td>Procedure error</td>
<td>-</td>
</tr>
</tbody>
</table>

Specific information:
This parameter may be used for passing any user specific information. Establishment and processing of the Specific information is not specified by GSM and shall be performed according to operator specific requirements.

5.3.5 MAP-P-ABORT service
This service enables the MAP service-provider to abort a MAP dialogue. The service is a provider-initiated service with service-primitive as shown in table 5.3/8.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider reason</td>
<td>M</td>
</tr>
<tr>
<td>Source</td>
<td>M</td>
</tr>
</tbody>
</table>

Provider reason:
This parameter indicates the reason for aborting the MAP dialogue:
- provider malfunction;
- supporting dialogue/transaction released;
- resource limitation;
- maintenance activity;
- version incompatibility;
- abnormal MAP dialogue.

Source:
This parameter indicates the source of the abort. For Transaction Capabilities (TC) applications the parameter may take the following values:

- MAP problem;
- TC problem;
- network service problem.

### Table 5.3/9: Values of provider reason and source parameters and examples of corresponding events

<table>
<thead>
<tr>
<th>Provider reason</th>
<th>Source</th>
<th>Corresponding event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provider malfunction</td>
<td>MAP</td>
<td>Malfunction at MAP level at peer entity</td>
</tr>
<tr>
<td></td>
<td>TC</td>
<td>&quot;Unrecognised message type&quot; or &quot;Badly formatted transaction portion&quot; or &quot;Incorrect transaction portion&quot; received in TC-P-ABORT &quot;Abnormal dialogue&quot;</td>
</tr>
<tr>
<td></td>
<td>Network service</td>
<td>Malfunction at network service level at peer entity</td>
</tr>
<tr>
<td>Supporting dialogue/transaction released</td>
<td>TC</td>
<td>&quot;Unrecognised transaction ID&quot; received in TC-ABORT</td>
</tr>
<tr>
<td>Resource limitation</td>
<td>MAP</td>
<td>Congestion towards MAP peer service-user</td>
</tr>
<tr>
<td>Maintenance activity</td>
<td>MAP</td>
<td>Maintenance at MAP peer service-user</td>
</tr>
<tr>
<td></td>
<td>Network service</td>
<td>Maintenance at network peer service level</td>
</tr>
<tr>
<td>Abnormal MAP dialogue</td>
<td>MAP</td>
<td>MAP dialogue is not in accordance with specified application context</td>
</tr>
<tr>
<td>Version incompatibility</td>
<td>TC</td>
<td>A Provider Abort indicating &quot;No common dialogue portion&quot; is received in the dialogue initiated state</td>
</tr>
</tbody>
</table>

### 5.3.6 MAP-NOTICE service

This service is used to notify the MAP service-user about protocol problems related to a MAP dialogue not affecting the state of the protocol machines.

The service is a provider-initiated service with service-primitive as shown in table 5.3/10.

### Table 5.3/10: Service-primitive for the MAP-NOTICE service

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem diagnostic</td>
<td>M</td>
</tr>
</tbody>
</table>

**Problem diagnostic:**

This parameter can take one of the following values:

- abnormal event detected by the peer;
- response rejected by the peer;
- abnormal event received from the peer
- message cannot be delivered to the peer.

### 5.4 Sequencing of services

The sequencing of services is shown in figure 5.4/1 and is as follows:
Opening:

The MAP-OPEN service is invoked before any user specific service-primitive is accepted. The sequence may contain none, one or several user specific service-primitives. If no user specific service-primitive is contained between the MAP-OPEN and the MAP-DELIMITER primitives, then this will correspond to sending an empty Begin message in TC. If more than one user specific service-primitive is included, all are to be sent in the same Begin message. The sequence ends with a MAP-DELIMITER primitive.
Continuing:

This sequence may not be present in some MAP dialogues. If it is present, it ends with a MAP-DELIMITER primitive. If more than one user specific service-primitive is included, all are to be included in the same Continue message.

Closing:

The sequence can only appear after an opening sequence or a continuing sequence. The sequence may contain none, one or several user specific service-primitives if the MAP-CLOSE primitive specifies normal release. If no user specific service-primitive is included, then this will correspond to sending an empty End message in TC. If more than one user specific service-primitive is included, all are to be sent in the same End message. If prearranged end is specified, the sequence cannot contain any user specific service-primitive. The MAP-CLOSE primitive must be sent after all user specific service-primitives have been delivered to the MAP service-provider.

Aborting:

A MAP service-user can issue a MAP-U-ABORT primitive at any time after the MAP dialogue has been opened or as a response to an attempt to open a MAP dialogue.

The MAP service-provider may issue at any time a MAP-P-ABORT primitive towards a MAP service-user for which a MAP dialogue exists.

MAP-U-ABORT primitives and MAP-P-ABORT primitives terminate the MAP dialogue.
d) Aborting

Figure 5.4/1: Sequencing of services

If the reason "resource unavailable (short term problem)" is indicated in the MAP-U-ABORT indication primitive, the MAP service-user may decide to attempt a new MAP dialogue establishment immediately.

Sequencing of user specific service-primitives is done by the MAP service-user and based on rules applicable for each MAP service-user instance.

A MAP-NOTICE indication primitive may be received at any time during the active period of a MAP dialogue.

5.5 General rules for mapping of services onto TC

5.5.1 Mapping of common services

Table 5.5/1 gives an overview of the mapping rules for mapping of common services onto TC-services. Table 5.5/2 gives the mapping rules for mapping of TC-services onto common services.

Protocol machine description is given in clauses 11 to 14.

<table>
<thead>
<tr>
<th>MAP service-primitive</th>
<th>TC service-primitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP-OPEN request</td>
<td>TC-BEGIN request</td>
</tr>
<tr>
<td>(+ any user specific service primitives)</td>
<td>(+ component handling primitives)</td>
</tr>
<tr>
<td>+ MAP-DELIMITER request</td>
<td></td>
</tr>
<tr>
<td>MAP-OPEN response</td>
<td>TC-CONTINUE request (note)</td>
</tr>
<tr>
<td>(+ any user specific service primitives)</td>
<td>(+ component handling primitives)</td>
</tr>
<tr>
<td>+ MAP-DELIMITER request</td>
<td></td>
</tr>
<tr>
<td>(any user specific service primitives)</td>
<td>TC-CONTINUE request</td>
</tr>
<tr>
<td>+ MAP-DELIMITER request</td>
<td>(+ component handling primitives)</td>
</tr>
<tr>
<td>(any user specific service primitives)</td>
<td>TC-END request</td>
</tr>
<tr>
<td>+ MAP-CLOSE request</td>
<td>(+ component handling primitives)</td>
</tr>
<tr>
<td>MAP-U-ABORT request</td>
<td>TC-U-ABORT request</td>
</tr>
</tbody>
</table>

NOTE: or TC-END if the MAP-CLOSE request has been received before the MAP-DELIMITER request.
### Table 5.5/2: Mapping of TC services on to common service

<table>
<thead>
<tr>
<th>TC service-primitive</th>
<th>MAP service-primitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC-BEGIN indication</td>
<td>MAP-OPEN indication</td>
</tr>
<tr>
<td>(+ component handling primitives)</td>
<td>(+ user specific service primitives)</td>
</tr>
<tr>
<td></td>
<td>+ MAP-DELIMITER indication (note 1)</td>
</tr>
<tr>
<td>TC-CONTINUE indication</td>
<td>First time:</td>
</tr>
<tr>
<td>(+ component handling primitives)</td>
<td>MAP-OPEN confirm</td>
</tr>
<tr>
<td></td>
<td>(+ user specific service primitives)</td>
</tr>
<tr>
<td></td>
<td>+ MAP-DELIMITER indication (note 1)</td>
</tr>
<tr>
<td></td>
<td>Subsequent times:</td>
</tr>
<tr>
<td></td>
<td>(user specific service primitives)</td>
</tr>
<tr>
<td></td>
<td>+ MAP-DELIMITER indication (note 1)</td>
</tr>
<tr>
<td>TC-END indication</td>
<td>MAP-OPEN confirm (note 6)</td>
</tr>
<tr>
<td>(+ component handling primitives)</td>
<td>(user specific service primitives)</td>
</tr>
<tr>
<td></td>
<td>+ MAP-CLOSE indication</td>
</tr>
<tr>
<td>TC-U-ABORT indication</td>
<td>MAP-U-ABORT indication or</td>
</tr>
<tr>
<td></td>
<td>MAP-P-ABORT indication (note 2)</td>
</tr>
<tr>
<td></td>
<td>MAP-OPEN confirmation (note 3)</td>
</tr>
<tr>
<td>TC-P-ABORT indication</td>
<td>MAP-P-ABORT indication (note 4)</td>
</tr>
<tr>
<td></td>
<td>MAP-OPEN confirmation (note 5)</td>
</tr>
</tbody>
</table>

NOTE 1: It may not be necessary to present this primitive to the user for MAP version 2 applications.

NOTE 2: The mapping depends on whether the TC-U-ABORT indication primitive contains a MAP-abort-PDU from the remote MAP service-provider or a MAP-user-abort-PDU from the remote MAP service-user.

NOTE 3: Only if the opening sequence is pending and if the "Abort Reason" in the TC-U-ABORT indication is set to "Application Context Not Supported".

NOTE 4: If the "Abort Reason" in the TC-P-ABORT indication is set to a value different from "Incorrect Transaction Portion".

NOTE 5: Only if the opening sequence is pending and if the "Abort Reason" in the TC-P-ABORT indication is set to "Incorrect Transaction Portion".

NOTE 6: Only if opening sequence is pending.

### 5.5.2 Mapping of user specific services

Table 5.5/3 gives the general mapping rules which apply to mapping of MAP user specific services onto TC services and table 5.5/4 gives the similar rules for mapping of TC services onto MAP user specific services. Detailed mapping is given in clauses 11 to 14.

### Table 5.5/3: Mapping of MAP user specific services onto TC services

<table>
<thead>
<tr>
<th>MAP service-primitive</th>
<th>TC-service-primitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP-xx request</td>
<td>TC-INVOLVE request</td>
</tr>
<tr>
<td>MAP-xx response</td>
<td>TC-RESULT-L request</td>
</tr>
<tr>
<td>(note 1)</td>
<td>TC-U-ERROR request</td>
</tr>
<tr>
<td></td>
<td>TC-U-REJECT request</td>
</tr>
<tr>
<td></td>
<td>TC-INVOLVE request (note 2)</td>
</tr>
</tbody>
</table>
Table 5.5/4: Mapping of TC services onto MAP user specific services

<table>
<thead>
<tr>
<th>TC-service-primitive</th>
<th>MAP service-primitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>TC-INVOKE indication</td>
<td>MAP-xx indication</td>
</tr>
<tr>
<td>TC-RESULT-L indication (note 4)</td>
<td>MAP-xx confirm</td>
</tr>
<tr>
<td>TC-U-ERROR indication</td>
<td></td>
</tr>
<tr>
<td>TC-INVOKE indication (note 2)</td>
<td></td>
</tr>
<tr>
<td>TC-L-CANCEL indication</td>
<td></td>
</tr>
<tr>
<td>TC-U-REJECT indication</td>
<td>MAP-xx confirm or</td>
</tr>
<tr>
<td>TC-L-REJECT indication</td>
<td>MAP-NOTICE indication (note 3)</td>
</tr>
<tr>
<td>TC-R-REJECT indication</td>
<td></td>
</tr>
</tbody>
</table>

Notes to tables 5.10 and 5.11:

NOTE 1: The mapping is determined by parameters contained in the MAP-xx response primitive.

NOTE 2: This applies only to TC class 4 operations where the operation is used to pass a result of another class 2 or class 4 operation.

NOTE 3: The detailed mapping rules are given in clause 13.

NOTE 4: If RESULT-NL components are present they are mapped on to the same MAP-xx confirm.

5.6 Definition of parameters

Following is an alphabetic list of parameters used in the common MAP-services in subclause 5.3:

- Application context name
- Refuse reason
- 5.3.1
- 5.3.1
- Destination address
- Release method
- 5.3.1
- 5.3.2
- Destination reference
- Responding address
- 5.3.1
- 5.3.1
- Diagnostic information
- Result
- 5.3.4
- 5.3.1
- Originating address
- Source
- 5.3.1
- 5.3.5
- Originating reference
- Specific information
- 5.3.1
- 5.3.1/5.3.2/5.3.4
- Problem diagnostic
- User reason
- 5.3.6
- 5.3.4
- Provider reason
- 5.3.5
Following is an alphabetic list of parameters contained in this clause:

Access connection status 5.6.9.3
Access signalling information 5.6.9.5
Alert Reason 5.6.8.8
Authentication set list 5.6.7.1
Basic Service Group 5.6.4.40
Bearer service 5.6.4.38
BSS-apdu 5.6.9.1
Call barring feature 5.6.4.19
Call barring information 5.6.4.18
Call reference 5.6.5.1
Called number 5.6.2.24
Calling number 5.6.2.25
CAMEL Subscription Info Withdraw Category 5.6.3.38
Ciphering mode 5.6.7.7
Ckan 5.6.7.5
CLI Restriction 5.6.4.5
CM service type 5.6.3.2
CUG feature 5.6.3.26
CUG index 5.6.3.25
CUG info 5.6.3.22
CUG interlock 5.6.3.24
CUG Outgoing Access indicator 5.6.3.8
CUG subscription 5.6.3.23
CUG Subscription Flag 5.6.3.37
Current location area Id 5.6.2.6
Current password 5.6.4.21
eMLPP Information 5.6.4.41
Equipment status 5.6.3.2
Extensible Basic Service Group 5.6.3.5
Extensible Bearer service 5.6.3.3
Extensible Call barring feature 5.6.3.21
Extensible Call barring information 5.6.3.20
Extensible Forwarding feature 5.6.3.16
Extensible Forwarding info 5.6.3.15
Extensible Forwarding Options 5.6.3.18
Extensible No reply condition timer 5.6.3.19
Extensible SS-Data 5.6.3.29
Extensible SS-Info 5.6.3.14
Extensible SS-Status 5.6.3.17
Extensible Teleservice 5.6.3.4
External Signal Information 5.6.9.4
Forwarded-to number 5.6.2.22
Forwarded-to subaddress 5.6.2.23
Forwarding feature 5.6.4.16
Forwarding information 5.6.4.15
Forwarding Options 5.6.4.6
GMSC CAMEL Subscription Info 5.6.3.34
Group Id 5.6.2.33
GSM bearer capability 5.6.3.6
Guidance information 5.6.4.22
Handover number 5.6.2.21
HLR Id 5.6.2.15
HLR number 5.6.2.13
HO-Number Not Required 5.6.6.7
IMEI 5.6.2.3
IMSI 5.6.2.1
Inter CUG options 5.6.3.27
Intra CUG restrictions 5.6.3.28
Invoke Id 5.6.1.1
Kc 5.6.7.4
Linked Id 5.6.1.2
Location Information 5.6.2.30
Location update type 5.6.9.6
More Messages To Send 5.6.8.7
MS ISDN 5.6.2.17
MSC number 5.6.2.11
MSIsdn-Alert 5.6.2.29
MWD status 5.6.8.3
Network resources 5.6.10.1
Network signal information 5.6.9.8
New password 5.6.4.20
No reply condition timer 5.6.4.7
ODB General Data 5.6.3.9
ODB HPLMN Specific Data 5.6.3.10
OMC Id 5.6.2.18
Originally dialed number 5.6.2.26
Originating entity number 5.6.2.10
Override Category 5.6.4.4
Previous location area Id 5.6.2.4
Protocol Id 5.6.9.7
Provider error 5.6.1.3
Rand 5.6.7.2
Regional Subscription Data 5.6.3.11
Regional Subscription Response 5.6.3.12
Requested Info 5.6.3.31
Roaming number 5.6.2.19
Roaming Restriction Due To Unsupported Feature 5.6.3.13
Service centre address 5.6.2.27
SM Delivery Outcome 5.6.8.6
SM-RP-DA 5.6.8.1
SM-RP-OA 5.6.8.2
SM-RP-PI 5.6.8.5
SM-RP-UI 5.6.8.4
Sres 5.6.7.3
SS-Code 5.6.4.1
SS-Data 5.6.4.3
SS-Info 5.6.4.24
SS-Status 5.6.4.2
Stored location area Id 5.6.2.5
Subscriber State 5.6.3.30
Subscriber Status 5.6.3.7
Supported CAMEL Phases 5.6.3.36
Suppress T-CSI 5.6.3.33
Suppression of Announcement 5.6.3.32
Target cell Id 5.6.2.8
Target location area Id 5.6.2.7
Target MSC number 5.6.2.12
Teleservice 5.6.4.39
TMSI 5.6.2.2
Trace reference 5.6.10.2
Trace type 5.6.10.3
User error 5.6.1.4
USSD Data Coding Scheme 5.6.4.36
USSD String 5.6.4.37
VBS Data 5.6.3.40
VGCS Data 5.6.3.39
VLR CAMEL Subscription Info 5.6.3.35
VLR number 5.6.2.14
Zone Code 5.6.2.28
5.6.1 Common parameters

The following set of parameters are used in several MAP service-primitives.

5.6.1.1 Invoke Id

This parameter identifies corresponding service primitives. The parameter is supplied by the MAP service-user and must be unique over each service-user/service-provider interface.

5.6.1.2 Linked Id

This parameter is used for linked services and it takes the value of the invoke Id of the service linked to.

5.6.1.3 Provider error

This parameter is used to indicate a protocol related type of error:

- duplicated invoke Id;
- not supported service;
- mistyped parameter;
- resource limitation;
- initiating release, i.e. the peer has already initiated release of the dialogue and the service has to be released;
- unexpected response from the peer;
- service completion failure;
- no response from the peer;
- invalid response received.

5.6.1.4 User error

This parameter can take values as follows:

NOTE: The values are grouped in order to improve readability; the grouping has no other significance.

a) Generic error:

- system failure, i.e. a task cannot be performed because of a problem in another entity. The type of entity or network resource may be indicated by use of the network resource parameter;
- data missing, i.e. an optional parameter required by the context is missing;
- unexpected data value, i.e. the data type is formally correct but its value or presence is unexpected in the current context;
- resource limitation;
- initiating release, i.e. the receiving entity has started the release procedure;
- facility not supported, i.e. the requested facility is not supported by the PLMN.
b) Identification or numbering problem:
- unknown subscriber, i.e. no such subscription exists;
- number changed, i.e. the subscription does not exist for that number any more;
- unknown MSC;
- unidentified subscriber, i.e. if the subscriber is not contained in the database and it has not or cannot be established whether or not a subscription exists;
- unallocated roaming number;
- unknown equipment;
- unknown location area.

c) Subscription problem:
- roaming not allowed, i.e. a location updating attempt is made in an area not covered by the subscription;
- illegal subscriber, i.e. illegality of the access has been established by use of authentication procedure;
- bearer service not provisioned;
- teleservice not provisioned;
- illegal equipment, i.e. the IMEI check procedure has shown that the IMEI is blacklisted or not whitelisted.

d) Handover problem:
- no handover number available;
- subsequent handover failure, i.e. handover to a third MSC failed for some reason.

e) Operation and maintenance problem:
- tracing buffer full, i.e. tracing cannot be performed because the tracing capacity is exceeded.

f) Call set-up problem:
- no roaming number available, i.e. a roaming number cannot be allocated because all available numbers are in use;
- absent subscriber, i.e. the subscriber has activated the detach service or the system detects the absence condition.;
- busy subscriber;
- no subscriber reply;
- forwarding violation, i.e. the call has already been forwarded the maximum number of times that is allowed;
- CUG reject, i.e. the call does not pass a CUG check; additional information may also be given in order to indicate rejection due to e.g. incoming call barred or non-CUG membership.
- call barred. Optionally, additional information may be included for indicating either that the call meets a barring condition set by the subscriber or that the call is barred for operator reasons.
- optimal routeing not allowed, i.e. the entity which sends the error does not support optimal routeing, or the HLR will not accept an optimal routeing interrogation from the GMSC, or the call cannot be optimally routed because it would contravene optimal routeing constraints.
- forwarding failed, i.e. the GMSC interrogated the HLR for forwarding information but the HLR returned an error.


g) Supplementary services problem:
- call barred;
- illegal SS operation;
- SS error status;
- SS not available;
- SS subscription violation;
- SS incompatibility;
- negative password check;
- password registration failure;
- Number of Password Attempts;
- USSD Busy;
- Unknown Alphabet.

For definition of these errors see GSM 04.80.

h) Short message problem:

- SM delivery failure with detailed reason as follows:
  - memory capacity exceeded;
  - MS protocol error;
  - MS not equipped;
  - unknown service centre (SC);
  - SC congestion;
  - invalid SME address;
  - subscriber is not an SC subscriber;
  - and possibly detailed diagnostic information, coded as specified in TS GSM 03.40, under SMS-SUBMIT-REPORT and SMS-DELIVERY-REPORT. If the SM entity which returns the SM Delivery Failure error includes detailed diagnostic information, it shall be forwarded in the MAP_MO_FORWARD_SHORT_MESSAGE and in the MAP_MT_FORWARD_SHORT_MESSAGE response.

- message waiting list full, i.e. no further SC address can be added to the message waiting list;

- Subscriber busy for MT SMS, i.e. the mobile terminated short message transfer cannot be completed because:
  - another mobile terminated short message transfer is going on and the delivery node does not support message buffering, or
  - another mobile terminated short message transfer is going on and it is not possible to buffer the message for later delivery, or
  - the message was buffered but it is not possible to deliver the message before the expiry of the buffering time defined in GSM 03.40.

- Absent Subscriber SM, i.e. the mobile terminated short message transfer cannot be completed because the network cannot contact the subscriber. Diagnostic information regarding the reason for the subscriber’s absence may be included with this error.
5.6.2 Numbering and identification parameter

5.6.2.1 IMSI
This parameter is the International Mobile Subscriber Identity defined in GSM 03.03.

5.6.2.2 TMSI
This parameter is the Temporary Mobile Subscriber Identity defined in GSM 03.03.

5.6.2.3 IMEI
This parameter is the International Mobile Equipment Identity defined in GSM 03.03.

5.6.2.4 Previous location area Id
This parameter refers to the identity of the location area from which the subscriber has roamed.

5.6.2.5 Stored location area Id
This parameter refers to the location area where the subscriber is assumed to be located.

5.6.2.6 Current location area Id
This parameter is used to indicate the location area in which the subscriber is currently located.

5.6.2.7 Target location area Id
This parameter refers to the location area into which the subscriber intends to roam.

5.6.2.8 Target cell Id
This parameter refers to the identity of the cell to which a call has to be handed over.

5.6.2.9 [Spare]

5.6.2.10 Originating entity number
This parameter refers to an application layer identification of a system component in terms of its associated ISDN number.

5.6.2.11 MSC number
This parameter refers to the ISDN number of an MSC.

5.6.2.12 Target MSC number
This parameter refers to the ISDN number of an MSC to which a call has to be handed over.

5.6.2.13 HLR number
This parameter refers to the ISDN number of an HLR.

5.6.2.14 VLR number
This parameter refers to the ISDN number of a VLR.
5.6.2.15   HLR Id
This parameter refers to the identity of an HLR derived from the IMSI defined in CCITT Recommendation E.212.

5.6.2.16   LMSI
This parameter refers to a local identity allocated by the VLR to a given subscriber for internal management of data in the VLR.

5.6.2.17   MS ISDN
This parameter refers to one of the ISDN numbers assigned to a mobile subscriber in accordance with CCITT Recommendation E.213.

5.6.2.18   OMC Id
This parameter refers to the identity of an operation and maintenance centre.

5.6.2.19   Roaming number
This parameter refers to the roaming number as defined in CCITT Recommendation E.213.

5.6.2.20   [Spare]

5.6.2.21   Handover number
This parameter refers to the number used for routing a call between MSCs during handover.

5.6.2.22   Forwarded-to number
This parameter refers to the address to which a call is to be forwarded. This may include a subaddress.

5.6.2.23   Forwarded-to subaddress
This parameter refers to the sub-address attached to the address to which a call is to be forwarded.

5.6.2.24   Called number
This parameter refers to a called party number as defined in CCITT Recommendation Q.767.

5.6.2.25   Calling number
This parameter refers to a calling party number as defined in CCITT Recommendation Q.767.

5.6.2.26   Originally dialled number
This parameter refers to the number dialled by the calling party in order to reach a mobile subscriber.

5.6.2.27   Service centre address
This parameter represents the address of a Short Message Service Centre.

5.6.2.28   Zone Code
This parameter is used to define location areas into which the subscriber is allowed or not allowed to roam (regional subscription). With a complete list of Zone Codes the VLR is able to determine for all its location areas whether roaming is allowed or not.
5.6.2.29 MSIsdn-Alert
This parameter refers to the MSISDN stored in a Message Waiting Data File in the HLR. It is used to alert the Service Centre when the MS is again attainable.

5.6.2.30 Location Information
This parameter indicates the location of the served subscriber as defined in GSM 03.18.

5.6.2.31 GMSC Address
This parameter refers to the E.164 address of a GMSC.

5.6.2.32 VMSC Address
This parameter refers to the E.164 address of a VMSC.

5.6.2.33 Group Id
This parameter is used to describe groups a subscriber can be member of. A subscriber can partake in all group calls (VBS/VGCS) where he subscribed to the respective groups.

5.6.3 Subscriber management parameters

5.6.3.1 Category
This parameter refers to the calling party category as defined in CCITT Recommendation Q.767.

5.6.3.2 Equipment status
This parameter refers to the status of the mobile equipment as defined in GSM 02.16.

5.6.3.3 Extensible Bearer service
This parameter may refer to a single bearer service, a set of bearer services or to all bearer services as defined in TS GSM 02.02. This parameter is used only for subscriber profile management. Extensible Bearer service values include all values defined for a Bearer service parameter (5.6.4.38).

5.6.3.4 Extensible Teleservice
This parameter may refer to a single teleservice, a set of teleservices or to all teleservices as defined in TS GSM 02.03. This parameter is used only for subscriber profile management. Extensible Teleservice values include all values defined for a Teleservice parameter (5.6.4.39).

5.6.3.5 Extensible Basic Service Group
This parameter refers to the Basic Service Group either as an extensible bearer service (see subclause 5.6.3.3) or an extensible teleservice (see subclause 5.6.3.4). This parameter is used only for subscriber profile management. The null value (i.e. neither extensible bearer service nor extensible teleservice) is used to denote the group containing all extensible bearer services and all extensible teleservices.

5.6.3.6 GSM bearer capability
This parameter refers to the GSM bearer capability information element defined in GSM 04.08.
5.6.3.7 Subscriber Status
This parameter refers to the barring status of the subscriber:
- service granted;
- Operator Determined Barring.

5.6.3.8 CUG Outgoing Access indicator
This parameter represents the Outgoing Access as defined in ETS 300 136.

5.6.3.9 Operator Determined Barring General Data
This parameter refers to the set of subscribers features that the network operator or the service provider can regulate. This set only includes those limitations that can be controlled in the VLR:
- All outgoing calls barred;
- International outgoing calls barred;
- International outgoing calls except those to the home PLMN country barred;
- Interzonal outgoing calls barred;
- Interzonal outgoing calls except those to the home PLMN country barred;
- Interzonal outgoing calls AND international outgoing calls except those directed to the home PLMN country barred;
- Premium rate (information) outgoing calls barred;
- Premium rate (entertainment) outgoing calls barred;
- Supplementary service access barred;
- Invocation of call transfer barred;
- Invocation of chargeable call transfer barred;
- Invocation of internationally chargeable call transfer barred;
- Invocation of interzonaully chargeable call transfer barred;
- Invocation of call transfer where both legs are chargeable barred;
- Invocation of call transfer if there is already an ongoing transferred call for the served subscriber in the serving MSC/VLR barred.

5.6.3.10 ODB HPLMN Specific Data
This parameter refers to the set of subscribers features that the network operator or the service provider can regulate only when the subscriber is registered in the HPLMN. This set only includes those limitations that can be controlled in the VLR:
- Operator Determined Barring Type 1;
- Operator Determined Barring Type 2;
- Operator Determined Barring Type 3;
- Operator Determined Barring Type 4.
5.6.3.11 Regional Subscription Data

This parameter defines the regional subscription area in which the subscriber is allowed to roam. It consists of a list of Zone Codes (see subclause 5.6.2.28).

5.6.3.12 Regional Subscription Response

This parameter indicates either that the regional subscription data cannot be handled or that the current MSC area is entirely restricted because of regional subscription.

5.6.3.13 Roaming Restriction Due To Unsupported Feature

This parameter defines that a subscriber is not allowed to roam in the current MSC area. It may be used by the HLR if a feature or service is indicated as unsupported by the VLR.

5.6.3.14 Extensible SS-Info

This parameter refers to all the information related to a supplementary service and is a choice between:

- extensible forwarding information (see subclause 5.6.3.15);
- call barring information (see subclause 5.6.3.20);
- extensible CUG info (see subclause 5.6.3.22);
- extensible SS-Data (see subclause 5.6.3.29).

5.6.3.15 Extensible Forwarding information

This parameter represents the information related to each call forwarding service:

- the SS-Code of the relevant call forwarding service (see subclause 5.6.4.1);
- if required, a list of extensible forwarding feature parameters (see subclause 5.6.3.16).

The list may contain one item per Basic Service Group.

5.6.3.16 Extensible Forwarding feature

This parameter applies to each combination of call forwarding service and Basic Service Group and contains the following information, as required:

- extensible Basic Service Group (see subclause 5.6.3.5);
- extensible SS-Status (see subclause 5.6.3.17);
- forwarded-to number (see subclause 5.6.2.22);
- forwarded-to subaddress (see subclause 5.6.2.23);
- extensible forwarding options (see subclause 5.6.3.18);
- extensible no reply condition timer (see subclause 5.6.4.19).

5.6.3.17 Extensible SS-Status

This parameter refers to the state information of individual supplementary services as defined in TS GSM 03.11.
5.6.3.18 Extensible Forwarding Options
This parameter refers to a set of forwarding options attached to a supplementary service. It contains the following informations:
- notification to forwarding party (see TS GSM 02.82 for the meaning of this parameter);
- notification to calling party (see TS GSM 02.82 for the meaning of this parameter);
- Forwarding reason (see TS GSM 02.82 for the meaning of this parameter).

5.6.3.19 Extensible No reply condition timer
This parameter refers to the extensible no reply condition timer for call forwarding on no reply.

5.6.3.20 Extensible Call barring information
This parameter contains for each call barring service:
- SS-Code (see subclause 5.6.4.1);
- a list of extensible call barring feature parameters (see subclause 5.6.3.21).

The list may contain one item per Basic Service Group.

5.6.3.21 Extensible Call barring feature
This parameter gives the status of call barring services as applicable to each Basic Service Group. The parameter contains the following information:
- Extensible Basic Service Group (see subclause 5.6.3.5);
- provisioned SS-Status (see subclause 5.6.3.17).

5.6.3.22 CUG info
This parameter refers to the overall information required for operation for each CUG:
- CUG subscriptionList;
- CUG featureList.

5.6.3.23 CUG subscription
This parameter refers to the set of basic information for each CUG defined in that subscription. The following information is stored:
- CUG index;
- CUG interlock;
- Intra CUG restrictions;
- Basic Service Group List.

5.6.3.24 CUG interlock
This parameter represents the CUG interlock code defined in ETS 300 138.

5.6.3.25 CUG index
This parameter represents the CUG index defined in ETS 300 138.
5.6.3.26 CUG feature

This parameter contains two parameters which are associated with the Basic Service Group. If the Basic Service Group Code is not present the feature applies to all Basic Services. The following parameters are included:

- Preferential CUG indicator:
  indicates which CUG index is to be used at outgoing call set-up using the associated Basic Service Group;

- Inter CUG Option:
  describes whether it for the associated Basic Service Group is allowed to make calls outside the CUG and whether incoming calls are allowed;

- Basic Service Group.

See TS GSM 02.85 for meaning of this parameter.

5.6.3.27 Inter CUG options

This parameter indicates the subscribers ability to make and receive calls outside a specific closed user group. It takes any of the following values:

- CUG only facility (only calls within CUG are allowed);
- CUG with outgoing access (calls outside CUG allowed);
- CUG with incoming access (calls from outside CUG into CUG allowed);
- CUG with both incoming and outgoing access (all calls allowed).

5.6.3.28 Intra CUG restrictions

This parameter describes whether or not the subscriber is allowed to originate calls to or to receive calls from within the CUG. It can take any of the following values:

- no CUG restrictions;
- CUG incoming calls barred;
- CUG outgoing calls barred.

5.6.3.29 Extensible SS-Data

This parameter refers to the necessary set of information required in order to characterise one supplementary service:

- SS-Code (see subclause 5.6.4.1);
- Extensible SS-Status (if applicable) (see subclause 5.6.3.17);
- Extensible Override subscription option (if applicable) (see subclause 5.6.3.30);
- Extensible CLI Restriction (if applicable) (see subclause 5.6.3.31);
- Extensible Basic Service Group Code (see subclause 5.6.3.5).

5.6.3.30 Subscriber State

This parameter indicates the state of the MS as defined in GSM 03.18.

5.6.3.31 Requested Info

This parameter indicates the subscriber information being requested as defined in GSM 03.18.
5.6.3.32 Suppression of Announcement
This parameter indicates if the announcement or tones shall be suppressed as defined in GSM 03.78.

5.6.3.33 Suppress T-CSI
This parameter is used to suppress the invocation of terminating CAMEL services.

5.6.3.34 GMSC CAMEL Subscription Info
This parameter contains CAMEL subscription information, i.e. O-CSI and/or T-CSI, which indicates to the GMSC that originating and/or terminating CAMEL services shall be invoked for the incoming call.

5.6.3.35 VLR CAMEL Subscription Info
This parameter identifies the subscriber as having CAMEL services which are invoked in the MSC.

5.6.3.36 Supported CAMEL Phases
This parameter indicates which phases of CAMEL are supported.

5.6.3.37 CUG Subscription Flag
This parameter indicates that a subscriber with a T-CSI also has a CUG subscription. It is defined in TS GSM 03.78.

5.6.3.38 CAMEL Subscription Info Withdraw
This parameter indicates that CAMEL Subscription Info shall be deleted from the VLR.

5.6.3.39 Voice Group Call Service (VGCS) Data
This parameter refers to one or more groups a subscriber may be member of for voice group calls.

5.6.3.40 Voice Broadcast Service (VBS) Data
This parameter refers to one or more groups a subscriber may be member of for the voice broadcast service. Per group it is further indicated whether the subscriber is only allowed to listen to respective group calls or whether he is in addition entitled to initiate respective voice broadcast calls.

5.6.4 Supplementary services parameters

5.6.4.1 SS-Code
This parameter may refer to one supplementary service or a set of supplementary services as defined in TS GSM 02.04. For MAP Release '96 this includes:

- Calling Line Identification Presentation service (CLIP);
- Calling Line Identification Restriction service (CLIR);
- Connected Line Identification Presentation service (COLP);
- Connected Line Identification Restriction service (COLR);
- All Call Forwarding services;
- Call Waiting (CW);
- Call Hold (HOLD);
- Multi-Party service (MPTY);
- Closed User Group (CUG);
- All Charging services;
- All Call Restriction services;
- Explicit Call Transfer service (ECT);
- enhanced Multi-Level Precedence and Pre-emption service (eMLPP).

5.6.4.2 SS-Status

This parameter refers to the state information of individual supplementary services as defined in GSM 03.11.

5.6.4.3 SS-Data

This parameter refers to the necessary set of information required in order to characterise one supplementary service:

- SS-Code  (see subclause 5.6.4.1);
- SS-Status (if applicable)  (see subclause 5.6.4.2);
- Override subscription option  (see subclause 5.6.4.4);
- CLI Restriction  (see subclause 5.6.4.5);
- Basic Service Group Code  (see subclause 5.6.4.40).

5.6.4.4 Override Category

This parameter refers to the subscription option Override Category attached to a supplementary service. It can take the following two values:

- Enabled;
- Disabled.

5.6.4.5 CLI Restriction Option

This parameter refers to the subscription option Restriction mode attached to the CLIR supplementary service. It can take the following three values:

- Permanent;
- Temporary (Default Restricted);
- Temporary (Default Allowed).

5.6.4.6 Forwarding Options

This parameter refers to a forwarding option attached to a supplementary service. It can take one of the following values:

- notification to forwarding party  (see GSM 02.82 for the meaning of this parameter);
- notification to calling party  (see GSM 02.82 for the meaning of this parameter);
- Forwarding reason  (see GSM 02.82 for the meaning of this parameter).

5.6.4.7 No reply condition timer

This parameter refers to the no reply condition timer for call forwarding on no reply.
5.6.4.8 - 5.6.4.14  [Spare]

5.6.4.15  Forwarding information

This parameter represents the information related to each call forwarding service:

- the SS-Code of the relevant call forwarding service (see subclause 5.6.4.1);
- if required, a list of forwarding feature parameters (see subclause 5.6.4.16).

The list may contain one item per Basic Service Group.

5.6.4.16  Forwarding feature

This parameter applies to each combination of call forwarding service and Basic Service Group and contains the following information, as required:

- Basic Service Group (see subclause 5.6.4.40);
- SS-Status (see subclause 5.6.4.2);
- forwarded-to number (see subclause 5.6.2.22);
- forwarded-to subaddress (see subclause 5.6.2.23);
- forwarding options (see subclause 5.6.4.6);
- no reply condition timer (see subclause 5.6.4.7).

5.6.4.17  [Spare]

5.6.4.18  Call barring information

This parameter contains for each call barring service:

- SS-Code (see subclause 5.6.4.1);
- a list of call barring feature parameters (see subclause 5.6.4.19).

The list may contain one item per Basic Service Group.

5.6.4.19  Call barring feature

This parameter gives the status of call barring services as applicable to each Basic Service Group. The parameter contains the following information:

- Basic Service Group (see subclause 5.6.4.40);
- SS-Status (see subclause 5.6.4.2).

5.6.4.20  New password

This parameter refers to the password which the subscriber just registered in the network.

This parameter refers to a password used by the subscriber for supplementary service control.

5.6.4.21  Current password

This parameter refers to a password used by the subscriber for supplementary service control.
5.6.4.22 Guidance information

This parameter refers to guidance information given to a subscriber who is requested to provide a password. One of the following information may be given:

- "enter password";
  This information is used for checking of the old password.
- "enter new password";
  This information is used during password registration for the request of the first new password.
- "enter new password again";
  This information is used during password registration for the request of the new password again for verification.

5.6.4.23 [Spare]

5.6.4.24 SS-Info

This parameter refers to all the information related to a supplementary service and is a choice between:

- forwarding information (see subclause 5.6.4.15);
- call barring information (see subclause 5.6.4.18);
- CUG info (see subclause 5.6.4.8);
- SS-Data (see subclause 5.6.4.3).
- eMLPP information (see subclause 5.6.4.41).

5.6.4.25 - 5.6.4.35 [Spare]

5.6.4.36 USSD Data Coding Scheme

This parameter contains the information of the alphabet and the language used for the unstructured information in an Unstructured Supplementary Service Data operation. The coding of this parameter is according to the Cell Broadcast Data Coding Scheme as specified in GSM 03.38.

5.6.4.37 USSD String

This parameter contains a string of unstructured information in an Unstructured Supplementary Service Data operation. The string is sent either by the mobile user or the network. The contents of a string sent by the MS are interpreted by the network as specified in GSM 02.90.

5.6.4.38 Bearer service

This parameter may refer to a single bearer service, a set of bearer services or to all bearer services as defined in TS GSM 02.02. This parameter is used only for supplementary service management.

5.6.4.39 Teleservice

This parameter may refer to a single teleservice, a set of teleservices or to all teleservices as defined in TS GSM 02.03. This parameter is used only for supplementary service management.
5.6.4.40  Basic Service Group

This parameter refers to the Basic Service Group either as a bearer service (see subclause 5.6.4.38) or a teleservice (see subclause 5.6.4.39). This parameter is used only for supplementary service management. The null value (i.e. neither bearer service nor teleservice) is used to denote the group containing all bearer services and all teleservices.

5.6.4.41  eMLPP information

This parameter contains two parameters which are associated with the eMLPP service. The following two parameters are included:

- maximum entitled priority:
  indicates the highest priority level the subscriber is allowed to apply for an outgoing call set-up;
- default priority:
  defines the priority level which shall be assigned to a call if no explicit priority is indicated during call set-up.

5.6.5  Call parameters

5.6.5.1  Call reference number

This parameter refers to a call reference number allocated by a call control MSC.

5.6.5.2  Interrogation type

This parameter refers to the type of interrogation for routing information which is sent from a GMSC to an HLR. It can take either of two values:

- basic call (for information to route a call before the call has been extended to the VMSC of the called party);
- forwarding (for information to route the call to the forwarded-to destination after the VMSC of the forwarding party has requested the GMSC to resume handling of the call).

5.6.5.3  OR interrogation

This parameter indicates that the GMSC which interrogated the HLR for routeing information is not in the same PLMN as the HLR, and therefore that the call will potentially be optimally routed.

5.6.5.4  OR capability

This parameter indicates the phase of OR which the GMSC supports.

5.6.5.5  Forwarding reason

This parameter indicates the reason for which the call is to be forwarded. It can take one of three values:

- busy subscriber;
- mobile subscriber not reachable;
- no subscriber reply.

5.6.5.6  Forwarding interrogation required

This parameter indicates that if the VMSC of the forwarding subscriber requests the GMSC to resume handling of the call the GMSC shall interrogate the HLR for forwarding information.

5.6.5.7  O-CSI

This parameter identifies the subscriber as having originating CAMEL services as defined in TS GSM 03.78
5.6.6 Radio parameters

5.6.6.1 - 5.6.6.6 [Spare]

5.6.6.7 HO-Number Not Required
This parameter indicates that no handover number allocation is necessary.

5.6.7 Authentication parameters

5.6.7.1 Authentication set list
This parameter represents a list of sets of authentication parameters for a given subscriber:
- Rand;
- Sres;
- Kc.

5.6.7.2 Rand
This parameter represents a random number used for authentication.

5.6.7.3 Sres
This parameter represents the response to an authentication request.

5.6.7.4 Kc
This parameter refers to a key used for ciphering purposes.

5.6.7.5 [Spare]

5.6.7.6 Cksn
This parameter refers to a ciphering key sequence number.

5.6.7.7 Ciphering mode
This parameter refers to the ciphering mode which is associated with a radio channel. It may take values as follows:
- no encryption;
- identification of specific ciphering algorithm.

5.6.8 Short message parameters

5.6.8.1 SM-RP-DA
This parameter represents the destination address used by the short message service relay sub-layer protocol. It can be either of the following:
- IMSI (see subclause 5.6.2.1);
- LMSI (see subclause 5.6.2.16);
- MS-ISDN (see subclause 5.6.2.17);
- roaming number (see subclause 5.6.2.19);
- service centre address (see subclause 5.6.2.27).

5.6.8.2 SM-RP-OA

This parameter refers to the originating address used by the short message service relay sub-layer protocol. It can be either of the following:

- MS-ISDN (see subclause 5.6.2.17);
- service centre address (see subclause 5.6.2.27).

5.6.8.3 MWD status

This parameter indicates whether or not the address of the originator service centre is already contained in the Message Waiting Data file. In addition, it contains the status of the Memory Capacity Exceeded Flag (MCEF) and the status of the Mobile subscriber Not Reachable Flag (MNRF).

5.6.8.4 SM-RP-UI

This parameter represents the user data field carried by the short message service relay sub-layer protocol.

5.6.8.5 SM-RP-PRI

This parameter is used to indicate whether or not delivery of the short message shall be attempted when a service centre address is already contained in the Message Waiting Data file.

5.6.8.6 SM Delivery Outcome

This parameter indicates the cause for setting the message waiting data. It can take one of the following values:

- Absent subscriber;
- MS memory capacity exceeded;
- Successful transfer.

5.6.8.7 More Messages To Send

This parameter is used to indicate whether or not the service centre has more short messages to send.

5.6.8.8 Alert Reason

This parameter is used to indicate the reason why the service centre is alerted. It can take one of the following values:

- MS present;
- Memory Available.

5.6.8.9 Absent Subscriber Diagnostic SM

This parameter is used to indicate the reason why the subscriber is absent. For the values for this parameter see TS GSM 03.40.
5.6.9 Access and signalling system related parameters

5.6.9.1 BSS-apdu
This parameter includes one or two concatenated complete 08.06 messages, as described in GSM 03.09 and GSM 09.10. The Protocol ID indicates that the message or messages are according to GSM 08.06. For the coding of the messages see GSM 08.06 and GSM 08.08.

5.6.9.2 CM service type
This parameter identifies the service category being requested by the subscriber:
- mobile originating call;
- emergency call establishment;
- short message service;
- mobile originating call re-establishment;
- mobile terminating call;
- SS request;
- Voice group call setup;
- Voice broadcast setup.

5.6.9.3 Access connection status
This parameter represents the following access connection status information:
- RR-connection status (established/not established);
- ciphering mode (on/off);
- authentication status (authenticated/not authenticated).

5.6.9.4 External Signal Information
This parameter contains concatenated information elements (including tag and length) which are defined by a common protocol version, preceded by the associated protocol ID. It is used to transport information of the indicated protocol via MAP interfaces.

5.6.9.5 Access signalling information
This parameter refers to any set of information elements imported from GSM 04.08.

5.6.9.6 Location update type
This parameter refers to the location update type (normal, periodic or IMSI attach) contained in the GSM 04.08 LOCATION REGISTRATION REQUEST message.

5.6.9.7 Protocol ID
This parameter refers to the protocol to which the coding of the content of the associated External Signal Information conforms.

The following values are defined:
- 04.08;
5.6.9.8  Network signal information

This parameter is transported as external signal information. The protocol ID shall be set to "ETS 300 102-1".

The network signal information may include the following information elements as defined in GSM 09.07:

- ISDN BC; the tag and length are defined by ETS 300 102-1.
  For the content, see GSM 09.07.
- HLC; the tag and length are defined by ETS 300 102-1.
  For the content, see GSM 09.07.
- LLC; the tag and length are defined by ETS 300 102-1.
  For the content, see GSM 09.07.

They are contained in the Signal Information parameter according to figure 5.6/1 (irrespective of the order).

![Figure 5.6/1: Network signal information parameter](image)

5.6.10  System operations parameters

5.6.10.1  Network resources

This parameter refers to a class or type of network resource:

- PLMN;
- HLR;
- VLR (current or previous);
- MSC (controlling or current);
- EIR;
- radio sub-system.
5.6.10.2 Trace reference
This parameter represents a reference associated with a tracing request. The parameter is managed by OMC.

5.6.10.3 Trace type
This parameter identifies the type of trace. Trace types are fully defined in GSM 12.08.

5.7 Representation of a list of a basic parameter in service-primitives
In some service-primitives several instances of a basic parameter of subclause 5.6 are required. In the service descriptions such cases will be represented as

| ParameterNameLIST |

in the tables where ParameterName refers to one of the parameters defined in subclause 5.6. This corresponds to the following construction rule:

Figure 5.7/1: Construction of Lists
6 Mobility services

6.1 Location management services

6.1.1 MAP_UPDATE_LOCATION_AREA service

6.1.1.1 Definition

This service is used between MSC and VLR to update location information in the network. It is initiated by an MS when changing the location area or at first registration. The detailed conditions are given in GSM 03.12.

The MAP_UPDATE_LOCATION_AREA service is a confirmed service using the primitives from table 6.1/1.

6.1.1.2 Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>Target location area Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td></td>
</tr>
<tr>
<td>Location update type</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSI</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMSI</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Previous location area Id</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CKSN</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.1.1.3 parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

Target location area Id

See definition in subclause 5.6.2.

Location update type

See definition in subclause 5.6.9.

IMSI

See definition in subclause 5.6.2. It is up to the MS to provide either IMSI or TMSI, but one shall be present.

TMSI

See definition in subclause 5.6.2. It is up to the MS to provide either IMSI or TMSI, but one shall be present.

Previous location area Id

See definition in subclause 5.6.2. This parameter is provided if the updating is not a first registration.

CKSN

See definition in subclause 5.6.7. The CKSN is given if TMSI is used.

User error
One of the following error causes defined in subclause 5.6.1 is sent by the user in case of location area updating failures, depending on the failure reason:

- **unknown subscriber;**
  
  This cause is used if the subscriber is not known in the VLR and even a correlated request to the subscriber's HLR gives a negative result (i.e. the IMSI is not allocated to a subscriber).

- **unknown location area;**
  
  This cause is used if the target location area identity given is not known in the VLR.

- **roaming not allowed;**
  
  This cause is used if the MS is not allowed to roam into the target location area indicated in the MAP_UPDATE_LOCATION_AREA Req. The cause will be qualified according to the roaming restriction reason, i.e. one of "National Roaming Not Allowed", "PLMN Not Allowed", "Location Area Not Allowed", or "Operator Determined Barring".

- **illegal subscriber;**
  
  This error is sent if a correlated authentication procedure has not authenticated the subscriber.

- **illegal equipment;**
  
  This error is sent if an IMEI check failed, i.e. the IMEI is blacklisted or not white-listed.

- **system failure;**

- **unexpected data value.**

**Provider error**

For definition of provider errors see subclause 5.6.1.

### 6.1.2 MAP_UPDATE_LOCATION service

#### 6.1.2.1 Definition

This service is used by the VLR to update the location information stored in the HLR.

The MAP_UPDATE_LOCATION service is a confirmed service using the service primitives given in table 6.1/2.

#### 6.1.2.2 Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSC Address</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VLR number</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LMSI</td>
<td>U</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>HLR number</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td>C</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 6.1.2.3 Parameter definitions and use

**Invoke Id**

See definition in subclause 5.6.1.
IMSI
See definition in subclause 5.6.2.

MSC Address
See definition in subclause 5.6.2. The MSC address is used for short message delivery only and for each incoming call set-up attempt the MSRN will be requested from the VLR.

VLR number
See definition in subclause 5.6.2.

LMSI
See definition in subclause 5.6.2. It is an operator option to provide the LMSI from the VLR; it is mandatory for the HLR to support the LMSI handling procedures.

HLR number
See definition in subclause 5.6.2. The presence of this parameter is mandatory in case of successful HLR updating.

User error
In case of unsuccessful updating, an error cause shall be returned by the HLR. The following error causes defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- unknown subscriber;
- roaming not allowed;
  
  This cause will be sent if the MS is not allowed to roam into the PLMN indicated by the VLR number. The cause is qualified by the roaming restriction reason "PLMN Not Allowed" or "Operator Determined Barring". If no qualification is received (HLR with MAP Version 1), "PLMN Not Allowed" is taken as default.
- system failure;
- unexpected data value.

Provider error
For definition of provider errors see subclause 5.6.1.

6.1.3 MAP_CANCEL_LOCATION service

6.1.3.1 Definition
This service is used between HLR and VLR to delete a subscriber record from the VLR. It may be invoked automatically when an MS moves from one VLR area to another, to remove the subscriber record from the old VLR, or by the HLR operator to enforce a location updating from the VLR to the HLR, e.g. on withdrawal of a subscription.

The MAP_CANCEL_LOCATION service is a confirmed service using the primitives defined in table 6.1/3.

6.1.3.2 Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>LMSI</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td>O</td>
<td>O(=)</td>
<td>O</td>
<td>O(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1/3: MAP_CANCEL_LOCATION
6.1.3.3  Parameter definitions and use

Invoke Id
See definition in subclause 5.6.1.

IMSI
See definition in subclause 5.6.2.

LMSI
See definition in subclause 5.6.2. The LMSI shall be included if it has been received from VLR.

Value 0000 0000 can be used to indicate that the LMSI is not in use.

User error
If the cancellation fails, an error cause is to be returned by the VLR. The following error cause defined in
subclause 5.6.1 shall be used:
- unexpected data value.

Provider error
For definition of provider errors see subclause 5.6.1.

6.1.4  MAP_SEND_IDENTIFICATION service

6.1.4.1  Definition
The MAP_SEND_IDENTIFICATION service is used between a VLR and a previous VLR to retrieve IMSI and
authentication sets for a subscriber registering afresh in that VLR.

The MAP_SEND_IDENTIFICATION service is a confirmed service using the service primitives defined in table 6.1/4.

6.1.4.2  Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>TMSI</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication set</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.1.4.3  Parameter definitions and use

Invoke Id
See definition in subclause 5.6.1.

TMSI
See definition in subclause 5.6.2.

IMSI
See definition in subclause 5.6.2. The IMSI is to be returned if the service succeeds.
Authentication set

See definition in subclause 5.6.7. If the service succeeds a list of up to five authentication sets is returned, if there are any available.

User error

This parameter is mandatory if the service fails. The following error cause defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- unidentified subscriber.

Provider error

For definition of provider errors see subclause 5.6.1.

6.1.5 MAP_DETACH_IMSI service

6.1.5.1 Definition

The MAP_DETACH_IMSI service is used by the MSC to indicate to the VLR that an MS is no longer reachable. The network needs this information e.g. to reject an incoming call without initiating paging on the radio path.

The MAP_DETACH_IMSI service is a non-confirmed service using the service primitives defined in table 6.1/5.

6.1.5.2 Service primitives

Table 6.1/5: MAP_DETACH_IMSI

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>TMSI</td>
<td>C</td>
<td>C(=)</td>
</tr>
</tbody>
</table>

6.1.5.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2. It is up to the MS to provide either IMSI or TMSI as subscriber identity, but one shall be present.

TMSI

See definition in subclause 5.6.2. It is up to the MS to provide either IMSI or TMSI as subscriber identity, but one shall be present.

6.1.6 MAP_PURGE_MS service

6.1.6.1 Definition

This service is used between the VLR and the HLR to cause the HLR to mark its data for an MS so that any request for routing information for a mobile terminated call or a mobile terminated short message will be treated as if the MS is not reachable. It is invoked when the subscriber record for the MS is to be deleted in the VLR, either by MMI interaction or automatically, e.g. because the MS has been inactive for several days.

The MAP_PURGE_MS service is a confirmed service using the primitives defined in table 6.1/6.
6.1.6.2 Service primitives

Table 6.1/6: MAP_PURGE_MS

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VLR number</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.1.6.3 Parameter definitions and use

Invoke ID
See definition in subclause 5.6.1.

IMSI
See definition in subclause 5.6.2.

VLR number
See definition in subclause 5.6.2.

Provider error
See definition of provider errors in subclause 5.6.1.

6.2 Paging and search

6.2.1 MAP_PAGE service

6.2.1.1 Definition
This service is used between VLR and MSC to initiate paging of an MS for mobile terminated call set-up, mobile terminated short message or unstructured SS notification.

The MAP_PAGE service is a confirmed service using the primitives from table 6.2/1.

6.2.1.2 Service primitives

Table 6.2/1: MAP_PAGE

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stored location area Id</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMSI</td>
<td>U</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td></td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

6.2.1.3 Parameter definitions and use

Invoke Id
See definition in subclause 5.6.1.

IMSI
See definition in subclause 5.6.2. The IMSI is used to define the paging subgroup. If the TMSI is not supplied, paging on the radio path uses the IMSI as an identifier.

**Stored location area Id**

See definition in subclause 5.6.2.

**TMSI**

See definition in subclause 5.6.2. The TMSI is included if paging on the radio channel is to use the TMSI as an identifier.

**User error**

The following error causes defined in subclause 5.6.1 may be sent by the user in case of a paging error, depending on the failure reason:

- absent subscriber;
- unknown location area;
- busy subscriber;
- system failure;

This corresponds to the case where there is no call associated with the MAP_PAGE service, i.e. if the call has been released but the dialogue to the VLR has not been aborted.

- unexpected data value.

**Provider error**

See definition in subclause 5.6.1.

### 6.2.2 MAP_SEARCH_FOR_MS service

#### 6.2.2.1 Definition

This service is used between VLR and MSC to initiate paging of an MS in all location areas of that VLR. It is used if the VLR does not hold location area information confirmed by radio contact.

The MAP_SEARCH_FOR_MS service is a confirmed service using the primitives from table 6.2/2.

#### 6.2.2.2 Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current location area Id</td>
<td>C</td>
<td>C(=)</td>
<td>C(=)</td>
<td>O</td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td>C(=)</td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 6.2.2.3 Parameter definitions and use

**Invoke Id**

See definition in subclause 5.6.1.

**IMSI**

See definition in subclause 5.6.2. The IMSI is used to identify the subscriber when paging on the radio path.
Current location area Id

See definition in subclause 5.6.2. In case of successful outcome of the service, i.e. if the MS responds to paging, the Location Area Id of the area in which the MS responded is given in the response.

User error

The following error causes defined in subclause 5.6.1 shall be sent by the user if the search procedure fails, depending on the failure reason:

- absent subscriber;
  
  This error cause is returned by the MSC if the MS does not respond to the paging request.

- system failure;
  
  This corresponds to the case where there is no call associated with the MAP_SEARCH_FOR_MS service, i.e. if the call has been released but the dialogue to the VLR has not been aborted.

- busy subscriber;

- unexpected data value.

Provider error

See definition in subclause 5.6.1.

6.3 Access management services

6.3.1 MAP_PROCESS_ACCESS_REQUEST service

6.3.1.1 Definition

This service is used between MSC and VLR to initiate processing of an MS access to the network, e.g. in case of mobile originated call set-up or after being paged by the network.

The MAP_PROCESS_ACCESS_REQUEST service is a confirmed service using the primitives from table 6.3/1.

6.3.1.2 Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>CM service type</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Access connection status</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current Location Area Id</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMSI</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cksn</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSI</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>IMEI</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>MSISDN</td>
<td>U</td>
<td></td>
<td>U</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.3.1.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

CM service type


See definition in subclause 5.6.9.

Access connection status
See definition in subclause 5.6.9.

Current Location Area Id
See definition in subclause 5.6.2. This parameter is used to update the VLR in case of previous VLR failure.

TMSI
See definition in subclause 5.6.2. Either TMSI or IMSI as received from the MS are included in the Request/Indication, but one shall be present. In case of CM Service Type "Emergency Call Establishment", the IMEI may replace IMSI/TMSI.

Cksn
See definition in subclause 5.6.7. In case of access with TMSI, the Cksn shall be present.

IMSI
See definition in subclause 5.6.2. Either TMSI or IMSI as received from the MS are included in the Request/Indication, but one shall be present. In case of CM Service Type "Emergency Call Establishment", the IMEI may replace IMSI/TMSI.

In the Response/Confirmation, the IMSI is to be sent in case of successful outcome of the service. In case of CM Service Type "Emergency Call Establishment", IMEI may replace IMSI.

IMEI
See definition in subclause 5.6.2. The IMEI may replace IMSI/TMSI in the Request/Indication and IMSI in the Response/Confirmation only in case the CM Service Type indicates "Emergency Call Establishment".

MSISDN
See definition in subclause 5.6.2. The MSISDN is included in case of successful outcome of the service as an operator option, e.g. if it is needed at the MSC for charging purposes in case of call forwarding.

User error
One of the following error causes defined in subclause 5.6.1 shall be sent by the user if the access request fails, depending on the failure reason:

- unidentified subscriber;
- illegal subscriber;
  This error is sent if a correlated authentication procedure has not authenticated the subscriber.
- illegal equipment;
  This error is sent if an IMEI check failed, i.e. the IMEI is blacklisted or not white-listed.
- roaming not allowed;
  This cause is used after VLR restart if the subscriber has no subscription for the current location area, e.g. due to regional subscription. The cause will be qualified by "location area not allowed" or "national roaming not allowed", respectively.
- unknown location area;
- system failure;
- unexpected data value.

Provider error
For definition of provider errors see subclause 5.6.1.

6.4 Handover services

6.4.1 MAP_PREPARE_HANDOVER service

6.4.1.1 Definition

This service is used between MSC-A and MSC-B (E-interface) when a call is to be handed over from MSC-A to MSC-B.

The MAP_PREPARE_HANDOVER service is a confirmed service using the primitives from table 6.4/1.

6.4.1.2 Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>Target Cell Id</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HO-NumberNotRequired</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSS-APDU</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Handover Number</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>O</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.4.1.3 Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

Target Cell Id

For definition of this parameter see subclause 5.6.2. This parameter is only included if the service is not in an ongoing transaction.

HO-Number Not Required

For definition of this parameter see subclause 5.6.6.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

Handover Number

For definition of this parameter see subclause 5.6.2. This parameter shall be returned, unless the parameter HO-NumberNotRequired is sent.

User error

For definition of this parameter see subclause 5.6.1. The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- No handover number available;
- System failure;
- Unexpected data value;
- DataMissing.
Provider error
See definition of provider errors in subclause 5.6.1.

6.4.2 MAP_SEND_END_SIGNAL service

6.4.2.1 Definition
This service is used between MSC-B and MSC-A (E-interface) indicating that the radio path has been established by MSC-B to the MS. MSC-A retains then the main control of the call until it clears.

The response is used by MSC-A to inform MSC-B that all resources for the call can be released in MSC-B, either because the call has been released in MSC-A or because the call has been successfully handed over from MSC-B to another MSC.

The MAP_SEND_END_SIGNAL service is a confirmed service using the primitives from table 6.4/2.

6.4.2.2 Service primitives

Table 6.4/2: MAP_SEND_END_SIGNAL

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>BSS-APDU</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.4.2.3 Parameter use

Invoke Id
For definition of this parameter see subclause 5.6.1.

BSS-APDU
For definition of this parameter see subclause 5.6.9.

Provider error
For definition of this parameter see subclause 5.6.1.

6.4.3 MAP_PROCESS_ACCESS_SIGNALLING service

6.4.3.1 Definition
This service is used between MSC-B and MSC-A (E-interface) to pass information received on the A-interface in MSC-B to MSC-A.

The MAP_PROCESS_ACCESS_SIGNALLING service is a non-confirmed service using the primitives from table 6.4/3.

6.4.3.2 Service primitives

Table 6.4/3: MAP_PROCESS_ACCESS_SIGNALLING

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>BSS-APDU</td>
<td>M</td>
<td>M(=)</td>
</tr>
</tbody>
</table>
6.4.3.3 Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

6.4.4 MAP_FORWARD_ACCESS_SIGNALLING service

6.4.4.1 Definition

This service is used between MSC-A and MSC-B (E-interface) to pass information to be forwarded to the A-interface of MSC-B.

The MAP_FORWARD_ACCESS_SIGNALLING service is a non-confirmed service using the primitives from table 6.4/4.

6.4.4.2 Service primitives

Table 6.4/4: MAP_FORWARD_ACCESS_SIGNALLING

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>BSS-APDU</td>
<td>M</td>
<td>M(=)</td>
</tr>
</tbody>
</table>

6.4.4.3 Parameter use

For the definition and use of all parameters and errors, see subclause 5.6.1.

Invoke Id

For definition of this parameter see subclause 5.6.1.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

6.4.5 MAP_PREPARE_SUBSEQUENT_HANDOVER service

6.4.5.1 Definition

This service is used between MSC-B and MSC-A (E-interface) to inform MSC-A that it has been decided that a handover to either MSC-A or a third MSC (MSC-B') is required.

The MAP_PREPARE_SUBSEQUENT_HANDOVER service is a confirmed service using the primitives from table 6.4/5.
6.4.5.2 Service primitives

Table 6.4/5: MAP_PREPARE_SUBSEQUENT_HANOVER

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>Target Cell Id</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target MSC Number</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSS-APDU</td>
<td>M</td>
<td>M(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

6.4.5.3 Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

Target Cell Id

For definition of this parameter see subclause 5.6.2.

Target MSC Number

For definition of this parameter see subclause 5.6.2.

BSS-APDU

For definition of this parameter see subclause 5.6.9.

User error

For definition of this parameter see subclause 5.6.1. The following error causes defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Unknown MSC;
- Subsequent handover failure;
- Unexpected data value;
- Data Missing.

Provider error

For definition of this parameter see subclause 5.6.1.

6.4.6 MAP_ALLOCATE_HANOVER_NUMBER service

6.4.6.1 Definition

This service is used between MSC and VLR (B-interface) to request a handover number.

The MAP_ALLOCATE_HANOVER_NUMBER service is a confirmed service using the primitives from table 6.4/6.
6.4.6.2 Service primitives

Table 6.4/6: MAP_ALLOCATE_HANDOVER_NUMBER

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td></td>
<td>C(=)</td>
<td>O</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.4.6.3 Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

User error

For definition of this parameter see subclause 5.6.1. The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- No handover number available.

Provider error

For definition of this parameter see subclause 5.6.1.

6.4.7 MAP_SEND_HANDOVER_REPORT service

6.4.7.1 Definition

This service is used between VLR and MSC-B (B-interface) to transfer the handover number to be forwarded to and used by MSC-A.

The MAP_SEND_HANDOVER_REPORT service is a confirmed service using the primitives from table 6.4/7.

6.4.7.2 Service primitives

Table 6.4/7: MAP_SEND_HANDOVER_REPORT

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>Handover Number</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>Provider error</td>
</tr>
<tr>
<td>Linked Id</td>
<td></td>
<td></td>
<td></td>
<td>Linked Id</td>
</tr>
</tbody>
</table>

6.4.7.3 Parameter use

Invoke Id

For definition of this parameter see subclause 5.6.1.

Handover Number

For definition of this parameter see subclause 5.6.2.

Linked Id

For definition of this parameter see subclause 5.6.1. This service is linked with MAP_ALLOCATE_HANDOVER_NUMBER.

Provider error
For definition of this parameter see subclause 5.6.1.

6.5 Authentication management services

6.5.1 MAP_AUTHENTICATE service

6.5.1.1 Definition

This service is used between the VLR and the MSC when the VLR receives a MAP service indication from the MSC concerning a location registration, call set-up, operation on a supplementary service or a request from the MSC to initiate authentication.

The service is a confirmed service and consists of four service primitives.

6.5.1.2 Service primitives

The service primitives are shown in table 6.5/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>RAND</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CKSN</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRES</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td>M</td>
<td></td>
<td>M(=)</td>
</tr>
</tbody>
</table>

6.5.1.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

RAND

See subclause 5.6.7 for the use of this parameter.

CKSN

See subclause 5.6.7 for the use of this parameter.

SRES

See subclause 5.6.7 for the use of this parameter.

Provider error

See subclause 5.6.1 for the use of this parameter.

6.5.2 MAP_SEND_AUTHENTICATION_INFO service

6.5.2.1 Definition

This service is used between the VLR and the HLR for the VLR to retrieve authentication information from the HLR. The VLR requests some sets of RAND/SRES/Kc vectors.

If the HLR cannot provide the VLR with triplets, an empty response is returned. The VLR may then re-use old authentication triplets, except where this is forbidden under the conditions specified in GSM 03.20 [24].
If the VLR receives a MAP-Send_AUTHENTICATION_INFO response containing a User Error parameter as part of the handling of an authentication procedure, the authentication procedure in the VLR shall fail.

Security related network functions are further described in GSM 03.20.

The service is a confirmed service and consists of four service primitives.

### 6.5.2.2 Service primitives

The service primitives are shown in table 6.5/2.

**Table 6.5/2: MAP_SEND_AUTHENTICATION_PARAMETERS parameters**

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AuthenticationSetList</td>
<td>C</td>
<td></td>
<td>C(=)</td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td></td>
<td>C(=)</td>
<td>O</td>
</tr>
<tr>
<td>Provider error</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 6.5.2.3 Parameter use

**Invoke id**

See subclause 5.6.1 for the use of this parameter.

**IMSI**

See subclause 5.6.2 for the use of this parameter.

**AuthenticationSetList**

A set of one to five authentication vectors are transferred from the HLR to the VLR, if the outcome of the service was successful.

**User error**

One of the following error causes defined in subclause 5.6.1 shall be sent by the user in case of unsuccessful outcome of the service, depending on the respective failure reason:

- unknown subscriber;
- unexpected data value;
- system failure;
- data missing.

**Provider error**

See subclause 5.6.1 for the use of this parameter.

### 6.6 Security management services

#### 6.6.1 MAP_SET_CIPHERING_MODE service

#### 6.6.1.1 Definitions

This service is used between the VLR and the MSC to set the ciphering mode and to start ciphering if applicable. It is called when another service requires that information is to be sent on the radio path in encrypted form.

The service is a non-confirmed service and consists of two service primitives.
6.6.1.2 Service primitives

The service primitives are shown in table 6.6/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>Ciphering mode</td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>Kc</td>
<td>C</td>
<td>C(=)</td>
</tr>
</tbody>
</table>

6.6.1.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

Ciphering mode

See subclause 5.6.7 for the use of this parameter.

Kc

The Kc parameter should be included when the ciphering mode parameter indicates that ciphering must be performed.

6.7 International mobile equipment identities management services

6.7.1 MAP_CHECK_IMEI service

6.7.1.1 Definition

This service is used between the VLR and the MSC and between the MSC and the EIR to request check of IMEI. If the IMEI is not available in the MSC, it is requested from the MS and transferred to the EIR in the service request.

The service is a confirmed service and consists of four service primitives.

6.7.1.2 Service primitives

The service primitives are shown in table 6.7/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMEI</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Equipment status</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6.7.1.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.
IMEI

See subclause 5.6.2 for the use of this parameter. The parameter shall not be included in the service request between the VLR and the MSC, but is mandatory in the service request from the MSC to the EIR. It is not included in the service response from the EIR to the MSC, but is mandatory in the service response from the MSC to the VLR on successful outcome.

Equipment status

See subclause 5.6.4 for the use of this parameter. This parameter is sent by the responder in case of successful outcome of the service.

User error

One of the following error causes defined in subclause 5.6.1 shall be sent by the user in case of unsuccessful outcome of the service, depending on the respective failure reason:

- unknown equipment;
  This error is returned by the responder when the IMEI is not known in the EIR.
- system failure;
- unexpected data value.

Provider error

See subclause 5.6.1 for the use of this parameter.

6.7.2 MAP_OBTAIN_IMEI service

6.7.2.1 Definition

This service is used between the VLR and the MSC to request the IMEI. If the IMEI is not available in the MSC, it is requested from the MS.

The service is a confirmed service and consists of four service primitives.

6.7.2.2 Service primitives

The service primitives are shown in table 6.7/2.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMEI</td>
<td></td>
<td>C</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td>C</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

6.7.2.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

IMEI

See subclause 5.6.2 for the use of this parameter. The parameter IS included in the service response from the MSC to the VLR on successful outcome of the service.

User error
If the service fails, the VLR sends the user error System Failure (see subclause 5.6.1) to the MSC.

**Provider error**

See subclause 5.6.1 for the use of this parameter.

6.8 Subscriber management services

6.8.1 MAP-INSERT-SUBSCRIBER-DATA service

6.8.1.1 Definition

This service is used by an HLR to update a VLR with certain subscriber data in the following occasions:

- the operator has changed the subscription of one or more supplementary services, basic services or data of a subscriber. Note that in case of withdrawal of a Basic or Supplementary service this primitive shall not be used;
- the operator has applied, changed or removed Operator Determined Barring;
- the subscriber has changed data concerning one or more supplementary services by using a subscriber procedure;
- the HLR provides the VLR with subscriber parameters at location updating of a subscriber or at restoration. In this case, this service is used to indicate explicitly that a supplementary service is not provisioned, if the supplementary service specification requires it. The only supplementary services which have this requirement are the CLIR and COLR services.

It is a confirmed service and consists of the primitives shown in table 6.8/1.

6.8.1.2 Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSISDN</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Category</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subscriber Status</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bearer service List</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Teleservice List</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Forwarding information List</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Call barring information List</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUG information List</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS-Data List</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>eMLPP Subscription Data</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operator Determined Barring General data</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Operator Determined Barring HPLMN data</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Roaming Restriction Due To Unsupported Feature</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regional Subscription Data</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VLR CAMEL Subscription Info</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voice Broadcast Data</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Voice Group Call Data</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS-Code List</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regional Subscription Response</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supported CAMEL Phases</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td>U</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6.8.1.3 Parameter use

All parameters are described in subclause 5.6. The following clarifications are applicable:

IMSI

It is only included if the service is not used in an ongoing transaction (e.g. location updating).

MSISDN

It is included either at location updating or when it is changed. The MSISDN sent shall be the basic MSISDN.

Category

It is included either at location updating or when it is changed.

Subscriber Status

It is included either at location updating or when it is changed.

To apply, remove or update Operator Determined Barring Categories the Subscriber Status is set to Operator Determined Barring. In this case ODB General Data shall also be present. If the Operator Determined Barring applies and the subscriber is registered in the HPLMN and HPLMN specific Operator Determined Barring applies then ODB HPLMN Specific Data shall also be present.

To remove all Operator Determined Barring Categories the Subscriber Status shall be set to "Service Granted".

Bearer service List

A list of Extensible Bearer service parameters (Extensible Bearer service is defined in subclause 5.6). An Extensible Bearer service parameter must be the code for an individual Bearer service, except in the cases described below.

The codes for the Bearer service groups "allAlternateSpeech-DataCDA" and "allAlternateSpeech-DataCDS" shall, if applicable, be sent from the HLR to the VLR as a pair. The codes for the Bearer service groups "allSpeechFollowedByDataCDA" and "allSpeechFollowedByDataCDS" shall, if applicable, be sent from the HLR to the VLR as a pair.

If it is included in the Request/Indication, it includes either all Extensible Bearer services subscribed (at location updating or at restoration) or only the ones added (at subscriber data modification).

If the VLR receives an Indication containing any Extensible Bearer service parameters which it does not support/allocate it returns them in the response to the HLR and discards the unsupported Extensible Bearer services (no error is sent back), except in the cases described below.

If the VLR receives the codes for the Bearer service groups "allSpeechFollowedByDataCDA" and "allSpeechFollowedByDataCDS" and supports one or more of the circuit-switched synchronous or asynchronous data rates specified for simple data bearer services, it shall accept the bearer service codes, and not return them in the response to the HLR. If the VLR does not support any of the circuit-switched synchronous or asynchronous data rates specified for simple data bearer services, and receives the pair of codes for "allAlternateSpeech-DataCDA" and "allAlternateSpeech-DataCDS" or the pair of codes for "allSpeechFollowedByDataCDA" and "allSpeechFollowedByDataCDS", it shall reject the pair of codes by returning them in the response to the HLR.

Teleservice List

A list of Extensible Teleservice parameters (Extensible Teleservice is defined in subclause 5.6). An Extensible Teleservice parameter must be the code for an individual Teleservice.

If it is included in the Request/Indication, it contains either all Extensible Teleservices subscribed (at location updating or at restoration) or the ones added (at subscriber data modification).

If the VLR receives an Indication containing any Extensible Teleservice parameters which it does not support/allocate it returns them in the response to the HLR and discards the unsupported Extensible Teleservices (no error is sent back).
Forwarding information List

A list of Extensible Forwarding information parameters (Extensible Forwarding information is defined in subclause 5.6). It includes Call Forwarding services either at location updating or at restoration or when they are changed. Each Extensible Forwarding information parameter shall be treated independently of all other parameters in the primitive.

The Extensible Forwarding information shall include the SS-Code for an individual call forwarding supplementary service. The Extensible Forwarding information shall contain one or more Extensible Forwarding Features (Extensible Forwarding Feature is defined in subclause 5.6).

The Extensible Forwarding Feature may include an Extensible Basic Service Group. This shall be interpreted according to the rules in subclause 6.8.1.4.

The Extensible Forwarding Feature shall contain an Extensible SS-Status parameter.

If the Extensible SS-Status indicates that call forwarding is registered then (except for call forwarding unconditional) the Extensible Forwarding Feature shall contain a forwarded-to number and, if available, the forwarded-to subaddress. In other states the forwarded-to number and, if applicable, the forwarded-to subaddress shall not be included. For call forwarding unconditional the forwarded-to number and, if applicable, the forwarded-to subaddress shall not be included. If the VLR does not receive a forwarded-to subaddress then it shall assume that a forwarded-to subaddress has not been registered.

The Extensible Forwarding Feature shall contain the extensible forwarding options (except for call forwarding unconditional where the extensible forwarding options shall not be included). Bits 3 and 4 of the extensible forwarding options shall be ignored by the VLR, and may be set to any value by the HLR.

For call forwarding on no reply: If the extensible SS-Status indicates that call forwarding is registered then the Extensible Forwarding Feature shall contain an extensible no reply condition timer. In other states the no reply condition timer shall not be included.

For call forwarding services other than call forwarding on no reply: The Extensible Forwarding Feature shall not contain a no reply condition timer.

If the VLR receives an Indication containing any Call Forwarding service codes which it does not support/allocate it returns them to the HLR in the parameter SS-Code List and discards the unsupported Call Forwarding service codes (no error is sent back).

Call barring information List

A list of Extensible Call barring information parameters (Extensible Call barring information is defined in subclause 5.6). It includes Call Barring services either at location updating or at restoration or when they are changed. Each Extensible Call barring information parameter shall be treated independently of all other parameters in the primitive.

The Extensible Call barring information shall include the SS-Code for an individual call barring supplementary service. The Extensible Call barring information shall contain one or more Extensible Call Barring Features (Extensible Call Barring Feature is defined in subclause 5.6).

The Extensible Call Barring Feature may include an Extensible Basic Service Group. This shall be interpreted according to the rules in subclause 6.8.1.4.

The Extensible Call Barring Feature shall contain an extensible SS-Status parameter.

If the VLR receives an Indication containing any Extensible Call Barring service codes which it does not support/allocate it returns them to the HLR in the parameter SS-Code List and discards the unsupported Extensible Call Barring service codes (no error is sent back).

CUG information List

A list of CUG information list parameters (CUG information is defined in subclause 5.6). It includes CUG information either at location updating or at restoration or when it is changed.

At location updating, restoration or when there is a change in CUG data, the HLR shall include the complete CUG-SubscriptionList and, if there are options per basic group, it shall also include the complete CUG-FeatureList. If there are not options per extensible basic service group the CUG-FeatureList shall not be included.
In any dialogue, the first insertSubscriberData message which contains CUG information shall include a non-empty CUG-SubscriptionList.

When the VLR receives CUG data it shall replace the stored CUG data with the received data set.

If CUG-FeatureList is omitted in the Insert Subscriber Data operation VLR shall interpret that no options per extensible basic service group exist, and then it shall apply the default values i.e. no outgoing access, no incoming access, no preferential CUG exists.

If CUG-Feature is received without preferential CUG, the VLR shall interpret that no preferential CUG applies.

If the VLR detects that there is overlapping in the information received within a dialogue, it shall send the error Unexpected Data Value.

Note that data consistency between CUG subscription data and CUG feature data is the responsibility of the HLR.

If the VLR does not support the CUG service it returns its code to the HLR in the parameter SS-Code List and discards the received information (no error is sent back).

**SS-Data List**

A list of Extensible SS-Data parameters (Extensible SS-Data is defined in subclause 5.6). It is sent for any other supplementary service than Call Forwarding, Call Barring, CUG and eMLPP either at location updating or at restoration or when they are changed. Each SS-Data parameter shall be treated independently of all other parameters in the primitive.

The Extensible SS-Data shall include the SS-Code for an individual supplementary service.

The Extensible SS-Data shall contain an Extensible SS-Status parameter and any subscription options that are applicable to the service defined by the SS-Code.

The SS-Data may include a Basic Service Group List. This shall be interpreted according to the rules in subclause 6.8.1.4.

If the VLR receives an Indication containing any supplementary service codes which it does not support/allocate it returns them to the HLR in the parameter SS-Code List and therefore discards the unsupported service codes received (no error is sent back).

**Operator Determined Barring General data**

If it is included in a Request/Indication, it includes all the Operator Determined Barring categories that may be applied to a subscriber registered in any PLMN. This parameter is only included in a Request/Indication when the parameter Subscriber Status is set to the value Operator Determined Barring. Note that all General Operator Determined Barring Categories shall be set to their actual status.

If the VLR receives an Indication containing Operator Determined Barring General Data which shows that the subscriber is subject to barring not supported / not allocated by the VLR, it returns Operator Determined Barring General Data in the response to the HLR to show the barring categories which are not supported / not allocated by the VLR.

**Operator Determined Barring HPLMN data**

It includes all the Operator Determined Barring categories that may be applied only to a subscriber registered in the HPLMN. Therefore, it shall only be transferred to the VLR when the subscriber is roaming into the HPLMN and when the parameter Subscriber Status is set to the value Operator Determined Barring. Note that all HPLMN Operator Determined Barring Categories shall be set to their actual status.

If Subscriber Status is set to the value Operator Determined Barring and no Operator Determined Barring HPLMN data is present then the VLR shall not apply any HPLMN specific ODB services to the subscriber.

**eMLPP Subscription Data**

If included in the Insert Subscriber Data request this parameter defines the priorities the subscriber might apply for a call (as defined in subclause 5.6). It contains both subparameters of eMLPP.
If the VLR does not support the eMLPP service it returns its code to the HLR in the parameter SS-Code List and therefore discards the received information (no error is sent back).

eMLPP subscription data that have been stored previously in a subscriber data record in the VLR are completely replaced by the new eMLPP subscription data received in a MAP_INSERT_SUBSCRIBER_DATA during either an Update Location or Restore Data procedure or a stand alone Insert Subscriber data procedure.

**Roaming Restriction Due To Unsupported Feature**

The HLR may decide to include this parameter in the request if certain services or features are indicated as not supported by the MSC/VLR (e.g. Advice of Charge Charging Level).

If this parameter is sent to the VLR the MSC area is restricted by the HLR and the VLR.

**Regional Subscription Data**

If included in the Insert Subscriber Data request this parameter defines the subscriber's subscription area for the addressed VLR (as defined in subclause 5.6). It contains the complete list of up to 10 Zone Codes that apply to a subscriber in the currently visited PLMN. The HLR shall send only those Zone Codes which are stored against the CC and NDC of the VLR to be updated.

**NOTE:** Support of this parameter is a network operator option and it will not be sent to networks which do not support Regional Subscription.

Regional subscription data that have been stored previously in a subscriber data record in the VLR are completely replaced by the regional subscription data received in an Insert Subscriber Data indication during either an Update Location or Restore Data procedure or a stand alone Insert Subscriber data procedure.

After the regional subscription data are inserted the VLR shall derive whether its location areas are allowed or not. If the whole MSC area is restricted it will be reported to HLR by returning the Regional Subscription Response.

The VLR returns a Regional Subscription Response indicating that a problem with the Zone Code has been detected in one of the following cases:

- Too Many Zone Codes: more than 10 Zone Codes are to be stored in the VLR;
- Regional Subscription Not Supported by the VLR;
- Zone Codes Conflict: the VLR detects that the zone codes indicate conflicting service permission for a location area.

Zone codes which have no mapping to location areas shall be ignored.

If a sequence of MAP_INSERT_SUBSCRIBER_DATA services is used during a dialogue, Regional Subscription Data shall be accepted only in one service. Regional Subscription Data received in a subsequent service shall be rejected with the error Unexpected Data Value.

If Regional Subscription Data are not included in any MAP_INSERT_SUBSCRIBER_DATA service, there is no restriction of roaming due to Regional Subscription.

**Voice Broadcast Data**

This parameter contains a list of group id's a user might have subscribed to; (VBS-Data is defined in subclause 5.6). It includes VBS information either at location updating or at restoration or when it is changed.

At location updating, restoration or when there is a change in VBS data, the HLR shall include the complete VBS-Data.

When the VLR receives VBS-Data within a dialogue it shall replace the stored VBS-data with the received data set. All subsequent VBS-data received within this dialogue shall be interpreted as add-on data.

If VBS-data is omitted in the Insert Subscriber Data operation the VLR shall keep the previously stored VBS data.

If the VLR detects that there is overlapping in the information received within a dialogue, it shall send the error Unexpected Data Value.

**Voice Group Call Data**
This parameter contains a list of group id's a user might have subscribed to; see subclause 5.6.

At location updating, restoration or when there is a change in VGCS data, the HLR shall include the complete VGCS-Data.

When the VLR receives VGCS-Data within a dialogue it shall replace the stored VGCS-Data with the received data set. All VGCS-Data received within this dialogue shall be interpreted as add-on data.

If VBCS-Data is omitted in the Insert Subscriber Data operation the VLR shall keep the previously stored VGCS-Data.

If the VLR detects that there is overlapping in the information received within a dialogue, it shall send the error Unexpected Data Value.

**SS-Code List**

The list of SS-Code parameters for the services that are provided to a subscriber but are not supported/allocated by the VLR (SS-Code is defined in subclause 5.6). The list can only include individual SS-Codes that were sent in the service request and SS-Codes for the eMLPP and/or CUG services if the above mentioned conditions, as described in eMLPP Subscription Data and/or CUG information List, are met (that is, eMLPP Subscription Data and/or CUG information List are received).

**Regional Subscription Response**

If included in the response this parameter indicates one of:

- MSC Area Restricted entirely because of regional subscription;
- Too Many Zone Codes to be inserted;
- Zone Codes Conflict;
- Regional Subscription not Supported by the VLR.

If the VLR determines after insertion of Regional Subscription Data that the entire MSC area is restricted, the VLR shall respond with a Regional Subscription Response indicating MSC Area Restricted. Otherwise MSC Area Restricted is not sent. The HLR shall check whether the current MSC area is no longer restricted.

**VLR CAMEL Subscription Info**

This parameter is sent for subscribers who have CAMEL services which are invoked in the MSC. In CAMEL phase 1 this parameter contains only the O-CSI. The VLR CAMEL Subscription Info is sent at location updating or when any information in the applicable CAMEL Subscription Info in the HLR has been changed. The entire set of CAMEL Subscription Info is sent. If a set of CAMEL Subscription Info is already stored in the VLR it is replaced by the received data.

**Supported CAMEL Phases**

The use of this parameter and the requirements for its presence are specified in GSM 03.78.

A VLR not supporting any CAMEL-Phase may omit this parameter.

**User error**

Only one of the following values is applicable:

- Unidentified subscriber;
- Data missing;
- Unexpected data value.
6.8.1.4 Basic service information related to supplementary services

A number of parameters that relate to supplementary services can be qualified by a Basic Service Group (or a Basic Service Group List). This subclause explains how this information is to be interpreted. Supplementary service parameters to which this subclause is applicable only apply to the basic service groups described in this subclause, and only those basic service groups shall be overwritten at the VLR.

The Basic Service Group (or Basic Service Group List) is optional.

If present the Basic Service Group (or the elements of the Basic Service Group List) shall be one of:

- an Elementary Basic Service Group for which the supplementary service is applicable to at least one basic service in the group; and to which the subscriber has a subscription to at least one basic service in the group;

- the group "All Teleservices" provided that the service is applicable to at least one teleservice and that the subscriber has a subscription to at least one teleservice that is in the same Elementary Basic Service Group as a teleservice to which the service is applicable;

- the group "All Bearer Services" provided that the service is applicable to at least one bearer service and that the subscriber has a subscription to at least one bearer service that is in the same Elementary Basic Service Group as a basic service to which the service is applicable.

If the Basic Service Group (or Basic Service Group List) is not present then the parameter shall apply to all Basic Service Groups.

If the basic service information is not a single Elementary Basic Service Group then the parameter shall be taken as applying individually to all the Elementary Basic Service Groups for which:

- the supplementary service is applicable to at least one basic service in the Basic Service Group; and

- the subscriber has a subscription to at least one basic service in the Basic Service Group.

The VLR is not required to store supplementary services data for Basic Service Groups that are not supported at the VLR.

6.8.2 MAP-DELETE-SUBSCRIBER-DATA service

6.8.2.1 Definition

This service is used by an HLR to remove certain subscriber data from a VLR if the subscription of one or more supplementary services or basic services is withdrawn. Note that this service is not used in case of erasure or deactivation of supplementary services.

It is a confirmed service and consists of the primitives shown in table 6.8/2.

6.8.2.2 Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic service List</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS-Code List</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Roaming Restriction Due To</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unsupported Feature</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Camel Subscription Info Withdraw</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regional Subscription Data</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VBS Group Indication</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VGCS Group Indication</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Regional Subscription Response</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td>C(=)</td>
<td>O</td>
</tr>
</tbody>
</table>
6.8.2.3 Parameter use

All parameters are described in subclause 5.6. The following clarifications are applicable:

Basic service List

A list of Extensible Basic service parameters (Extensible Basic service is defined in subclause 5.6). It is used when one, several or all basic services are to be withdrawn from the subscriber. If the VLR receives a value for an Extensible Basic Service which it does not support, it shall ignore that value.

SS-Code List

A list of SS-Code parameters (SS-Code is defined in subclause 5.6). It is used when several or all supplementary services are to be withdrawn from the subscriber.

There are three possible options:
- deletion of basic service(s);
  The parameter Basic service List is only included.
- deletion of supplementary service(s);
  The parameter SS-Code List is only included.
- deletion of basic and supplementary services;
  Both Basic service List and SS-Code List are included.

Roaming Restriction Due To Unsupported Feature

This parameter is used if Roaming Restriction Due To Unsupported Feature is deleted from the subscriber data. This may occur if unsupported features or services are removed from the subscriber data in the HLR.

If this parameter is sent the VLR shall check if the current Location Area is possibly allowed now.

CAMEL Subscription Info Withdraw

This parameter is used to indicate that CAMEL Subscription Info shall be deleted from the VLR. All CAMEL Subscription Info for the subscriber shall be deleted.

Regional Subscription Identifier

Contains one single Zone Code (as defined subclause 5.6) and is used if all Zone Codes shall be deleted from the subscriber data. When all the Zone Codes are deleted, the VLR shall check for its location areas whether they are allowed or not. If the whole MSC area is restricted, it will be reported to HLR by returning the Regional Subscription Response "MSC Area Restricted".

The binary coding of the Zone Code value received in a Delete Subscriber Data request shall not be checked by the VLR.

Note that support of this parameter is a network operator option and it shall not be sent to networks which do not support Regional Subscription.

If Regional Subscription is not supported by the VLR, the request for deletion of Zone Codes is refused by sending the Regional Subscription Response "Regional Subscription Not Supported" to the HLR.

If no Zone Codes are stored in the respective subscriber data record, the request for deleting all Zone Code information shall be ignored and no Regional Subscription Response shall be returned.

VBS Group Indication

Contains an indication (flag) which is used if all Group Id's shall be deleted from the subscriber data for the Voice Broadcast teleservice.

If VBS is not supported in the VLR or no Group Ids are stored for VBS in the respective subscriber record, the request for deletion of all Group Ids shall be ignored.
VGCS Group Indication

Contains an indication (flag) which is used if all Group Id's shall be deleted from the subscriber data for the Voice Group Call teleservice.

If VGCS is not supported in the VLR or no Group Ids are stored for VGCS in the respective subscriber record, the request for deletion of all Group Ids shall be ignored.

Regional Subscription Response

If included in the Delete Subscriber Data response this parameter indicates one of:

- MSC Area Restricted;
- Regional Subscription Not Supported.

User error

Only one of the following values is applicable:

- Unidentified subscriber;
- Data missing;
- Unexpected data value.

6.9 Identity management services

6.9.1 MAP-PROVIDE-IMSI service

6.9.1.1 Definition

This service is used by a VLR in order to get, via the MSC, the IMSI of a subscriber (e.g. when a subscriber has identified itself with a TMSI not allocated to any subscriber in the VLR).

It is a confirmed service and consists of the primitives shown in table 6.9/1.

6.9.1.2 Service primitives

Table 6.9/1: MAP-PROVIDE-IMSI

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

6.9.1.3 Parameter use

All parameters are described in subclause 5.6. The following clarifications are applicable:

IMSI

This parameter is received when the request is successfully carried out. It contains the requested IMSI.

User error

Only one of the following values is applicable:

- Absent subscriber.
6.9.2 MAP-FORWARD-NEW-TMSI service

6.9.2.1 Definition

This service is used by a VLR to allocate, via MSC, a new TMSI to a subscriber during an ongoing transaction (e.g. call set-up, location updating or supplementary services operation).

It is a confirmed service and consists of the primitives shown in table 6.9/2.

6.9.2.2 Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id TMSI</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
</tbody>
</table>

6.9.2.3 Parameter use

The parameter TMSI is described in subclause 5.6.

6.10 Fault recovery services

6.10.1 MAP_RESET service

6.10.1.1 Definition

This service is used by the HLR, after a restart, to indicate to a list of VLRs that a failure occurred.

The MAP_RESET service is a non-confirmed service using the service primitives defined in table 6.10/1

6.10.1.2 Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>HLR number</td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>HLR Id LIST</td>
<td>U</td>
<td>C(=)</td>
</tr>
</tbody>
</table>

6.10.1.3 Parameter definition and use

Invoke Id

See definition in subclause 5.6.1.

HLR number

See definition in subclause 5.6.2.

HLR Id LIST

The HLR Id List is a list of HLR Id. If the parameter is present in the indication, the VLR may base the retrieval of subscribers to be restored on their IMSI: the subscribers affected by the reset are those whose IMSI leading digits are equal to one of these numbers. If the parameter is absent, subscribers to be restored are those for which the OriginatingEntityNumber received at location updating time matches the equivalent parameter of the Reset Indication.
6.10.2 MAP_FORWARD_CHECK_SS_INDICATION service

6.10.2.1 Definition

This service may be used by an HLR as an implementation option, to indicate to a mobile subscriber that supplementary services parameters may have been altered, e.g. due to a restart. If received from the HLR, the VLR shall forward this indication to the MSC, which in turn forwards it to the MS. The HLR only sends this indication after successful completion of the subscriber data retrieval from HLR to VLR that ran embedded in a MAP_UPDATE_LOCATION procedure.

The MAP_FORWARD_CHECK_SS_INDICATION service is a non-confirmed service using the service primitives defined in table 6.10/2.

6.10.2.2 Service primitives

Table 6.10/2: MAP_FORWARD_CHECK_SS_INDICATION

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
</tr>
</tbody>
</table>

6.10.2.3 Parameter definition and use

Invoke Id

See definition in subclause 5.6.1.

6.10.3 MAP_RESTORE_DATA service

6.10.3.1 Definition

This service is invoked by the VLR on receipt of a MAP_PROVIDE_ROAMING_NUMBER indication for an unknown IMSI, or for a known IMSI with the indicator "Confirmed by HLR" set to "Not confirmed". The service is used to update the LMSI in the HLR, if provided, and to request the HLR to send all data to the VLR that are to be stored in the subscriber's IMSI record.

The MAP_RESTORE_DATA service is a confirmed service using the service primitives defined in table 6.10/3.

6.10.3.2 Service primitives

Table 6.10/3: MAP_RESTORE_DATA

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LMSI</td>
<td>U</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HLR number</td>
<td></td>
<td></td>
<td></td>
<td>C(=)</td>
</tr>
<tr>
<td>MS Not Reachable Flag</td>
<td></td>
<td></td>
<td></td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td></td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>
6.10.3.3 Parameter definitions and use

Invoke Id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2.

LMSI

See definition in subclause 5.6.2. It is an operator option to provide the LMSI from the VLR; it is mandatory for the HLR to support the LMSI handling procedures.

HLR number

See definition in subclause 5.6.2. The presence of this parameter is mandatory in case of successful outcome of the service.

MS Not Reachable Flag

See definition in subclause 5.6.8. This parameter shall be present in case of successful outcome of the service, if the "MS Not Reachable flag" was set in the HLR.

User error

In case of unsuccessful outcome of the service, an error cause shall be returned by the HLR. The following error causes defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- unknown subscriber;
- system failure;
- unexpected data value;
- data missing.

Provider error

For definition of provider errors see subclause 5.6.1.

6.11 Subscriber Information services

6.11.1 MAP-ANY-TIME-INTERROGATION service

6.11.1.1 Definition

This service is used by the gsmSCF, to request information (e.g. subscriber state and location) from the HLR at any time.
6.11.1.2 Service primitives

Table 6.11/1: Any_Time_Interrogation

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>Requested Info</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gsmSCF-Address</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSI</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSISDN</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location Information</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Subscriber State</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

6.11.1.3 Parameter definition and use

All parameters are described in subclause 5.6.

The HLR may be able to use the value of the parameter gsmSCF-address to screen an MAP_Any_Time_Interrogation indication.

The use of the parameters and the requirements for their presence are specified in GSM 03.78.

User error

This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- System Failure;
- Any Time Interrogation Not Allowed;
- Data Missing;
- Unexpected Data Value;
- Unknown Subscriber.

Provider error

These are defined in subclause 5.6.1.
6.11.2 MAP-PROVIDE-SUBSCRIBER-Info service

6.11.2.1 Definition

This service is used to request information (e.g. subscriber state and location) from the VLR at any time.

6.11.2.2 Service primitives

<table>
<thead>
<tr>
<th>Table 6.11/2: Provide_Subscriber_Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter name</td>
</tr>
<tr>
<td>Invoke id</td>
</tr>
<tr>
<td>Requested Info</td>
</tr>
<tr>
<td>IMSI</td>
</tr>
<tr>
<td>LMSI</td>
</tr>
<tr>
<td>Location Information</td>
</tr>
<tr>
<td>Subscriber State</td>
</tr>
<tr>
<td>User error</td>
</tr>
<tr>
<td>Provider error</td>
</tr>
</tbody>
</table>

6.11.2.3 Parameter definition and use

All parameters are defined in section 5.6. The use of these parameters and the requirements for their presence are specified in GSM 03.18.

User error

This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- Data Missing;
- Unexpected Data Value.

Provider error

These are defined in subclause 5.6.1.
7 Operation and maintenance services

7.1 Subscriber tracing services

7.1.1 MAP-ACTIVATE-TRACE-MODE service

7.1.1.1 Definition

This service is used between the HLR and the VLR to activate subscriber tracing in the VLR.

The MAP-ACTIVATE-TRACE-MODE service is a confirmed service using the primitives from table 7.1/1.

7.1.1.2 Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trace reference</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td></td>
</tr>
<tr>
<td>Trace type</td>
<td>M</td>
<td></td>
<td>M(=)</td>
<td></td>
</tr>
<tr>
<td>OMC Id</td>
<td>U</td>
<td>C(=)</td>
<td></td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

7.1.1.3 Parameter use

Invoke id

See definition in subclause 5.6.1.

IMSI

See definition in subclause 5.6.2. The IMSI is a mandatory parameter in a stand-alone operation.

Trace reference

See definition in subclause 5.6.10.

Trace type

See definition in subclause 5.6.10.

OMC Id

See definition in subclause 5.6.2. The use of this parameter is an operator option.

User error

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Unidentified Subscriber;
- Facility Not Supported;
- Tracing Buffer Full;
- System Failure;
- Unexpected Data Value;
- Data missing.

**Provider error**

For definition of provider errors see subclause 5.6.1.

### 7.1.2 MAP-DEACTIVATE-TRACE-MODE service

#### 7.1.2.1 Definition

This service is used between the VLR and the HLR for deactivating subscriber tracing in the VLR.

The MAP-DEACTIVATE-TRACE-MODE service is a confirmed service using the primitives from table 7.1/2.

#### 7.1.2.2 Service primitives

**Table 7.1/2: MAP-DEACTIVATE-TRACE-MODE**

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>=</td>
<td>M (=)</td>
<td>M (=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>C</td>
<td>(=)</td>
<td>C (=)</td>
<td>C (=)</td>
</tr>
<tr>
<td>Trace reference</td>
<td>M</td>
<td>(=)</td>
<td>M (=)</td>
<td>O</td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td>C</td>
<td>O</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 7.1.2.3 Parameter use

**Invoke id**

See definition in subclause 5.6.1.

**IMSI**

See definition in subclause 5.6.2. The IMSI is a mandatory parameter in a stand-alone operation.

**Trace reference**

See definition in subclause 5.6.10.

**User error**

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Unidentified Subscriber;
- Facility Not Supported;
- System Failure;
- Unexpected Data Value;
- Data missing.

**Provider error**

For definition of provider errors see subclause 5.6.1.

### 7.1.3 MAP-TRACE-SUBSCRIBER-ACTIVITY service

#### 7.1.3.1 Definition

This service is used between the VLR and the MSC to activate the subscriber tracing in the MSC.
The MAP-TRACE-SUBSCRIBER-ACTIVITY service is a non-confirmed service using the primitives from table 7.1/3.

### 7.1.3.2 Service primitives

#### Table 7.1/3: MAP-TRACE-SUBSCRIBER-ACTIVITY

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Trace reference</td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>Trace type</td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>OMC id</td>
<td>U</td>
<td>C(=)</td>
</tr>
</tbody>
</table>

#### 7.1.3.3 Parameter use

**Invoke id**

See definition in subclause 5.6.1.

**IMSI**

See definition in subclause 5.6.2. The controlling MSC shall provide either the IMSI or the IMEI to the servicing MSC.

**Trace reference**

See definition in subclause 5.6.10.

**Trace type**

See definition in subclause 5.6.10.

**OMC Id**

See definition in subclause 5.6.2. The use of this parameter is an operator option.

### 7.2 Other operation and maintenance services

#### 7.2.1 MAP-SEND-IMSI service

#### 7.2.1.1 Definition

This service is used by a VLR in order to fetch the IMSI of a subscriber in case of some Operation & Maintenance procedure where subscriber data are needed in the Visited PLMN and MSISDN is the only subscriber's identity known.

It is a confirmed service and consists of the primitive shown in figure 7.2/1.

#### 7.2.1.2 Service primitives

#### Table 7.2/1: MAP-SEND-IMSI

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>MSISDN</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSI</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td>O</td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>
7.2.1.3 Parameter use

All parameters are described in subclause 5.6. The following clarifications are applicable:

User error

Only one of the following values is applicable:

- Unknown subscriber;
- Unexpected data value;
- Data missing

8 Call handling services

8.1 MAP_SEND_ROUTING_INFORMATION service

8.1.1 Definition

This service is used between the Gateway MSC and the HLR. The service is invoked by the Gateway MSC to perform the interrogation of the HLR in order to route a call towards the called MS.

This is a confirmed service using the primitives listed in table 8.1/1.

8.1.2 Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>Interrogation Type</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GMSC Address</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSISDN</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OR Interrogation</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OR Capability</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUG Interlock</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>CUG Outgoing Access</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Number of Forwarding</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Signal Info</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supported CAMEL Phases</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Suppress T-CSI</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Suppression of Announcement</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Call Reference Number</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forwarding Reason</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic Service Group</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSI</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSRN</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forwarding Data</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forwarding Interrogation Required</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VMSC address</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GMSC Camel Subscription Info</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location Information</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subscriber State</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic Service Code</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUG Subscription Flag</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS-List</td>
<td>U</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
8.1.3 Parameter use

See subclause 5.6 for a definition of the parameters used in addition to the following. Note that:

- a conditional parameter whose use is defined only in GSM 03.78 shall be absent if the sending entity does not support CAMEL;
- a conditional parameter whose use is defined only in GSM 03.79 shall be absent if the sending entity does not support optimal routeing;
- a conditional parameter whose use is defined only in GSM 03.78 & GSM 03.79 shall be absent if the sending entity supports neither CAMEL nor optimal routeing.

Interrogation Type
See GSM 03.79 [99] for the use of this parameter.

GMSC address
The E.164 address of the GMSC.

MSISDN
This is the Mobile Subscriber ISDN number assigned to the called subscriber.

OR Interrogation
See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

OR Capability
See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

CUG Interlock
See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

CUG Outgoing Access
See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

Number of Forwarding
See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

Network Signal Info
See GSM 03.18 [97] for the conditions for the presence of the components of this parameter.

Supported CAMEL Phases
The use of this parameter and the requirements for its presence are specified in GSM 03.78

T-CSI Suppression
The use of this parameter and the requirements for its presence are specified in GSM 03.78

Suppression Of Announcement
The use of this parameter and the requirements for its presence are specified in GSM 03.78

Call Reference Number
The use of this parameter and the conditions for its presence are specified in GSM 03.78 [98] and GSM 03.79 [99].

Forwarding Reason
See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.
Basic Service Group
See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

IMSI
See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

MSRN
See GSM 03.18 [97] and GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

Forwarding Data
This parameter includes the forwarded-to number, the forwarding option Notification to calling party and the forwarding reason, and can include the forwarded-to subaddress. See GSM 03.18 [97] and GSM 03.79 [99] for the conditions for the presence of its components.

Forwarding Interrogation Required
See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

VMSC address
See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

GMSC CAMEL Subscription Info
The use of this parameter and the requirements for its presence are specified in GSM 03.78.

Location Information
The use of this parameter and the requirements for its presence are specified in GSM 03.78.

Subscriber State
The use of this parameter and the requirements for its presence are specified in GSM 03.78.

CUG Subscription Flag
The use of this parameter and the requirements for its presence are specified in GSM 03.78.

SS-List
This parameter includes SS-codes and will be returned as an operator option. The HLR shall not send PLMN-specific SS-codes across PLMN boundaries. However if the GMSC receives PLMN-specific SS-codes from a foreign PLMN’s HLR the GMSC may ignore it. If the GMSC attempts to process the PLMN specific SS codes, this may lead to unpredictable behaviour but the GMSC shall continue call processing.

Basic Service Code
The use of this parameter and the requirements for its presence are specified in GSM 03.78.

If the CAMEL service is not involved, this parameter includes the basic service code and will be returned as an operator option. The HLR shall not send a PLMN-specific Basic Service Code across PLMN boundaries. However if the GMSC receives a PLMN-specific Basic Service Code from a foreign PLMN’s HLR the GMSC may ignore it. If the GMSC attempts to process the PLMN specific Basic Service codes, this may lead to unpredictable behaviour but the GMSC shall continue call processing.

User error
This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- Unknown Subscriber;
- Number changed;
- Call Barred;
This error will indicate that either incoming calls are barred for this MS or that calls are barred due to Operator
Determined Barring (see GSM 02.41 for a definition of this network feature).

- CUG Reject;
  The value of this error cause will indicate the reason for CUG Reject.
- Bearer Service Not Provisioned;
- Teleservice Not Provisioned;
  A subscription check has been performed and the call has not passed the check due to incompatibility with
  regard to the requested service. Depending on the nature of the incompatibility, either of these messages will be
  returned.
- Facility Not Supported;
- Absent Subscriber;
  This indicates that the location of the MS is not known (either the station is not registered and there is no
  location information available or the Provide Roaming Number procedure fails due to IMSI detached flag being
  set), or the GMSC requested forwarding information with a forwarding reason of not reachable, and the call
  forwarding on MS not reachable service is not active.
- Busy Subscriber;
  This indicates that Call Forwarding on Busy was not active for the specified basic service group when the
  GMSC requested forwarding information with a forwarding reason of busy.
- No Subscriber Reply;
  This indicates that Call Forwarding on No Reply was not active for the specified basic service group when the
  GMSC requested forwarding information with a forwarding reason of no reply.
- OR Not Allowed;
  This indicates that the HLR is not prepared to accept an OR interrogation from the GMSC, or that calls to the
  specified subscriber are not allowed to be optimally routed.
- Forwarding Violation;
- System Failure;
- Data Missing;
- Unexpected Data Value.

See subclause 5.6 for a definition of these errors.

Provider error

These are defined in subclause 5.6.

### 8.2 MAP_PROVIDE_ROAMING_NUMBER service

#### 8.2.1 Definition

This service is used between the HLR and VLR. The service is invoked by the HLR to request a VLR to send back a
roaming number to enable the HLR to instruct the GMSC to route an incoming call to the called MS.

This is a confirmed service which uses the Primitives described in table 8.2/1.
8.2.2 Service primitives

Table 8.2/1: MAP_PROVIDE_ROAMING_NUMBER parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSC Number</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSISDN</td>
<td>U</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LMSI</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GSM Bearer Capability</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Network Signal Info</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Suppression Of Announcement</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Call Reference Number</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GMSC Address</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OR Interrogation</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Roaming Number</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td>O</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8.2.3 Parameter use

See subclause 5.6 for a definition of the parameters used, in addition to the following. Note that:

- a conditional parameter whose use is defined only in GSM 03.78 shall be absent if the sending entity does not support CAMEL;
- a conditional parameter whose use is defined only in GSM 03.79 shall be absent if the sending entity does not support optimal routing;
- a conditional parameter whose use is defined only in GSM 03.78 & GSM 03.79 shall be absent if the sending entity supports neither CAMEL nor optimal routing.

IMSI

This is the IMSI of the called Subscriber.

MSC Number

This is the ISDN number assigned to the MSC currently serving the MS. The MSC number will have been stored in the HLR as provided at location updating.

MSISDN

See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

LMSI

See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

GSM Bearer Capability

See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

This information is passed according to the rules specified in TS GSM 09.07.

There may be two GSM Bearer Capabilities supplied.

Network Signal Info

See GSM 03.18 [97] for the conditions for the presence of the components of this parameter.

Suppression Of Announcement

The use of this parameter and the requirements for its presence are specified in GSM 03.78.
Call Reference Number

The use of this parameter and the conditions for its presence are specified in GSM 03.78 [98] and GSM 03.79 [99].

GMSC Address

The use of this parameter and the conditions for its presence are specified in GSM 03.78 [98] and GSM 03.79 [99].

OR Interrogation

See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

Roaming Number

See GSM 03.18 [97] for the use of this parameter and the conditions for its presence.

User error

This parameter is sent by the responder when an error is detected and if present, takes one of the following values:

- Absent Subscriber;
  This error will be returned if the IMSI detach flag is set.
- No Roaming Number Available;
- OR Not Allowed;
  This indicates that the MAP_PROVIDE_ROAMING_NUMBER indication included the OR interrogation indicator, but the VLR does not support optimal routeing.
- Facility Not Supported;
- System Failure;
- Data Missing;
- Unexpected Data Value.

See subclause 5.6 for a definition of these reasons.

Provider error

These are defined in subclause 5.6.

8.3 MAP_RESUME_CALL_HANDLING service

8.3.1 Definition

This service is used between the terminating VMSC and the GMSC. The service is invoked by the terminating VMSC to request the GMSC to resume handling the call and forward it to the specified destination.

This is a confirmed service which uses the Primitives listed in table 8.3/1.
8.3.2 Service primitives

Table 8.3/1: MAP_RESUME_CALL_HANDLING parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>Call Reference Number</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>Basic Service Group</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>Forwarding Data</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>CUG Interlock</td>
<td>C</td>
<td>C(=)</td>
<td>C(=)</td>
<td>C(=)</td>
</tr>
<tr>
<td>CUG Outgoing Access</td>
<td>C</td>
<td>C(=)</td>
<td>C(=)</td>
<td>C(=)</td>
</tr>
<tr>
<td>O-CSI</td>
<td>C</td>
<td>C(=)</td>
<td>C(=)</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td>C(=)</td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td>C</td>
<td>C(=)</td>
<td>C(=)</td>
<td>C(=)</td>
</tr>
</tbody>
</table>

8.3.3 Parameter use

See subclause 5.6 for a definition of the parameters used, in addition to the following.

Call Reference Number
See GSM 03.79 [99] for the use of this parameter.

Basic Service Group
See GSM 03.79 [99] for the use of this parameter.

IMSI
This is the IMSI of the forwarding Subscriber.

Forwarding Data
Includes the forwarded-to number, the forwarding reason, an indication of whether the calling party is to be notified that the call has been forwarded and possibly a forwarded-to subaddress.

CUG Interlock
See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

CUG Outgoing Access
See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

O-CSI
See GSM 03.79 [99] for the use of this parameter and the conditions for its presence.

For CAMEL phase 1, the O-CSI shall contain only one set of O-BCSM TDP data.

User error
This parameter is sent by the responder when an error is detected and if present, takes one of the following values:
- Optimal Routeing not allowed;
- Forwarding failed.

Provider error
These are defined in subclause 5.6.
9 Supplementary services related services

9.1 MAP_REGISTER_SS service

9.1.1 Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to register data related to a supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and consists of four service primitives.

9.1.2 Service primitives

The service primitives are shown in table 9.1/1.

<table>
<thead>
<tr>
<th>Table 9.1/1: MAP_REGISTER_SS parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter name</td>
</tr>
<tr>
<td>Request</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Invoke id</td>
</tr>
<tr>
<td>SS-Code</td>
</tr>
<tr>
<td>Basic service</td>
</tr>
<tr>
<td>Forwarded-to number with subaddress</td>
</tr>
<tr>
<td>No reply condition time</td>
</tr>
</tbody>
</table>

9.1.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to register.

Basic service

This parameter indicates for which basic service group the supplementary service is to be registered. If it is not included, the registration request applies to all basic services.

Forwarded-to number with subaddress

This parameter is obligatory if the registration applies to one or more call forwarding supplementary services. It can optionally include a sub-address.

No reply condition time

This parameter is included if the registration applies to the Call Forwarding on No Reply supplementary service (or a superset of this service) and the mobile subscriber supplies a value for this time.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the registration request concerned one or a group of Call Forwarding supplementary services.

User error
This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in subclause 5.6.1:

- System failure;
- Data missing;
- Unexpected data value;
- Call Barred;
- Bearer service not provisioned;
  This error is returned only if not even a subset of the requested bearer service group has been subscribed to.
- Teleservice not provisioned;
  This error is returned only if not even a subset of the requested teleservice group has been subscribed to.
- Illegal SS operation;
- SS error status;
- SS incompatibility.

Provider error
See subclause 5.6.1 for the use of this parameter.

9.2  MAP_ERASE_SS service

9.2.1  Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to erase data related to a supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and consists of four service primitives.

9.2.2  Service primitives

The service primitives are shown in table 9.2/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>SS-Code</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>Basic service</td>
<td>C</td>
<td>C(=)</td>
<td>C(=)</td>
<td>C(=)</td>
</tr>
<tr>
<td>Forwarding information</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9.2.3  Parameter use

Invoke id
See subclause 5.6.1 for the use of this parameter.

SS-Code
This parameter indicates the supplementary service which the mobile subscriber wants to erase.

Basic service
This parameter indicates for which basic service group the supplementary service should be erased. If it is not included, the erasure request applies to all basic services.

**Forwarding information**

This parameter is returned by the responder at successful outcome of the service, if the erasure request concerned one or a group of Call Forwarding supplementary services.

**User error**

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Bearer service not provisioned;
  
  This error is returned only if not even a subset of the requested bearer service group has been subscribed to.
- Teleservice not provisioned;
  
  This error is returned only if not even a subset of the requested teleservice group has been subscribed to.
- Call Barred;
- Illegal SS operation;
- SS error status.

**Provider error**

See subclause 5.6.1 for the use of this parameter.

### 9.3 MAP_ACTIVATE_SS service

#### 9.3.1 Definition

This service is used between the MSC and the VLR and between the VLR and the HLR to activate a supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and consists of four service primitives.

#### 9.3.2 Service primitives

The service primitives are shown in table 9.3/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>SS-Code</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic service</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forwarding information</td>
<td></td>
<td>C</td>
<td>C(=)</td>
<td></td>
</tr>
<tr>
<td>Call barring information</td>
<td></td>
<td>C</td>
<td>C(=)</td>
<td></td>
</tr>
<tr>
<td>SS-Data</td>
<td></td>
<td>C</td>
<td>C(=)</td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td>C</td>
<td>C(=)</td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>
9.3.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to activate.

Basic service

This parameter indicates for which basic service groups the requested supplementary service(s) should be activated. If it is not included, the activation request applies to all basic services.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned Call Forwarding.

Call barring information

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned Call Barring.

SS-Data

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned for example Call Waiting.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Bearer service not provisioned;
  
  This error is returned only if not even a subset of the requested bearer service group has been subscribed to.
- Teleservice not provisioned;
  
  This error is returned only if not even a subset of the requested teleservice group has been subscribed to.
- Call Barred;
- Illegal SS operation;
- SS error status;
- SS subscription violation;
- SS incompatibility;
- Negative PW check;
- Number Of PW Attempts Violation.

Provider error

See subclause 5.6.1 for the use of this parameter.
9.4 MAP_DEACTIVATE_SS service

9.4.1 Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR to deactivate a supplementary service. The VLR will relay the message to the HLR.

The service is a confirmed service and consists of four service primitives.

9.4.2 Service primitives

The service primitives are shown in table 9.4/1.

Table 9.4/1: MAP_DEACTIVATE_SS parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>SS-Code</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic service</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forwarding information</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Call barring information</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS-Data</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td>O</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9.4.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates the supplementary service which the mobile subscriber wants to deactivate.

Basic service

This parameter indicates for which basic service group the requested supplementary service(s) should be deactivated. If it is not included the deactivation request applies to all basic services.

Forwarding information

This parameter is returned by the responder at successful outcome of the service, if the deactivation request concerned one or a group of Call Forwarding supplementary services.

Call barring information

This parameter is returned by the responder at successful outcome of the service, if the activation request concerned one or a group of Call Barring supplementary services.

SS-Data

This parameter is returned by the responder at successful outcome of the service, for example if the deactivation request concerned the Call Waiting supplementary service.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Bearer service not provisioned;
  This error is returned only if not even a subset of the requested bearer service group has been subscribed to.
- Teleservice not provisioned;
  This error is returned only if not even a subset of the requested teleservice group has been subscribed to.
- Call Barred;
- Illegal SS operation;
- SS error status;
- SS subscription violation;
- Negative PW check;
- Number Of PW Attempts Violation.

Provider error
See subclause 5.6.1 for the use of this parameter.

9.5 MAP_INTERROGATE_SS service

9.5.1 Definitions
This service is used between the MSC and the VLR and between the VLR and the HLR to retrieve information related to a supplementary service. The VLR will relay the message to the HLR if necessary.

The service is a confirmed service and consists of four service primitives.

9.5.2 Service primitives
The service primitives are shown in table 9.5/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>SS-Code</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic service</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SS-Status</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Basic service Group LIST</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Forwarding feature LIST</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>CLI restriction Info</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

9.5.3 Parameter use
For additional information on parameter use refer to the GSM 04.8x and 04.9x-series of technical specifications.

Invoke id
See subclause 5.6.1 for the use of this parameter.
SS-Code

The mobile subscriber can only interrogate a single supplementary service per service request.

Basic service

This parameter indicates for which basic service group the given supplementary service is interrogated. If it is not included, the interrogation request applies to all basic services.

SS-Status

This parameter is included by the responder if:

- the interrogated supplementary service can only be subscribed for all applicable basic services simultaneously;
  or
- the interrogated supplementary service is not active for any of the interrogated basic services.

Basic service group LIST

This parameter LIST is used to include one or a series of basic service groups for which the interrogated supplementary service is active. If the interrogated supplementary service is not active for any of the interrogated (and provisioned) basic service groups, the SS-Status parameter is returned.

Forwarding feature LIST

The forwarding feature parameter is described in subclause 5.6.4. A list of one or more forwarding features is returned by the responder when the interrogation request applied to Call Forwarding supplementary service.

If no basic service code parameter is provided within this sequence, the forwarding feature parameter applies to all provisioned basic services.

CLI restriction Info

The CLI-RestrictionInfo parameter is returned by the responder when the interrogation request applies to the CLIR supplementary service.

User error

This error is sent by the responder upon unsuccessful outcome of the interrogation service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Bearer Service not provisioned;
  This error is returned only if not even a subset of the interrogated bearer services are provided.
- Teleservice not provisioned;
  This error is returned only if not even a subset of the interrogated teleservices are provided.
- Call Barred;
- Illegal SS operation;
- SS not available.

Provider error

See subclause 5.6.1 for the use of this parameter.
9.6 MAP_INVOKE_SS service

9.6.1 Definitions

This service is used between the MSC and the VLR to check the subscriber's subscription to a given supplementary service in the VLR, in connection with in-call invocation of that supplementary service, i.e. after the call set-up phase is finished. For supplementary service invocation during call set-up phase, please refer to the call handling descriptions.

The service is a confirmed service and consists of four service primitives.

9.6.2 Service primitives

The service primitives are shown in table 9.6/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>SS-Code</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>Basic service</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td>O</td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

9.6.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This SS-Code can only refer to a single supplementary service, e.g. the Call Hold or Multi Party supplementary services.

Basic service

This parameter indicates for which basic service the supplementary service invocation is required.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values:

- System Failure;
- Data Missing;
- Unexpected data value;
- Call Barred;
- Illegal SS operation;
- SS error status;
- SS not available.

Provider error

See subclause 5.6.1 for the use of this parameter.
9.7 MAP_REGISTER_PASSWORD service

9.7.1 Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR if the mobile subscriber requests to register a new password. The VLR will relay the message to the HLR.

The service is a confirmed service and consists of four service primitives.

9.7.2 Service primitives

The service primitives are shown in table 9.7/1.

Table 9.7/1: MAP_REGISTER_PASSWORD parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>SS-Code</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New password</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9.7.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

SS-Code

This parameter indicates for which supplementary service(s) the password should be registered.

New Password

See subclause 5.6.4 for the use of this parameter.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values, defined in subclause 5.6.1:

- System failure;
- Data Missing;
- Unexpected data value;
- Call Barred;
- SS subscription violation;
- Password registration failure;
- Negative PW check;
- Number Of PW Attempts Violation.

Provider error

See subclause 5.6.1 for the use of this parameter.
9.8 MAP_GET_PASSWORD service

9.8.1 Definitions

This service is used between the HLR and the VLR and between the VLR and the MSC when the HLR receives a request from the mobile subscriber for an operation on a supplementary service which requires a password from the subscriber. The VLR will relay the message to the MSC.

The service is a confirmed service and consists of four service primitives.

9.8.2 Service primitives

The service primitives are shown in table 9.8/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>Linked id</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Guidance info</td>
<td>M</td>
<td>M(=)</td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>Current password</td>
<td></td>
<td></td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9.8.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

Linked Id

See subclause 5.6.1 for the use of this parameter. If the MAP GET PASSWORD service is used in conjunction with the MAP REGISTER PASSWORD service, this parameter must be present; otherwise it must be absent.

Guidance info

See subclause 5.6.4 for the use of this parameter.

Current password

See subclause 5.6.4 for the use of this parameter.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.9 MAP_PROCESS_UNSTRUCTURED_SS_REQUEST service

9.9.1 Definitions

This service is used between the MSC and the VLR and between the VLR and the HLR to relay information in order to allow unstructured supplementary service operation.

The MAP_PROCESS_UNSTRUCTURED_SS_REQUEST service is a confirmed service using the primitives from table 9.9/1.
9.9.2 Service primitives

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>USSD Data Coding Scheme</td>
<td>M</td>
<td>M(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>USSD String</td>
<td>M</td>
<td>M(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9.9.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

USSD Data Coding Scheme:

See subclause 5.6.4 for the use of this parameter. The presence of the parameter in the response is dependent on the unstructured supplementary service application. If this parameter is present, then the USSD String parameter has to be present.

USSD String:

See subclause 5.6.1 for the use of this parameter. The presence of the parameter in the response is dependent on the unstructured supplementary service application. If this parameter is present, then the USSD Data Coding Scheme parameter has to be present.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in subclause 5.6.1:

- System failure;
- Data missing;
- Unexpected data value;

This error is returned by the responder if it is not able to deal with the contents of the USSD string.

- Call Barred;
- Unknown Alphabet.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.10 MAP_UNSTRUCTURED_SS_REQUEST service

9.10.1 Definitions

This service is used between the HLR and the VLR and between the VLR and the MSC when the invoking entity requires information from the mobile user, in connection with unstructured supplementary service handling.

The MAP_UNSTRUCTURED_SS_REQUEST service is a confirmed service using the primitives from table 9.10/1.

9.10.2 Service primitives

The service primitives are shown in table 9.10/1.
Table 9.10/1: MAP_UNSTRUCTURED_SS_REQUEST parameters

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>USSD Data Coding Scheme</td>
<td>M</td>
<td>M(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>USSD String</td>
<td>M</td>
<td>M(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

9.10.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

USSD Data Coding Scheme:

See subclause 5.6.4 for the use of this parameter. The presence of the parameter in the response is dependent on the mobile user's MMI input. If this parameter is present, then the USSD String parameter has to be present.

USSD String:

See subclause 5.6.1 for the use of this parameter. The presence of the parameter in the response is dependent on the mobile user's MMI input. If this parameter is present, then the USSD Data Coding Scheme parameter has to be present.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in subclause 5.6.1:

- System failure;
- Data missing;
- Unexpected data value;
  This error is returned by the responder if it is not able to deal with the contents of the USSD string.
- Absent Subscriber;
- Illegal Subscriber;
  This error indicates that delivery of the unstructured supplementary service data failed because the MS failed authentication.
- Illegal Equipment;
- USSD Busy;
- Unknown Alphabet.

Provider error

See subclause 5.6.1 for the use of this parameter.

9.11 MAP_UNSTRUCTURED_SS_NOTIFY service

9.11.1 Definitions

This service is used between the HLR and the VLR and between the VLR and the MSC when the invoking entity requires a notification to be sent to the mobile user, in connection with unstructured supplementary services handling.

The MAP_UNSTRUCTURED_SS_NOTIFY service is a confirmed service using the primitives from table 9.11/1.
9.11.2 Service primitives

The service primitives are shown in table 9.11/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>USSD Data Coding Scheme</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>USSD String</td>
<td>M</td>
<td>M(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9.11.3 Parameter use

Invoke id

See subclause 5.6.1 for the use of this parameter.

USSD Data Coding Scheme:

See subclause 5.6.4 for the use of this parameter.

USSD String:

See subclause 5.6.1 for the use of this parameter.

User error

This parameter is sent by the responder upon unsuccessful outcome of the service, and then takes one of the following values defined in subclause 5.6.1:

- System failure;
- Data missing;
- Unexpected data value;
  This error is returned by the responder if it is not able to deal with the contents of the USSD string.
- Absent Subscriber;
- Illegal Subscriber;
  This error indicates that delivery of the unstructured supplementary service data failed because the MS failed authentication.
- Illegal Equipment;
- USSD Busy;
- Unknown Alphabet.

Provider error

See subclause 5.6.1 for the use of this parameter.
10 Short message service management services

10.1 MAP-SEND-ROUTING-INFO-FOR-SM service

10.1.1 Definition

This service is used between the gateway MSC and the HLR to retrieve the routing information needed for routing the short message to the servicing MSC.

The MAP-SEND-ROUTING-INFO-FOR-SM is a confirmed service using the primitives from table 10.1/1.

10.1.2 Service primitives

The service primitives are shown in table 10.1/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>MSISDN</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SM-RP-PRI</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Centre Address</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSI</td>
<td></td>
<td>C</td>
<td>C(=)</td>
<td></td>
</tr>
<tr>
<td>MSC Number</td>
<td></td>
<td>C</td>
<td>C(=)</td>
<td></td>
</tr>
<tr>
<td>LMSI</td>
<td></td>
<td>C</td>
<td>C(=)</td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td>C</td>
<td>C(=)</td>
<td>O</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td>C</td>
<td>C(=)</td>
<td></td>
</tr>
</tbody>
</table>

10.1.3 Parameter use

**Invoke id:**

See definition in subclause 5.6.1.

**MSISDN:**

See definition in subclause 5.6.2.

**SM-RP-PRI:**

See definition in subclause 5.6.8.

**Service Centre Address:**

See definition in subclause 5.6.2.

**IMSI:**

See definition in subclause 5.6.2. The presence of this parameter is mandatory in a successful case.

**MSC Number:**

See definition in subclause 5.6.2. This parameter is provided in a successful response.

**LMSI:**

See definition in subclause 5.6.2. It is an operator option to provide this parameter from the VLR; it is mandatory for the HLR to include the LMSI in a successful response, if the VLR has used the LMSI.

**User error:**

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:
10.2 MAP-MO-FORWARD-SHORT-MESSAGE service

10.2.1 Definition

This service is used between the serving MSC and the SMS Interworking MSC to forward mobile originated short messages.

The MAP-MO-FORWARD-SHORT-MESSAGE service is a confirmed service using the service primitives given in table 10.2/1.

10.2.2 Service primitives

The service primitives are shown in table 10.2/1.

Table 10.2/1: MAP-MO-FORWARD-SHORT-MESSAGE

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>SM RP DA</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SM RP OA</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SM RP UI</td>
<td>M</td>
<td>M(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td></td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

10.2.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

SM RP DA:

See definition in subclause 5.6.8.

In the mobile originated SM transfer this parameter contains the Service Centre address received from the mobile station.

SM RP OA:

See definition in subclause 5.6.8.

The MSISDN received from the VLR is inserted in this parameter in the mobile originated SM transfer.
SM RP UI:
See definition in subclause 5.6.8. The short message transfer protocol data unit received from the Service Centre is inserted in this parameter.

User error:
The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Facility Not Supported;
- System Failure;
- SM Delivery Failure;
  - The reason of the SM Delivery Failure can be one of the following in the mobile originated SM:
    - unknown Service Centre address;
    - Service Centre congestion;
    - invalid Short Message Entity address;
    - subscriber not Service Centre subscriber;
    - protocol error.
- Unexpected Data Value.

Provider error:
For definition of provider errors see subclause 5.6.1.

10.3 MAP-REPORT-SM-DELIVERY-STATUS service

10.3.1 Definition
This service is used between the gateway MSC and the HLR. The MAP-REPORT-SM-DELIVERY-STATUS service is used to set the Message Waiting Data into the HLR or to inform the HLR of successful SM transfer after polling. This service is invoked by the gateway MSC.

The MAP-REPORT-SM-DELIVERY-STATUS service is a confirmed service using the service primitives given in table 10.3/1.

10.3.2 Service primitives
The service primitives are shown in table 10.3/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>MSISDN</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Centre Address</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SM Delivery Outcome</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Absent Subscriber</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diagnostic SM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSISdn-Alert</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
10.3.3 Parameter use

Invoke id:
See definition in subclause 5.6.1.

MSISDN:
See definition in subclause 5.6.2.

Service Centre Address:
See definition in subclause 5.6.2.

SM Delivery Outcome:
See definition in subclause 5.6.8. This parameter indicates the status of the mobile terminated SM delivery.

Absent Subscriber Diagnostic SM:
See definition in subclause 5.6.8.

MSIsdn-Alert:
See definition in subclause 5.6.2. This parameter shall be present in case of unsuccessful delivery, when the MSISDN received in the operation is different from the stored MSIsdn-Alert; the stored MSIsdn-Alert is the value that is returned to the gateway MSC.

User error:
The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Unknown Subscriber;
- Message Waiting List Full;
- Unexpected Data Value;
- Data missing.

Provider error:
For definition of provider errors see subclause 5.6.1.

10.4 MAP-READY-FOR-SM service

10.4.1 Definition

This service is used between the MSC and VLR and as well between the VLR and the HLR. The MSC initiates this service if a subscriber indicates memory available situation. The VLR uses the service to indicate this to the HLR.

The VLR initiates this service if a subscriber, whose message waiting flag is active in the VLR, has radio contact in the MSC.

The MAP-READY-FOR-SM service is a confirmed service using the primitives from table 10.4/1.
10.4.2 Service primitives

The service primitives are shown in table 10.4/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>IMSI</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TMSI</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alert Reason</td>
<td>M</td>
<td>M(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td>M</td>
<td>M(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

10.4.3 Parameter use

**Invoke id:**
See definition in subclause 5.6.1.

**IMSI:**
See definition in subclause 5.6.2. The IMSI is used always between the VLR and the HLR. Between the MSC and the VLR the identification can be either IMSI or TMSI.

**TMSI:**
See definition in subclause 5.6.2. The identification can be either IMSI or TMSI between MSC and VLR.

**Alert Reason:**
See definition in subclause 5.6.8. This parameter indicates if the mobile subscriber is present or the MS has memory available.

**User error:**
The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:
- Unknown Subscriber;
- Facility Not Supported;
- System Failure;
- Unexpected Data Value;
- Data missing;

**Provider error:**
For definition of provider errors see subclause 5.6.1.

10.5 MAP-ALERT-SERVICE-CENTRE service

10.5.1 Definition

This service is used between the HLR and the interworking MSC. The HLR initiates this service, if the HLR detects that a subscriber, whose MSISDN is in the Message Waiting Data file, is active or the MS has memory available.

The MAP-ALERT-SERVICE-CENTRE service is a confirmed service using the primitives from table 10.5/1.
10.5.2 Service primitives

The service primitives are shown in table 10.5/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>MSIsdn-Alert</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Centre Address</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td>C</td>
<td>C(=)</td>
<td></td>
<td>O</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10.5.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

MSIsdn-Alert:

See definition in subclause 5.6.2. The provided MSISDN shall be the one which is stored in the Message Waiting Data file.

Service Centre Address:

See definition in subclause 5.6.2.

User error:

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- System Failure;
- Unexpected Data Value;
- Data missing.

Provider error:

For definition of provider errors see subclause 5.6.1.

10.6 MAP-INFORM-SERVICE-CENTRE service

10.6.1 Definition

This service is used between the HLR and the gateway MSC to inform the Service Centre which MSISDN number is stored in the Message Waiting Data file. If the stored MSISDN number is not the same than the one received from the gateway MSC in the MAP-SEND-ROUTING-INFO-FOR-SM service primitive the stored MSISDN number is included in the message.

Additionally the status of MCEF and MNRF flags and the inclusion of the particular Service Centre address in the Message Waiting Data list is informed to the gateway MSC when appropriate.

The MAP-INFORM-SERVICE-CENTRE service is a non-confirmed service using the primitives from table 10.6/1.
10.6.2 Service primitives

The service primitives are shown in table 10.6/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
</tr>
<tr>
<td>MSISdn-Alert</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>MWD Status</td>
<td>C</td>
<td>C(=)</td>
</tr>
</tbody>
</table>

10.6.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

MSISdn-Alert:

See definition in subclause 5.6.2 This parameter refers to the MSISDN stored in a Message Waiting Data file in the HLR.

MWD Status:

See definition in subclause 5.6.8. This parameter indicates the status of the MCEF and MNRF flags and the status of the particular SC address presence in the Message Waiting Data list.

10.7 MAP-SEND-INFO-FOR-MT-SMS service

10.7.1 Definition

This service is used between the MSC and the VLR. The service is invoked by the MSC receiving an mobile terminated short message to request subscriber related information from the VLR.

The MAP-SEND-INFO-FOR-MT-SMS service is a confirmed service using the primitives from table 10.7/1.

10.7.2 Service primitives

The service primitives are shown in table 10.7/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>SM RP DA</td>
<td>M</td>
<td>M(=)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSISDN</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

10.7.3 Parameter use

Invoke id:

See definition in subclause 5.6.1.

SM RP DA:

See definition in subclause 5.6.8. This parameter shall contain either an IMSI or a LMSI.
MSISDN:
See definition in subclause 5.6.2.

User error:
The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:
- Unknown subscriber;
- Unidentified subscriber;
- Absent subscriber;
- Unexpected Data Value;
- Data Missing;
- Illegal subscriber;
- Illegal equipment;
- Subscriber busy for MT SMS;
- System Failure.

Provider error:
For definition of provider errors see subclause 5.6.1.

10.8 MAP-SEND-INFO-FOR-MO-SMS service

10.8.1 Definition
This service is used between the MSC and the VLR. The service is invoked by the MSC which has to handle a mobile originated short message request to request the subscriber related information from the VLR.

The MAP-SEND-INFO-FOR-MO-SMS service is a confirmed service using the primitives from table 10.8/1.

10.8.2 Service primitives
The service primitives are shown in table 10.8/1.

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>Service Centre Address</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSISDN</td>
<td>C</td>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User error</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10.8.3 Parameter use
Invoke id:
See definition in subclause 5.6.1.

Service Centre Address:
See definition in subclause 5.6.2.
MSISDN:
See definition in subclause 5.6.2.

User error:
The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:
- Teleservice Not Provisioned;
- Call Barred;
- Unexpected Data Value;
- Data Missing.

Provider error:
For definition of provider errors see subclause 5.6.1.

10.9 MAP-MT-FORWARD-SHORT-MESSAGE service

10.9.1 Definition
This service is used between the gateway MSC and the servicing MSC to forward mobile mobile terminated short messages.
The MAP-MT-FORWARD-SHORT-MESSAGE service is a confirmed service using the service primitives given in table 10.9/1.

10.9.2 Service primitives
The service primitives are shown in table 10.9/1.

Table 10.9/1: MAP-MT-FORWARD-SHORT-MESSAGE

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Request</th>
<th>Indication</th>
<th>Response</th>
<th>Confirm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoke Id</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>SM RP DA</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>SM RP OA</td>
<td>M</td>
<td>M(=)</td>
<td>M(=)</td>
<td>M(=)</td>
</tr>
<tr>
<td>SM RP UI</td>
<td>M</td>
<td>M(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>More Messages To Send</td>
<td>C</td>
<td>C(=)</td>
<td>C</td>
<td>C(=)</td>
</tr>
<tr>
<td>User error</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider error</td>
<td></td>
<td></td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

10.9.3 Parameter use

Invoke id:
See definition in subclause 5.6.1.

SM RP DA:
See definition in subclause 5.6.8. This parameter can contain either an IMSI or a LMSI. The use of the LMSI is an operator option. The LMSI can be provided if it is received from the HLR. The IMSI is used if the use of the LMSI is not available.

This parameter is omitted in the mobile terminated subsequent SM transfers.

SM RP OA:
See definition in subclause 5.6.8. The Service Centre address received from the originating Service Centre is inserted in this parameter.
This parameter is omitted in the mobile terminated subsequent SM transfers.

**SM RP UI:**

See definition in subclause 5.6.8. The short message transfer protocol data unit received from the Service Centre is inserted in this parameter. A short message transfer protocol data unit may also be inserted in this parameter in the message delivery acknowledgment from the MSC to the Service Centre.

**More Messages To Send:**

See definition in subclause 5.6.8. The short message transfer protocol data unit received from the Service Centre is inserted in this parameter.

**User error:**

The following errors defined in subclause 5.6.1 may be used, depending on the nature of the fault:

- Unidentified subscriber;
- Absent Subscriber_SM;
- Subscriber busy for MT SMS;
- Facility Not Supported;
- Illegal Subscriber indicates that delivery of the mobile terminated short message failed because the mobile station failed authentication;
- Illegal equipment indicates that delivery of the mobile terminated short message failed because an IMEI check failed, i.e. the IMEI was blacklisted or not white-listed;
- System Failure;
- SM Delivery Failure;
- The reason of the SM Delivery Failure can be one of the following in the mobile terminated SM:
  - memory capacity exceeded in the mobile equipment;
  - protocol error;
  - mobile equipment does not support the mobile terminated short message service.
- Unexpected Data Value;
- Data Missing.

**Provider error:**

For definition of provider errors see subclause 5.6.1.

---

# 11 General

## 11.1 Overview

Clause 11 to 14 specify the protocol elements to be used to provide the MAP services described in clause 5.

Clause 12 specifies the elements of procedures for the MAP protocol. Clause 13 specifies the mapping on to TC service primitives. Clause 14 specifies the application contexts, operation packages and abstract syntaxes for the MAP protocol as well as the encoding rules to be applied.
11.2 Underlying services

The MAP protocol relies on the services provided by the Transaction Capabilities (TC) of signalling system number 7, as referenced in clause 4.

11.3 Model

The MAP Protocol Machine (MAP PM) can be modelled as a collection of service state machines (SSMs) - one per MAP specific service invoked - coordinated by a MAP dialogue control function with its one state machine: MAP dialogue state machine (DSM). There are two types of Service State Machines: Requesting Service State Machines (RSM) and Performing Service State Machines (PSM).

A new invocation of a MAP PM is employed on the receipt of a MAP-OPEN request primitive or a TC-BEGIN indication primitive. Each invocation controls exactly one MAP dialogue. For each MAP specific service invoked during a dialogue, a MAP RSM is created at the requestor's side and a MAP PSM is created at the performer's side.

This modelling is used only to facilitate understanding and the MAP behaviour descriptions and is not intended to suggest any implementation. SDL descriptions are organized according to this model.

How the MAP-service-user and the MAP refer to a MAP dialogue (i.e. a MAP PM invocation) is a local implementation matter.

How TC dialogue identifiers are assigned to a MAP PM invocation is also a local implementation matter.

11.4 Conventions

The behaviour of the MAP PM depends on the application-context-name associated with the dialogue. One major difference is that the MAP requests the transfer of the application-context-name by TC only for those contexts which do not belong to the so-called "version one context set".

The "version one context set" is a set of application-contexts which model the behaviour of a MAP V1 implementation according to the latest phase 1 version of GSM 09.02. This set is defined in clause 12.

The procedures described in clause 12 are used when the application-context-name does not refer to a dialogue between an MSC and its VLR. When the application-context-name refers to a dialogue between an MSC and its VLR the MAP PM procedures are a local implementation matter.

12 Elements of procedure

12.1 Dialogue establishment

The establishment of a MAP dialogue involves two MAP-service-users, one that is the dialogue-initiator and one that is the dialogue-responder.

This procedure is driven by the following signals:

- a MAP-OPEN request primitive from the dialogue-initiator;
- a TC-BEGIN indication primitive occurring at the responding side;
- a MAP-OPEN response primitive from the dialogue-responder;
- the first TC-CONTINUE indication primitive occurring at the initiating side;
and under specific conditions:

- a TC-END indication primitive occurring at the initiating side;
- a TC-U-ABORT indication primitive occurring at the initiating side;
- a TC-P-ABORT indication primitive occurring at the initiating side.

12.1.1 Handling of unknown operations

Unknown operations (i.e. a standard operation introduced in a later version of 09.02 or a private operation) can be introduced in MAP in a backwards compatible way. This means, that the receiver of an unknown operation shall, if the dialogue state allows it, send a TC-REJECT component to the sender of the operation indicating ‘unrecognised operation’ and continue with the processing of further components or messages exchanged within the dialogue as if the unknown operation had not been received.

The standardised structure of a MAP dialogue shall not be affected by the invocation of unknown operations, i.e. if a dialogue uses only a TC-BEGIN message which is acknowledged by a TC-END message, a TC-CONTINUE message shall not be used to invoke an unknown operation. However the standardised structure of a MAP dialogue may be affected by the rejection of unknown operations, i.e. if a dialogue uses only a TC-BEGIN message which is acknowledged by a TC-END message, a TC-CONTINUE message followed by a TC-END message may be used to carry the rejection of an unknown operation and the response to the standardised operation. The entity which initiated a dialogue whose standardised structure is a TC-BEGIN message which is acknowledged by a TC-END message shall not send any messages in that dialogue after the TC-BEGIN.

Note that if the dialogue structure is affected as described in this paragraph the TC-CONTINUE shall include the dialogue portion required to confirm the acceptance of the dialogue.

Unknown operations can be invoked in the following types of messages (there is no restriction as to how many unknown operations can be invoked in a message):

- TC-BEGIN the component to invoke the unknown operation shall follow the component of the standard operation that is included in this message;
- TC-CONTINUE: the component to invoke the unknown operation may be transported as the only component in a stand-alone message or can be grouped with existing operations. In the latter case a specific sequencing of components is not required;
- TC-END: if the component to invoke the unknown operation is grouped with an existing operation a specific sequencing of components is not required.

The TC-REJECT component may be sent in the following messages:

- TC-CONTINUE or TC-END: either as the only component of the message or grouped with an existing component. The choice is up to the MAP-Service User;
- If the received message contains only unknown operations the MAP-Service User shall send the TC-REJECT components in a TC-CONTINUE message to the peer entity, if the dialogue state allows it;
- If the received message contains unknown operations and standard operations and the standardised structure of the dialogue requires the response to the standard operation to be sent within a TC-END message, then the MAP-Service User may send the response to the standard operations and the TC-REJECT components for the unknown operations in a TC-CONTINUE message followed by a TC-END message. A specific distribution of the components to the TC messages or a specific sequencing of components is not required.

Note that SDLs of chapters 16-21 do not show the report to the MAP-Service User about the reception of the unknown operation. This has been done for the sake of simplicity of description; the MAP PM may inform the MAP-Service User.

The sender of the unknown operation shall ensure that there is enough room in the used message for the unknown operation.
12.1.2 Receipt of a MAP-OPEN request primitive

On receipt of a MAP-OPEN request primitive the behaviour of the MAP PM shall be as follows.

The MAP PM shall accept zero, one or several user request primitives until a MAP-DELIMITER request primitive is received.

For each user request primitive, the MAP PM shall request the invocation of the associated operation using the TC-INVoke service. See subclause 12.6 for a description of the associated SSMs.

On receipt of the MAP-DELIMITER request primitive the MAP PM shall issue a TC-BEGIN request primitive. The application-context-name as well as the user information parameter (if any) shall be mapped to the corresponding TC-BEGIN parameters.

The requesting MAP PM waits for a TC indication primitive and does not accept any other primitive from its user, except a MAP-U-ABORT request or a MAP-CLOSE request.

12.1.3 Receipt of a TC-BEGIN indication

On receipt of a TC-BEGIN indication primitive, the MAP PM shall:

- if no application-context-name is included in the primitive and if the "Components present" indicator indicates "no components", issue a TC-U-ABORT request primitive (note 2). The local MAP-User is not informed;

- if no application-context-name is included in the primitive and if presence of components is indicated, wait for the first TC-INVOKE primitive, and derive a version 1 application-context-name from the operation code according to table 12.1/1 (note 1);

NOTE 1: In some cases, it may be necessary to analyse the operation argument.

Then:

a) if no application-context-name can be derived (i.e. the operation code does not exist in MAP V1 specifications), the MAP PM shall issue a TC-U-ABORT request primitive (note 2). The local MAP-User is not informed.

b) if an application-context-name can be derived and if it is acceptable from a load control point of view, the MAP PM shall:

i) if this primitive requests the beginSubscriberActivity operation, the MAP PM shall check whether more components have been received associated with this operation. If more components are present, the MAP PM shall issue a MAP-OPEN indication primitive with the version 1 application-context-name "networkFunctionalSsContext-v1". The Destination-reference shall include the IMSI taken from the argument of the beginSubscriberActivity operation; the Originating-reference shall cover the originatingEntityNumber.

A beginSubscriberActivity operation that is not associated with any other Component shall be rejected by the MAP PM by issuing a TC-U-ABORT request primitive (note 2). The local MAP-User shall not be informed.

ii) otherwise, the MAP PM shall issue a MAP-OPEN indication primitive with the version 1 application-context-name set according to table 12.1/1. DestinationReference and OriginatingReference must not be included in the MAP-OPEN indication primitive.

Then the MAP PM shall function in a way that the dialogue responding MAP behaves as specified in the GSM phase 1 protocol (latest version of TS GSM 09.02 phase 1).

NOTE 2: If no AARQ apdu was included in the BEGIN message, TC (Component Sub-layer) will not include an AARE apdu or an ABRT apdu in a TR-U-ABORT request primitive that is to be issued on receipt of a TC-U-ABORT request primitive from the local MAP service provider.

c) if an application-context-name can be derived but if it is not acceptable from a load control point of view, the MAP PM shall ignore this dialogue request and not inform the MAP-user;
- if a version 1 application-context-name is included, the MAP PM shall issue a TC-U-ABORT request primitive with abort-reason "User-specific" and user-information "MAP-ProviderAbortInfo" indicating "abnormalDialogue". The local MAP-user shall not be informed.

- if an application-context-name different from version 1 is included in the primitive and if User-information is present, the User-information must constitute a syntactically correct MAP-OPEN dialogue PDU. Otherwise a TC-U-ABORT request primitive with abort-reason "User-specific" and user-information "MAP-ProviderAbortInfo" indicating "abnormalDialogue" shall be issued and the local MAP-user shall not be informed.

- if no User-information is present it is checked whether presence of User Information in the TC-BEGIN indication primitive is required for the received application-context-name. If User Information is required but not present, a TC-U-ABORT request primitive with abort-reason "User-specific" and user-information "MAP-ProviderAbortInfo" indicating "abnormalDialogue" shall be issued. The local MAP-user shall not be informed.

- if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN indication primitive but is not acceptable from a load control point of view, the MAP PM shall ignore this dialogue request. The MAP-user is not informed.

- if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN indication primitive and if it is acceptable from a load control point of view, the MAP PM shall check whether the application-context-name is supported.

NOTE 3: Unknown application-context-names are treated like unsupported ones.

If it is, the MAP PM shall issue a MAP-OPEN indication primitive with all parameters (application-context-name included) set according to the value of the corresponding parameter of the TC-BEGIN indication primitive.

The MAP PM shall then process any other indication primitives received from TC as described in subclause 12.6. Once all the received components have been processed, the MAP PM shall inform the local MAP service user by a MAP-DELIMITER indication primitive.

If the TC-BEGIN indication primitive is not associated with any component, the MAP PM shall inform the MAP User by a MAP-DELIMITER indication primitive.

Once all the received primitives have been processed, the MAP PM does not accept any primitive from the provider and waits for a MAP-OPEN response primitive from its user.

- if an application-context-name different from version 1 is received in a syntactically correct TC-BEGIN indication primitive and if it is acceptable from a load control point of view but the application-context-name is not supported, the MAP PM shall issue a TC-U-ABORT request primitive with abort-reason indicating "application-context-not-supported". If an alternative application-context-name cannot be offered, the received application-context-name shall be returned in the TC-U-ABORT Req primitive.

In the following cases an alternative application-context can be offered and its name included in the TC-U-ABORT Req primitive:

a) if an application-context of version 2 or higher is requested, but only version 1 application-context supported, then the v1 application context shall be returned;

b) if an application-context of version 3 or higher is requested, but only version 2 application-context supported, then the v2 application context shall be returned.

c) if an application-context of version 4 or higher is requested, but only version 3 application-context supported, then the v3 application context shall be returned.
Table 12.1/1: Mapping of V1 operation codes on to application-context-names

<table>
<thead>
<tr>
<th>Operation</th>
<th>Application-context-name (note 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>updateLocation</td>
<td>networkLocUpContext-v1</td>
</tr>
<tr>
<td>cancelLocation</td>
<td>locationCancellationContext-v1</td>
</tr>
<tr>
<td>provideRoamingNumber</td>
<td>roamingNumberEnquiryContext-v1</td>
</tr>
<tr>
<td>insertSubscriberData</td>
<td>subscriberDataMngtContext-v1</td>
</tr>
<tr>
<td>deleteSubscriberData</td>
<td>subscriberDataMngtContext-v1</td>
</tr>
<tr>
<td>sendParameters</td>
<td>infoRetrievalContext-v1</td>
</tr>
<tr>
<td>beginSubscriberActivity</td>
<td>networkFunctionalSSContext-v1</td>
</tr>
<tr>
<td>sendRoutingInfo</td>
<td>locationInfoRetrievalContext-v1</td>
</tr>
<tr>
<td>performHandover</td>
<td>handoverControlContext-v1</td>
</tr>
<tr>
<td>reset</td>
<td>resetContext-v1</td>
</tr>
<tr>
<td>activateTraceMode</td>
<td>tracingContext-v1</td>
</tr>
<tr>
<td>deactivateTraceMode</td>
<td>tracingContext-v1</td>
</tr>
<tr>
<td>sendRoutingInfoForSM</td>
<td>shortMsgGatewayContext-v1</td>
</tr>
<tr>
<td>forwardSM</td>
<td>shortMsgRelayContext-v1</td>
</tr>
<tr>
<td>reportSM-deliveryStatus</td>
<td>shortMsgGatewayContext-v1</td>
</tr>
<tr>
<td>noteSubscriberPresent</td>
<td>mwdMngtContext-v1</td>
</tr>
<tr>
<td>alertServiceCentreWithoutResult</td>
<td>shortMsgAlertContext-v1</td>
</tr>
<tr>
<td>checkIMEI</td>
<td>EquipmentMngtContext-v1</td>
</tr>
</tbody>
</table>

NOTE 1: These symbolic names refer to the object identifier value defined in clause 14 and allocated to each application-context used for the MAP.

NOTE 2: The choice between the application contexts is based on the parameters received in the operation.

12.1.4 Receipt of a MAP-OPEN response

On receipt of a MAP-OPEN response primitive indicating that the dialogue is accepted, the MAP PM shall build a MAP-Accept PDU if the user-information parameter is included in the response primitive and accept any MAP specific service request or service response until a MAP-DELIMITER request or a MAP-CLOSE request is received from the MAP user. The MAP PM shall process the MAP specific primitives as described in subclause 12.6. The MAP PM shall then issue a TC-CONTINUE request primitive after it receives the MAP-DELIMITER request primitive if no MAP-CLOSE request primitive has been received, otherwise it shall issue a TC-END request primitive. In both cases the MAP-Accept PDU (if any) is included in the user-information parameter of the TC primitive.

If the dialogue is not associated with a version 1 application context, the MAP PM shall include the application-context-name in the TC primitive.

If no MAP-CLOSE request has been received, the MAP PM waits for a request primitive from its user or an indication primitive from TC.

On receipt of a MAP-OPEN response primitive indicating that the dialogue is not accepted, the MAP PM shall build a MAP-Refuse PDU and request its transfer using the TC-U-ABORT req primitive (abort reason = user specific).

12.1.5 Receipt of the first TC-CONTINUE ind

On receipt of the first TC-CONTINUE indication primitive for a dialogue, the MAP PM shall check the value of the application-context-name parameter. If this value matches the one used in the MAP-OPEN request primitive, the MAP PM shall issue a MAP-OPEN confirm primitive with the result parameter indicating "accepted", then process the following TC component handling indication primitives as described in subclause 12.6, and then waits for a request primitive from its user or an indication primitive from TC, otherwise it shall issue a TC-U-ABORT request primitive with a MAP-providerAbort PDU indicating "abnormal dialogue" and a MAP-P-ABORT indication primitive with the "provider-reason" parameter indicating "abnormal dialogue".
12.1.6 Receipt of a TC-END ind

On receipt of a TC-END indication primitive in the dialogue initiated state, the MAP PM shall check the value of the application-context-name parameter. If this value does not match the one used in the MAP-OPEN request primitive, the MAP PM shall discard any following component handling primitive and shall issue a MAP-P-ABORT indication primitive with the "provider-reason" parameter indicating "abnormal dialogue".

Otherwise it shall issue a MAP-OPEN confirm primitive with the result parameter set to "accepted" and process the following TC component handling indication primitives as described in subclause 12.6; then it shall issue a MAP-CLOSE indication primitive and return to idle all state machines associated with the dialogue.

12.1.7 Receipt of a TC-U-ABORT ind

On receipt of a TC-U-ABORT indication primitive in the "Dialogue Initiated" state with an abort-reason parameter indicating "ApplicationContextNotSupported", the MAP PM shall issue a MAP-OPEN confirm primitive with the result parameter indicating "Dialogue Refused" and the refuse-reason parameter indicating "ApplicationContextNotSupported".

On receipt of a TC-U-ABORT indication primitive in the "Dialogue Initiated" state with an abort-reason parameter indicating "User Specific" and without user information, the MAP PM shall issue a MAP-OPEN confirm primitive with the result parameter indicating "Dialogue Refused" and the refuse-reason parameter indicating "Potential Version Incompatibility".

On receipt of a TC-U-ABORT indication primitive in the "Dialogue Initiated" state with an abort-reason parameter indicating "User Specific" and a MAP-Refuse PDU included as user information, the MAP PM shall issue a MAP-OPEN confirm primitive with the result set to refused and the refuse reason set as received in the MAP Refuse PDU.

Receipt of a TC-U-ABORT indication primitive with abort-reason "User Specific" and with user information is described as part of abnormal termination (see subclause 12.4.2).

12.1.8 Receipt of a TC-P-ABORT ind

On receipt of a TC-P-ABORT indication primitive in the "Dialogue Initiated" state with a P-abort parameter indicating "Incorrect Transaction Portion", the MAP PM shall issue a MAP-OPEN confirm primitive with the result parameter indicating "Dialogue Refused" and the refuse reason parameter indicating "Potential Version Incompatibility".

On receipt of a TC-P-ABORT indication primitive in the "Dialogue Initiated" state with a P-abort parameter indicating "No Common Dialogue Portion", the MAP PM shall issue a MAP-P-ABORT indication primitive with the provider reason parameter indicating "Version Incompatibility".

Receipt of a TC-P-ABORT indication primitive with another P-abort parameter value is described as part of abnormal termination (see subclause 12.5.2).

12.2 Dialogue continuation

Once established the dialogue is said to be in a continuation phase.

Both MAP users can request the transfer of MAP APDUs until one of them requests the termination of the dialogue.

12.2.1 Sending entity

The MAP PM shall accept any MAP specific service request or response primitives and process them as described in subclause 12.6.

On receipt of a MAP-DELIMITER request primitive, the MAP PM shall issue a TC-CONTINUE request primitive.

12.2.2 Receiving entity

On receipt of a TC-CONTINUE indication primitive the MAP PM shall accept zero, one or several TC component handling indication primitives and process them as described in subclause 12.6.
12.3 Dialogue termination

Both the dialogue-initiator and the dialogue-responder have the ability to request the termination of a dialogue after it has been established.

The dialogue termination procedure is driven by the following events:

- a MAP-CLOSE request primitive;
- a TC-END indication primitive.

12.3.1 Receipt of a MAP-CLOSE request

On receipt of a MAP-CLOSE request primitive, the MAP PM shall issue a TC-END request primitive and, if applicable, return to idle the associated active SSMs. Note that if the release method parameter of the MAP-CLOSE request indicates "normal" the TC-END request primitive will trigger the transmission of components associated with any user specific request or response primitives which may have been issued after the last MAP-DELIMITER request.

12.3.2 Receipt of a TC-END indication

On receipt of a TC-END indication primitive, the MAP shall accept any component handling indication primitives and process them as described in subclause 12.6.

Once all the received primitives have been processed, the MAP PM shall return to idle the associated SSMs and issue a MAP-CLOSE indication primitive.

12.4 User Abort

Both the dialogue-initiator and the dialogue-responder have the ability to abort a dialogue at any time.

The user abort procedure is driven by one of the following events:

- a MAP-U-ABORT request primitive;
- a TC-U-ABORT indication primitive carrying a MAP-user-abort PDU.

12.4.1 MAP-U-ABORT request

On receipt of a MAP-U-ABORT request the MAP PM shall construct a MAP-user-abort PDU from the user-reason and diagnostic parameters and issue a TC-U-ABORT request primitive. All state machines associated with the dialogue are returned to idle.

12.4.2 TC-U-ABORT ind

On receipt of a TC-U-ABORT indication carrying a MAP-user-abort PDU, the MAP PM shall issue a MAP-U-ABORT indication primitive. The user-reason and diagnostic information elements are mapped to the corresponding parameters of the MAP-U-ABORT indication primitive.

All state machines associated with the dialogue are returned to idle.

12.5 Provider Abort

The MAP has the ability to abort a dialogue at both the dialogue-initiator side and the dialogue-responder side.

The provider abort procedure is driven by one of the following events:

- a MAP PM error situation;
- a TC-P-ABORT indication primitive;
- a TC-U-ABORT indication primitive carrying a MAP-abort PDU.

12.5.1 MAP PM error situation

In the case of an abnormal situation detected at the MAP level during an established dialogue, the MAP PM shall:

- issue a MAP-P-ABORT indication primitive with the appropriate value of the provider-reason parameter;
- construct a MAP-abort PDU from the value of these parameters and request its transfer using a TC-U-ABORT request primitive.

12.5.2 TC-P-ABORT ind

On receipt of a TC-P-ABORT indication, the MAP PM shall issue a MAP-P-ABORT indication primitive.

All state machines associated with the dialogue are returned to idle.

12.5.3 TC-U-ABORT ind

On receipt of a TC-U-ABORT indication carrying a MAP-abort PDU, the MAP PM shall issue a MAP-P-ABORT indication primitive, with the appropriate value of the provider-reason parameter. The source parameter shall indicate "MAP-provider".

All state machines associated with the dialogue are returned to idle.

12.6 Procedures for MAP specific services

This subclause describes the MAP procedures for MAP specific services.

These procedures are driven by the following types of events:

- a MAP specific request or a MAP specific MAP response primitive;
- a component handling primitive from TC.

A Service State Machine is activated on receipt of one of the following signals:

- a MAP request primitive, which activates a requesting SSM;
- a TC-INVOKE indication primitive without linked identifier, which activates a responding SSM.

For component handling primitives there are two types of events:

- events which activate a Service State Machine or which can be related to an existing one;
  The procedure elements driven by these events are described in subclauses 12.6.1 to 12.6.4.
- events which cannot be related to a Service State Machine.
  The procedure elements driven by these events are described in subclause 12.6.5.

12.6.1 Service invocation

The MAP specific procedures are initiated by the MAP request primitives.

On receipt of a MAP request primitive, the MAP PM shall build an operation argument from the parameters received in the request primitive and request the invocation of the associated operation using the TC-INVOKE procedure. If a linked ID parameter is inserted in the primitive this indicates a child service and implies that the operation on which the service is mapped is linked to the operation on which the parent service is mapped.

The mapping of MAP specific services on to remote operations is given in table 13.2/1.
12.6.2 Service invocation receipt

On receipt of a TC-INVOKE indication primitive, the MAP PM shall:

- if the invoke ID is already in use by an active service, request the transfer of a reject component using the TC-U-REJECT request primitive with the appropriate problem code (duplicated invokeID) and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event received from the peer";

- if the operation code does not correspond to an operation supported by the application-context, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognized operation), and if the dialogue version is lower than 3- issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event received from the peer";

- if a linked ID is included, perform the following checks: If the operation referred to by the linked ID does not allow linked operations or if the operation code does not correspond to a permitted linked operation, issue a TC-U-REJECT request primitive with the appropriate problem code (linked response unexpected or unexpected linked operation);

- if the type of the argument is not the one defined for the operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event from the peer";

- if the type of the argument is correct but the values of the information elements it contains do not permit the type of MAP service being invoked to be determined, request the transfer of an error component using the TC-U-ERROR request primitive with an error code set to "unexpected data value" and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event from the peer";

NOTE 1: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

- if the type of the argument is correct but information elements required for the service being invoked are missing, request the transfer of an error component using the TC-U-ERROR request primitive with an error code set to "data missing" and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event from the peer";

NOTE 2: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

- if the type of the argument is correct but contains information elements which are not relevant for the type of MAP service being invoked, request the transfer of an error component using the TC-U-ERROR request primitive with an error code set to "unexpected data value" and issue a MAP-NOTICE indication primitive with a diagnostic parameter set to "abnormal event from the peer";

NOTE 3: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

- Otherwise, issue the relevant MAP indication primitive to the MAP-service-user. If the service is to be user confirmed, the MAP PM waits for the corresponding response primitive.

12.6.3 Service response

For user confirmed services, the MAP PM shall accept a MAP response primitive and shall:

- if no error indication is included in the primitive and the service maps on to a class 1 or 3 operation, construct a result information element from the parameters received and request its transfer using the TC-RESULT-L service and optionally the TC-RESULT-NL service.

The TC-RESULT-NL services shall be used when the user specific parameters of the response primitives cannot be transferred in a single signalling frame and no segmenting mechanism is available from the underlying layers. The MAP PM shall issue one or several TC-RESULT-NL request primitives followed by a TC-RESULT-L primitive. The user parameters shall be split so that each portion contains sufficient information to construct a value compatible with the type defined for the result of the associated operation.
- if no error indication is included in the primitive and the service response maps on to a class 4 linked operation, construct an operation argument from the parameters received and request its transfer using the TC-INVOKE service for this class 4 linked operation. The operation to be invoked is deduced from the value of the result parameter of the service primitive;

- if an error indication is included in the primitive and the service maps on to a class 1 or 2 operation, either issue a TC-U-REJECT request primitive if the user error parameter indicates "resource limitation" or "initiating release", or construct an error parameter from the parameters received and request its transfer using the TC-U-ERROR request primitive. The error code should be the one associated with the value of the user error parameter of the response primitive.

NOTE: The only user errors that a MAP user can generate in addition to the list of errors attached to the operation which is associated with the service are: resource limitation and initiating release. Any other abnormal situation is detected either by the TC entity or by the MAP entity.

- if an error indication is received and the operation maps on to a class 3 operation, or if no error indication is received but the service maps on to a class 2 operation which has no class 4 linked operation, return the local service state machine to idle without requesting any service from TC.

12.6.4 Receipt of a response

A component handling indication primitive is considered as driving a response for a confirmed service if the invoke ID parameter value matches the one stored for the service, or if the linked ID parameter value matches the one stored for the service and the operation invoked is a class 4 operation. On receipt of a response (except a TC-L-CANCEL indication) for an unconfirmed service the MAP PM shall issue a MAP-NOTICE indication primitive with the appropriate provider error (return result unexpected or return error unexpected).

12.6.4.1 Receipt of a TC-RESULT-NL indication

If the type of the partial result parameter is not compatible with the one defined for the complete result of this operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter) and issue a confirm primitive with the provider error parameter set to "invalid response received". The MAP PM shall also issue a TC-U-CANCEL request primitive so that all subsequent result components for this operation are discarded by TC.

Otherwise, store the value of the partial result parameter and wait for subsequent TC-RESULT-NL indication primitives until a TC-RESULT-L indication primitive is received.

12.6.4.2 Receipt of a TC-RESULT-L indication

If the type of the result parameter is not the one defined for the result of this operation, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and issue a confirm primitive with the provider error parameter set to "invalid response received".

If the type of the result parameter is correct but does not contain all the information elements required by the service associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response received".

NOTE 1: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

If the type of the result parameter is correct but contains information elements which are not relevant for the service associated with the invocation are missing, issue a confirm primitive with the provider error parameter set to "invalid response received".

NOTE 2: These checks are only relevant when there is not a one-to-one mapping between a service and an operation.

Otherwise, issue a MAP confirm primitive to the MAP-service-user mapping the result parameter of the TC-RESULT-L primitive on to the MAP specific parameters.

If partial results have been previously received, the value of the partial result parameters shall also be taken into account before performing the three previous checks.
12.6.4.3 Receipt of a TC-U-ERROR indication

If the error code is not defined for the MAP or is not one associated with the operation referred to by the invoke identifier, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (unrecognized error or unexpected error), and issue a confirm primitive with the provider error parameter set to "invalid response received".

If the type of the error parameter is not the one defined for this error, request the transfer of a reject component using the TC-U-REJECT request primitive, with the appropriate problem code (mistyped parameter), and issue a confirm primitive with the provider error parameter set to "invalid response received".

If the type of the error parameter is correct but does not contain all the information elements required by the service associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response received".

NOTE 1: In some cases, it may be necessary to analyse the operation argument.

If the type of the error parameter is correct but its value includes information elements which are not relevant for the service associated with the invocation, issue a confirm primitive with the provider error parameter set to "invalid response received".

NOTE 2: In some cases, it may be necessary to analyse the operation argument.

Otherwise, issue a MAP confirm primitive to the MAP-service-user with the user error parameter set according to the received error code. If applicable the error parameter is mapped to the diagnostic parameter.

12.6.4.4 Receipt of a TC-INVOKE indication

A TC-INVOKE indication primitive is considered as carrying a possible response to a specific service if the linked ID refers to an active specific service and the associated operation is a class 4 operation. Note that the presence of a linked ID parameter in a TC-INVOKE primitive requesting a non class 4 operation indicates a child service whose procedures are the same as the procedures for the parent service.

On receipt of a TC-INVOKE indication confirming an active service, the MAP PM shall:

- if the operation code is not defined for MAP and the dialogue version is at least 3, issue a TC-U-REJECT request primitive with the appropriate problem code (unrecognized operation).

- if the operation code is not defined for MAP and the dialogue version is lower than 3, or if the operation referred to by the linked ID does not allow linked operations or if the operation code does not correspond to an allowed linked operation, issue a TC-U-REJECT request primitive with the appropriate problem code (unrecognized operation, linked response unexpected or unexpected linked operation). If the service is confirmed, the MAP shall also issue a Confirm primitive with provider error indication "unexpected response from the peer", otherwise it may issue a MAP-NOTICE indication primitive with an appropriate diagnostic "abnormal event received from the peer".

- otherwise issue a confirm primitive mapping the operation argument parameter to the user specific parameters and setting the result parameter according to the operation code of the linked operation.

12.6.4.5 Receipt of a TC-U-REJECT indication

On receipt of a TC-U-REJECT indication primitive which affects a pending service, the MAP PM shall issue a MAP confirm primitive to the MAP-service-user with the appropriate value of the provider error or user error parameter.

The mapping of TC invoke problem codes on to MAP Provider Error and MAP User Error parameter values is described in clause 13.

12.6.4.6 Receipt of a TC-L-REJECT indication

This event occurs when the local TC detects a protocol error in an incoming component which affects an active specific service.
On receipt of a TC-L-REJECT indicating "return result problem, unexpected return result", the MAP shall issue a confirm primitive with the parameter provider error indicating "unexpected response from the peer".

On receipt of a TC-L-REJECT indicating "return error problem, unexpected error result", the MAP shall issue a confirm primitive with the parameter provider error indicating "unexpected response from the peer".

Note that when the problem code indicates a general problem, it is considered that the event cannot be related to an existing SSM even if the invoke Id is provided by TC. This is because whether the invoke Id refers to a local or remote invocation is ambiguous. The behaviour of the MAP PM in such a case is described in subclause 12.6.5.3.

12.6.4.7 Receipt of a TC-L-CANCEL indication

On receipt of a TC-L-CANCEL indication, the MAP PM shall:
- if the associated operation is a class 1 operation, issue a confirm primitive with the provider error cause indicating "no response from the peer";
- if the associated operation is a class 2 operation and no linked operations are defined for this operation, issue a confirm primitive without parameter (i.e. indicating implicitly the successful completion of the service);
- if the associated operation is a class 2 operation and has linked operations but none of them has been invoked, issue a confirm primitive with the provider error parameter indicating "service completion failure";
- if the associated operation is a class 2 operation and a linked operation invocation has already been received in response to this operation, ignore the primitive;
- if the associated operation is a class 3 operation, issue a confirm primitive with the provider error cause indicating "service completion failure";
- if the associated operation is a class 4 operation, ignore the primitive.

NOTE: When a TC-L-CANCEL ind primitive is received before the dialogue has been confirmed (i.e. no backward message is received by the dialogue initiator node), the MAP PM shall first issue a MAP-OPEN Cnf primitive with the result parameter indicating "accepted" (which means that the dialogue is considered as being implicitly accepted). Then, as indicated above, the TC-L-CANCEL Indication is interpreted according to the class of the operation to which it refers.

12.6.4.8 Receipt of a TC-NOTICE indication

If a TC-NOTICE indication primitive is received before the dialogue has been confirmed (i.e. no backward message is received by the dialogue initiator node), the MAP PM shall issue a MAP-OPEN Cnf primitive with the result parameter indicating Refused and a refuse reason Remote node not reachable".

If a TC-NOTICE indication primitive is received after the dialogue has been confirmed, the MAP PM shall issue a MAP-NOTICE indication to the user, with a problem diagnostic indicating "message cannot be delivered to the peer".

12.6.5 Other events

This subclause describes the behaviour of the MAP PM on receipt of a component handling indication primitive which cannot be related to any service or which does not affect a pending one. The MAP user is only informed that an abnormal event occurred during the associated dialogue. It is up to the MAP user to abort, continue or terminate the dialogue.

12.6.5.1 Receipt of a TC-U-REJECT

On receipt of a TC-U-REJECT indication primitive which does not affect an active SSM (i.e. indicating a return result or return error problem), the MAP PM shall issue a MAP-NOTICE indication primitive with the diagnostic parameter set to "response rejected by the peer".

This is also applicable for invoke problems related to a class 4 linked operation.
12.6.5.2 Receipt of a TC-R-REJECT indication

On receipt of a TC-R-REJECT indication (i.e. when a protocol error has been detected by the peer TC entity) which does not affect an active SSM, the MAP PM shall either discard this indication or issue a MAP-NOTICE indication primitive with the provider error indicating "abnormal event detected by the peer".

In case of notification, it is up to the MAP user to continue, abort or terminate the dialogue. Note also that for MAP V1 the reject component is received in an END message and therefore the dialogue is terminated anyway.

12.6.5.3 Receipt of a TC-L-REJECT indication

On receipt of a TC-L-REJECT indication primitive (i.e. when a protocol error has been detected by the local TC entity) which cannot be related to an active SSM, the MAP PM shall either discard this indication or issue a MAP-NOTICE indication primitive with the provider error indicating "abnormal event received from the peer".

In case of notification, it is up to the MAP user to continue, or to terminate the dialogue and implicitly trigger the transmission of the reject component or to abort the dialogue.

12.6.6 Parameter checks

As described in the previous subclauses, the MAP PM performs a set of checks to ensure the correctness of the information elements received; these are:

- check if the syntax and encoding (note) of the operation argument, result or error parameter are correct.

  NOTE: Depending on the implementation, encoding problems on the TC user portion may be detected at TC level or by the MAP user. In the second case the problem is reported in a similar manner to a syntactical problem.

  The syntax shall be considered incorrect if a mandatory information element is missing in any constructed element or if the value of an information element is out of the range defined for the type it is supposed to belong to;

- if there is not a one-to-one mapping between a service and an operation:

  i) check if the value of the information elements (generally a single one) permits the MAP PM to determine the service associated with the operation invocation;

  ii) check that there are no information elements which are irrelevant for the indication or a confirm primitive to be issued

- check if all the information elements required to built an indication or a confirm primitive are available.

However some additional checks may have to be performed by the MAP user (see clause 15).

12.6.7 Returning state machines to idle

Unlike TC invocation state machines, service state machines exist at both requestor and performer side.

A service state machine at the requestor side is returned to idle when the MAP-specific confirm primitive is issued or when the dialogue terminates.

A service state machine at the performer side is returned to idle on receipt of a MAP-specific response primitive from the MAP user, when the dialogue terminates or at expiry of an implementation dependent watch-dog timer which is started when the state machine is created.
12.6.8 Load control

As stated in the previous subclauses, before issuing a MAP-OPEN indication primitive the MAP PM performs a check to verify if there are sufficient resources to open the dialogue taking into account possible overload conditions.

The decision is based on the priority allocated to the application-context whose name is explicitly included in the TC-BEGIN indication primitive or implied by the first operation invocation when V1 contexts are in use. How a V1 application-context-name is derived from an operation code is described in table 12.1/1.

The priority level allocated to each application-context is described in clause 3 tables 3.1/1 and 3.1/2.

13 Mapping on to TC services

13.1 Dialogue control

Dialogue control services are mapped to TC dialogue handling services. The TC-UNI service is not used by the MAP PM.

13.1.1 Directly mapped parameters

The following parameters of the MAP-OPEN request and indication primitives are directly mapped on to the corresponding parameters of the TC-BEGIN primitives:

- destination address;
- originating address.

13.1.2 Use of other parameters of dialogue handling primitives

13.1.2.1 Dialogue Id

The value of this parameter is associated with the MAP PM invocation in an implementation dependent manner.

13.1.2.2 Application-context-name

The application-context-name parameter of a MAP primitive is mapped to the application-context-name parameter of TC dialogue handling primitives according to the rules described in subclause 12.1.

13.1.2.3 User information

The user information parameter of TC dialogue primitives is used to carry the MAP dialogue APDUs.

13.1.2.4 Component present

This parameter is used by the MAP PM as described in CCITT Recommendation Q.771. It is not visible to the MAP user.

13.1.2.5 Termination

The value of this parameter of the TC-END request primitive is set by the MAP PM on the basis of the release method parameter of the MAP-CLOSE request primitive, except when the dialogue state machine is in the state DIALOGUE INITIATED, in which case the Termination parameter shall always indicate "pre-arranged end".
13.1.2.6 P-Abort-Cause

Values of the P-abort-cause parameter are mapped to the values of the provider-reason parameter of the MAP-P-ABORT indication primitive according to table 13.1/1, except in the dialogue initiated phase for the "incorrectTransactionPortion" and "noCommonDialoguePortion" values which are mapped to the "potential incompatibility problem" value of the refuse-reason parameter of the MAP-OPEN cnf primitive. The source parameter in the MAP-P-ABORT ind takes the value "TC problem".

13.1.2.7 Quality of service

The quality of service of TC request primitives is set by the MAP as shown below.

- Return option: "Return message on error" or "Discard message on error" as required by the network operator;
- Sequence control: "Sequence guaranteed" or "Sequence result not guaranteed" as required by the network operator;

"Sequence guaranteed" shall be used when a segmented result is to be transferred (e.g. subscriber data in response to SendParameters). It may also be appropriate to use Sequence guaranteed when a series of InsertSubscriberData, ProcessAccessSignalling or ForwardAccessSignalling operations is used.

It is essential that the TC message which indicates acceptance of a dialogue opening request is received by the dialogue initiator before any subsequent message in that dialogue; otherwise the dialogue opening will fail. The dialogue responder shall ensure that this requirement is met by:

- Sending the dialogue acceptance message in a TC-END, if the dialogue structure requires it; or
- Using "Sequence guaranteed", if the dialogue acceptance message is sent in a TC-CONTINUE; or
- Waiting until the dialogue acceptance message has been acknowledged by the dialogue initiator before sending a subsequent message, if the dialogue acceptance message is sent in a TC-CONTINUE.

<table>
<thead>
<tr>
<th>TC P-Abort cause</th>
<th>MAP provider-reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>unrecognized message type</td>
<td>provider malfunction</td>
</tr>
<tr>
<td>unrecognized transaction Id</td>
<td>supporting dialogue released</td>
</tr>
<tr>
<td>badlyFormattedTransactionPortion</td>
<td>provider malfunction</td>
</tr>
<tr>
<td>incorrectTransactionPortion</td>
<td>provider malfunction (note)</td>
</tr>
<tr>
<td>resourceLimitation</td>
<td>resource limitation</td>
</tr>
<tr>
<td>abnormalDialogue</td>
<td>provider malfunction</td>
</tr>
<tr>
<td>noCommonDialoguePortion</td>
<td>version incompatibility</td>
</tr>
</tbody>
</table>

NOTE: Or version incompatibility in the dialogue initiated phase.

13.2 Service specific procedures

Specific services are mapped to TC component handling services.

13.2.1 Directly mapped parameters

The Invoke Id parameter of the MAP request and indication primitive is directly mapped on to the Invoke Id parameter of the component handling primitives.

13.2.2 Use of other parameters of component handling primitives

13.2.2.1 Dialogue Id

The value of this parameter is associated with the MAP PM invocation in an implementation dependent manner.
13.2.2.2 Class

The value of this parameter is set by the MAP PM according to the type of the operation to be invoked.

13.2.2.3 Linked Id

When a service response is mapped to a class 4 operation, the value of this parameter is set by the MAP PM and corresponds to the value assigned by the user to the initial service request (i.e. the value of the invoke ID parameter of the request primitive). Otherwise if such a parameter is included in MAP request/indication primitives it is directly mapped to the linked ID parameter of the associated TC-INVOKE request/indication primitives.

13.2.2.4 Operation

When mapping a request primitive on to a Remote Operations PDU (invoke), the MAP PM shall set the operation code according to the mapping described in table 13.2/1.

When mapping a response primitive on to a Remote Operations service, the MAP PM shall set the operation code of the TC-RESULT-L/NL primitive (if required) to the same value as the one received at invocation time.

### Table 13.2/1: Mapping of MAP specific services on to MAP operations

<table>
<thead>
<tr>
<th>MAP-SERVICE</th>
<th>operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP-ACTIVATE-SS</td>
<td>activateSS</td>
</tr>
<tr>
<td>MAP-ACTIVATE-TRACE-MODE</td>
<td>activateTraceMode</td>
</tr>
<tr>
<td>MAP-ALERT-SERVICE-CENTRE</td>
<td>alertServiceCentre</td>
</tr>
<tr>
<td>MAP-ANY-TIME-INTERROGATION</td>
<td>anyTimeInterrogation</td>
</tr>
<tr>
<td>MAP-CANCEL-LOCATION</td>
<td>cancelLocation</td>
</tr>
<tr>
<td>MAP-CHECK-IMEI</td>
<td>checkIMEI</td>
</tr>
<tr>
<td>MAP-DEACTIVATE-SS</td>
<td>deactivateSS</td>
</tr>
<tr>
<td>MAP-DEACTIVATE-TRACE-MODE</td>
<td>deactivateTraceMode</td>
</tr>
<tr>
<td>MAP-DELETE-SUBSCRIBER-DATA</td>
<td>deleteSubscriberData</td>
</tr>
<tr>
<td>MAP-ERASE-SS</td>
<td>eraseSS</td>
</tr>
<tr>
<td>MAP-FORWARD-ACCESS-SIGNALLING</td>
<td>forwardAccessSignalling</td>
</tr>
<tr>
<td>MAP-FORWARD-CHECK-SS-INDICATION</td>
<td>forwardCheckSsIndication</td>
</tr>
<tr>
<td>MAP-MT-FORWARD-SHORT-MESSAGE</td>
<td>mt-forwardSM</td>
</tr>
<tr>
<td>MAP-MO-FORWARD-SHORT-MESSAGE</td>
<td>mo-forwardSM</td>
</tr>
<tr>
<td>MAP-GET-PASSWORD</td>
<td>getPassword</td>
</tr>
<tr>
<td>MAP-INFORM-SERVICE-CENTRE</td>
<td>informServiceCentre</td>
</tr>
<tr>
<td>MAP-INSERT-SUBSCRIBER-DATA</td>
<td>insertSubscriberData</td>
</tr>
<tr>
<td>MAP-INTERROGATE-SS</td>
<td>interrogateSs</td>
</tr>
<tr>
<td>MAP-PREPARE-HANOVER</td>
<td>prepareHandover</td>
</tr>
<tr>
<td>MAP-PREPARE-SUBSEQUENT-HANOVER</td>
<td>prepareSubsequentHandover</td>
</tr>
<tr>
<td>MAP-PROCESS-ACCESS-SIGNALLING</td>
<td>processAccessSignalling</td>
</tr>
<tr>
<td>MAP-PROCESS-UNSTRUCTURED-SS-REQUEST</td>
<td>processUnstructuredSS-Request</td>
</tr>
<tr>
<td>MAP-PROVIDE-ROAMING-NUMBER</td>
<td>provideRoamingNumber</td>
</tr>
<tr>
<td>MAP-PROVIDE-SUBSCRIBER-INFO</td>
<td>provideSubscriberInfo</td>
</tr>
<tr>
<td>MAP-PURGE-MS</td>
<td>purgeMS</td>
</tr>
<tr>
<td>MAP-READY-FOR-SM</td>
<td>readyForSM</td>
</tr>
<tr>
<td>MAP-REGISTER-PASSWORD</td>
<td>registerPassword</td>
</tr>
<tr>
<td>MAP-REGISTER-SS</td>
<td>registerSS</td>
</tr>
<tr>
<td>MAP-REPORT-SM-DELIVERY-STATUS</td>
<td>reportSmDeliveryStatus</td>
</tr>
<tr>
<td>MAP-RESET</td>
<td>reset</td>
</tr>
<tr>
<td>MAP-RESTORE-DATA</td>
<td>restoreData</td>
</tr>
<tr>
<td>MAP-SEND-END-SIGNAL</td>
<td>sendEndSignal</td>
</tr>
<tr>
<td>MAP-SEND-AUTHENTICATION-INFO</td>
<td>sendAuthenticationInfo</td>
</tr>
<tr>
<td>MAP-SEND-IMSI</td>
<td>sendIMSI</td>
</tr>
<tr>
<td>MAP-SEND-IDENTIFICATION</td>
<td>sendIdentification</td>
</tr>
<tr>
<td>MAP-SEND-ROUTING-INFO-FOR-SM</td>
<td>sendRoutingInfoForSM</td>
</tr>
<tr>
<td>MAP-SEND-ROUTING-INFO</td>
<td>sendRoutingInfo</td>
</tr>
<tr>
<td>MAP-UNSTRUCTURED-SS-NOTIFY</td>
<td>unstructuredSS-Notify</td>
</tr>
<tr>
<td>MAP-UNSTRUCTURED-SS-REQUEST</td>
<td>unstructuredSS-Request</td>
</tr>
<tr>
<td>MAP-UPDATE-LOCATION</td>
<td>updateLocation</td>
</tr>
</tbody>
</table>
13.2.2.5  Error

The error parameter in a TC-U-ERROR indication primitive is mapped to the user error parameter in the MAP confirm primitive of the service associated with the operation to which the error is attached.

The user error parameter in MAP response primitives is mapped to the error parameter of the TC-U-ERROR request primitive, except for "initiating-release" and "resource-limitation" which are mapped to the problem code parameter of the TC-U-REJECT request primitive.

13.2.2.6  Parameters

The parameters of MAP specific request and indication primitives are mapped to the argument parameter of TC-INVOKE primitives.

The parameters of MAP specific response and confirm primitives are mapped to the result parameter of TC-RESULT-L primitives, the parameter of TC-U-ERROR primitives or the argument of TC-INVOKE primitives when mapping on linked class 4 operations is used.

13.2.2.7  Time out

The value of this parameter is set by the MAP PM according to the type of operation invoked.

13.2.2.8  Last component

This parameter is used by the MAP PM as described in CCITT Recommendation Q.711. It is not visible from the MAP user.

13.2.2.9  Problem code

13.2.2.9.1  Mapping to MAP User Error

The following values of the user error parameter are mapped as follows to values of the TC problem code parameter. These values are generated by the MAP user. This mapping is valid from the TC-U-REJECT indication primitive to the MAP confirm service primitive and from the MAP response service primitive to the TC-U-REJECT request primitive.

<table>
<thead>
<tr>
<th>MAP User Error</th>
<th>TC problem code</th>
</tr>
</thead>
<tbody>
<tr>
<td>resource limitation</td>
<td>resource limitation</td>
</tr>
<tr>
<td>initiating release</td>
<td>initiating release</td>
</tr>
</tbody>
</table>

13.2.2.9.2  Mapping to MAP Provider Error parameter

The following values of the TC problem code parameter of the TC-U-REJECT indication primitive are mapped as follows to values of the MAP Provider Error parameter of the MAP confirm primitive.

<table>
<thead>
<tr>
<th>TC problem code</th>
<th>MAP Provider Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>duplicated invoke Id</td>
<td>duplicated invoke id</td>
</tr>
<tr>
<td>unrecognized operation</td>
<td>service not supported</td>
</tr>
<tr>
<td>mistyped parameter</td>
<td>mistyped parameter</td>
</tr>
</tbody>
</table>

The following values of the problem code parameters of the TC-L-REJECT primitive are mapped to values of the provider error parameter of the MAP confirm primitive as follows:
Table 13.2/4: Mapping of TC problem code in TC-L-REJECT on to MAP Provider Error parameter

<table>
<thead>
<tr>
<th>TC problem code</th>
<th>Map Provider Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>return result unexpected</td>
<td>unexpected response from the peer</td>
</tr>
<tr>
<td>return error unexpected</td>
<td>unexpected response from the peer</td>
</tr>
</tbody>
</table>

13.2.2.9.3 Mapping to diagnostic parameter

The following values of the problem code parameter of the TC-R-REJECT and TC-U-REJECT primitive are mapped to values of the diagnostic parameter of the MAP-NOTICE indication primitive as follows.

Table 13.2/5: Mapping of TC problem code of TC-R-REJECT and TC-U-REJECT on to diagnostic parameter

<table>
<thead>
<tr>
<th>TC problem code</th>
<th>MAP diagnostic</th>
</tr>
</thead>
<tbody>
<tr>
<td>General problem</td>
<td>abnormal event detected by the peer</td>
</tr>
<tr>
<td>Invoke problem</td>
<td>- unrecognized linked ID</td>
</tr>
<tr>
<td></td>
<td>- linked response unexpected</td>
</tr>
<tr>
<td></td>
<td>- unexpected linked operation</td>
</tr>
<tr>
<td>Return result problem</td>
<td>- unrecognized invoke ID</td>
</tr>
<tr>
<td></td>
<td>- return result unexpected</td>
</tr>
<tr>
<td></td>
<td>- mistyped parameter</td>
</tr>
<tr>
<td>Return error problem</td>
<td>- unrecognized invoke ID</td>
</tr>
<tr>
<td></td>
<td>- return error unexpected</td>
</tr>
<tr>
<td></td>
<td>- unrecognized error</td>
</tr>
<tr>
<td></td>
<td>- unexpected error</td>
</tr>
<tr>
<td></td>
<td>- mistyped parameter</td>
</tr>
</tbody>
</table>

The following values of the problem code parameter of the TC-L-REJECT primitive are mapped to values of the diagnostic parameter of the MAP-NOTICE indication primitive as follows.

Table 13.2/6: Mapping of TC problem code of TC-L-REJECT on to diagnostic parameter

<table>
<thead>
<tr>
<th>TC problem code</th>
<th>MAP diagnostic</th>
</tr>
</thead>
<tbody>
<tr>
<td>General problems:</td>
<td>- abnormal event received from the peer</td>
</tr>
<tr>
<td>Invoke problem:</td>
<td>- unrecognized linked ID</td>
</tr>
<tr>
<td>Return result problem:</td>
<td>- abnormal event received from the peer</td>
</tr>
<tr>
<td>Return error problem:</td>
<td>- abnormal event received from the peer</td>
</tr>
</tbody>
</table>

13.3 SDL descriptions

The following SDL specification describes a system which includes three blocks: MAP-user, MAP-provider and TC. Such a system resides in each network component supporting MAP and communicates with its peers via the lower layers of the signalling network which are part of the environment.

Only the MAP-provider is fully described in this subclause. The various type of processes which form the MAP-User block and the TC block are described respectively in clauses 15 to 21 of this ETS and in CCITT Recommendation Q.774.

The MAP-Provider block communicates with the MAP_USER via two channels U1 and U2. Via U1 the MAP-provider receives the MAP request and response primitives. Via U2 it sends the MAP indication and confirm primitives.
The MAP-Provider block communicates with TC via two channels P1 and P2. Via P1 the MAP-Provider sends all the TC request primitives. Via P2 it receives all the TC indication primitives.

The MAP-Provider block is composed of the four following types of processes:

a) MAP_DSM: This type of process handles a dialogue. There exists one process instance per MAP dialogue.
b) LOAD_CTRL: This type of process is in charge of load control. There is only one instance of this process in each system.
c) PERFORMING_MAP_SSM: This type of process handle a MAP service performed during a dialogue. An instance of this process is created by the instance of the MAP_DSM process for each MAP-service to be performed.
d) REQUESTING_MAP_SSM: This type of process handle a MAP service requested during a dialogue. An instance of this process is created by the instance of the MAP_DSM process for each requested MAP-service.

A process MAP_DSM exchanges external signals with other blocks as well as internal signals with the other processes of the MAP-Provider block. The external signals are either MAP service primitives or TC service primitives.

The signal routes used by the various processes are organized as follows:

a) A process MAP_DSM receives and sends events from/to the MAP_user via signal route User1/User2. These routes uses respectively channel U1 and U2.
b) A process MAP_DSM receives and sends events from/to the TC via signal route Tc1/Tc2. These routes uses respectively channel P1 and P2.
c) A process MAP_DSM receives and sends events from/to the LOAD_CTRL process via signal route Load1/Load2. These routes are internal.
d) A process MAP_DSM sends events to the PERFORMING_MAP_SSM processes via signal route Intern1. This route is internal.
e) A process MAP_DSM sends events to the REQUESTING_MAP_SSM processes via signal route Intern2. This route is internal.
f) A process MAP_PERFORMING_SSM sends events to the MAP_USER via signal route User4. This route uses channel U2.
g) A process MAP_PERFORMING_SSM sends events to TC via signal route Tc3. This route uses channel P1.
h) A process MAP_REQUESTING_SSM sends events to the MAP_USER via signal route User5. This route uses channel U2.
j) A process MAP_REQUESTING_SSM sends events to TC via signal route Tc4. This route uses channel P1.
Figure 13.2/1: System MAP_STACK

Figure 13.2/2: Block MAPPROVIDER
Figure 13.2/3 (sheet 1 of 11): Process MAP_DSM
Figure 13.2/3 (sheet 2 of 11): Process MAP_DSM
Figure 13.2/3 (sheet 3 of 11): Process MAP_DSM
Figure 13.2/3 (sheet 4 of 11): Process MAP_DSM
Figure 13.2/3 (sheet 5 of 11): Process MAP_DSM
MAP-Provider_AbortReason := abnormalDialogue

MAP_OPEN_CNF_VIA_USER2

User Information := MAP_Provider_Abort_Info

TC_U_ABORT_REQ_VIA_TC1

DIALOGUE_ESTABLISHED

MAP_PROV_ABORT_IND_VIA_USER2

Abort_Reason := User_Specific

TC_CONTINUE_IND

the first one

AC name unchanged

yes

no

PROCESS_COMPONENTS

MAP_DELIMITER_Ind

DIALOGUE_ESTABLISHED

Figure 13.2/3 (sheet 6 of 11): Process MAP_DSM
Figure 13.2/3 (sheet 7 of 11): Process MAP_DSM
Figure 13.2/3 (sheet 8 of 11): Process MAP_DSM
Abort_reason := User_specific
User_info := MAP-UserAbortInfo

Abort_reason := User_Specific
Build_MAP_Refuse_PDU

Abort_reason := User_specific
User_info := MAP-UserAbortInfo

Map_U_ABORT_REQ

MAP_OPEN_RSP

Result

refused

accepted

no

yes

Build_MAP_ACCEPT_PDU

DIALOGUE_PENDING

DIALOGUE_ACCEPTED

Figure 13.2/3 (sheet 9 of 11): Process MAP_DSM
Figure 13.2/3 (sheet 10 of 11): Process MAP_DSM
Figure 13.2/3 (sheet 11 of 11): Process MAP_DSM
Comments: Components from TCAP:
- DCL
- OP_CODE INTEGER,
- OP_EXIST, LAST_COMPONENT, INVOKEID_ASS, LINKEDID_PREF, LINKEDID_ASS BOOLEAN.

Figure 13.2/4 (sheet 1 of 4): Procedure PROCESS_COMPONENTS
Figure 13.2/4 (sheet 2 of 4): Procedure PROCESS_COMPONENTS
Figure 13.2/4 (sheet 3 of 4): Procedure PROCESS_COMPONENTS
Figure 13.2/4 (sheet 4 of 4): Procedure PROCESS_COMPONENTS
Figure 13.2/5: Process LOAD_CTRL
Figure 13.2/6 (sheet 1 of 3): Process PERFORMING_MAP_SSM
Figure 13.2/6 (sheet 2 of 3): Process PERFORMING_MAP_SSM
Figure 13.2/6 (sheet 3 of 3): Process PERFORMING_MAP_SSM
Comment MAP Service State Machine:

ARGUMENT_CORRECT, ERROR_CODE_CORRECT, LINKED_REQ_DEF, SYNTAX_CORRECT, MAP_INITIATED, CNF, LINKED_OPERATION_ALLOWED BOOLEAN, OP_CLASS INTEGER;

Figure 13.2/7 (sheet 1 of 4): Process REQUESTING_MAP_SSM
Figure 13.2/7 (sheet 2 of 4): Process REQUESTING_MAP_SSM
Figure 13.2/7 (sheet 3 of 4): Process REQUESTING_MAP_SSM
Figure 13.2/7 (sheet 4 of 4): Process REQUESTING_MAP_SSM
14  Abstract syntax of the MAP protocol

14.1  General

This subclause specifies the Abstract Syntaxes for the Mobile Application Part as well as the associated set of Operations and Errors, using the Abstract Syntax Notation One (ASN.1), defined in CCITT Recommendation X.208 (1988) or X.680 (1994) with additions as defined in subclause 14.1.4 on Compatibility Considerations and the OPERATION and ERROR external MACROs, defined in CCITT Recommendation Q.773.

The Abstract Syntax is defined for all interfaces specified in subclause 2.4 except for the A- and B-interfaces.

The Mobile Application Part protocol is defined by two Abstract Syntaxes:

- one Abstract Syntax which encompass all Operations; and
- Errors identified by the various MAP subsystem numbers.

This Abstract Syntax represents the set of values each of which is a value of the ASN.1 type TCAPMessages.MessageType as defined in CCITT Recommendation Q.773 with the ANY DEFINED BY sections resolved by the operation and error codes included in the ASN.1 module MAP-Protocol. However, only the subset of this abstract syntax which is required by the procedures defined for an entity needs to be supported:

- one Abstract Syntax identified by the OBJECT IDENTIFIER value MAP-DialogueInformation.map-DialogueAS.

This Abstract Syntax represents the set of values each of which is a value of the ASN.1 type MAP-DialogueInformation.MAP-DialoguePDU. Such a value of the ASN.1 single-ASN.1-type element is contained within the user-information element of the TCAPMessages.DialoguePortion ASN.1 type. This Abstract Syntax name is to be used as a direct reference.

14.1.1  Encoding rules

The encoding rules which are applicable to the defined Abstract Syntaxes are the Basic Encoding Rules for Abstract Syntax Notation One, defined in CCITT Recommendation X.690 with the same exceptions as in CCITT Recommendation Q.773 section 4 Message Representation.

When the definite form is used for length encoding, a data value of length less than 128 octets must have the length encoded in the short form.

When the long form is employed to code a length, the minimum number of octets shall be used to code the length field.

OCTET STRING values and BIT STRING values must be encoded in a primitive form.

There is no restriction to the use of empty constructors (e.g. an empty SEQUENCE type). That is, the encoding of the content of any data value shall consist of zero, one or more octets.

14.1.2  Use of TC

The mapping of OPERATION and ERROR to TC components is defined in ETS 300 287 (version 2) which is based on CCITT Recommendation Q.773 (1992).

NOTE 1: The class of an operation is not stated explicitly but is specified as well in the ASN.1 operation type definition.

Class 1: RESULT and ERROR appear in ASN.1 operation type definition.

Class 2: only ERROR appears in ASN.1 operation type definition.

Class 3: only RESULT appears in ASN.1 operation type definition.

Class 4: both RESULT and ERROR do not appear in ASN.1 operation type definition.
The ASN.1 data type which follows the keywords "ARGUMENT", "PARAMETER" or "RESULT" (for OPERATION and ERROR) is always optional from a syntactic point of view. However, except when specifically mentioned with the ASN.1 comment «-- optional», the «parameter» part of a component has to be considered as mandatory from a semantic point of view.

When an optional element is missing in an invoke component or in an inner data structure while it is required by the context, an error component is returned if specified in the operation type; the associated type of error is DataMissing. This holds also when the entire parameter of an invoke component is missing while it is required by the context.

NOTE 2: When a mandatory element is missing in the parameter or inner data structure of any component, a reject component is returned (if the dialogue still exists). The problem code to be used is "Mistyped parameter".

The Timer Values used in the operation type definitions are indicated as ASN.1 comment. The Timer Value Ranges are:

- \(s\) = from 3 seconds to 10 seconds;
- \(m\) = from 15 seconds to 30 seconds;
- \(ml\) = from 1 minute to 10 minutes;
- \(l\) = from 28 hours to 38 hours.

### 14.1.2.1 Use of Global Operation and Error codes defined outside MAP

An entity supporting an application context greater than 2 shall be capable of receiving an operation or error code, within an application context defined in GSM 09.02, encoded as either an Object Identifier (as defined in CCITT Recommendation X.690 (1994)) or an integer value (as defined in section 14.5). Related restrictions regarding the use of Object Identifiers are as follows:

- The length of the Object Identifier shall not exceed 16 octets and the number of components of the Object Identifier shall not exceed 16.
- Object Identifiers shall be used only for operations or errors defined outside of GSM 09.02
- Global error codes may be sent only in response to a global operation. If a standard operation is received then a global error code shall not be sent in response.

Handling of an unknown operation codes by the receiving entity is defined in section 12.1.1

### 14.1.3 Use of information elements defined outside MAP

An information element or a set of information elements (messages) transparently carried in the Mobile Application Part but defined in other recommendation/technical specifications are handled in one of the following ways:

i) The contents of each information element (without the octets encoding the identifier and the length in the recommendation/technical specification where it is defined) is carried as the value of an ASN.1 NamedType derived from the OCTET STRING data type. Additionally, the internal structure may be explained by means of comments. In case of misalignment the referred to recommendation/technical specification takes precedence.

ii) The complete information element (including the octets encoding the identifier and the length in the recommendation/technical specification where it is defined) or set of information elements and the identity of the associated protocol are carried as the value of the ExternalSignalInfo data type defined in this ETS. Where more than one information element is carried, the information elements are sent contiguously with no filler octets between them.

### 14.1.4 Compatibility considerations

The following ASN.1 modules conform to CCITT Recommendation X.208 (1988) or X.680 (1994) (the only module which makes use of X.680 is MAP-ExtensionDataTypes), but in addition Ellipsis Notation ("..." - notation) is used as described in ITU-T Recommendation X.680 Amendment 1 (1995) wherever future protocol extensions are foreseen.

The "..." construct applies only to SEQUENCE and ENUMERATED data types. An entity supporting a version greater than 1 shall not reject an unsupported extension following "..." of that SEQUENCE or ENUMERATED data type. The
Encoding Rules from subclause 14.1.1 apply to every element of the whole Transfer Syntax especially to the ASN.1 type EXTERNAL.

Private extensions shall:

1) if included in operations of an AC of V2, follow the extension marker and be tagged using PRIVATE tags up to and including 29.

NOTE 1: This type of extension is in most cases used only within a PLMN.

2) if included in operations of an AC of V3 or higher: be included only in the Private Extension Container that is defined in the specification.

NOTE 2: This type of extension can be used between PLMNs.

Private extensions shall not be included in v2 supplementary service operations.

PCS extensions shall be included in the PCS Extension Container that is defined in this specification.

In order to improve extensibility, a few error parameters have been defined as a CHOICE between the version 2 description and a SEQUENCE including the version 2 description and an extension container. Operations used in a v2-application-context must consider only the first alternative while operations used in a vn-application-context (n>2) must consider only the second alternative.

14.1.5 Structure of the Abstract Syntax of MAP

For each MAP parameter which has to be transferred by a MAP Protocol Data Unit (MAP message), there is a PDU field (an ASN.1 NamedType) whose ASN.1 identifier has the same name as the corresponding parameter, except for the differences required by the ASN.1 notation (blanks between words are removed or replaced by hyphen, the first letter of the first word is lower-case and the first letter of the following words are capitalized, e.g. "no reply condition time" is mapped to "noReplyConditionTime"). Additionally some words may be abbreviated as follows:

bs basic service
ch call handling
cug closed user group
ho handover
ic incoming call
id identity
info information
ms mobile service
oc outgoing call
om operation & maintenance
pw Password
sm short message service
ss supplementary service

The MAP protocol is composed of several ASN.1 modules dealing with either operations, errors, data types, and, if applicable, split into those dealing with mobile services, call handling services, supplementary services and short message services. For operations and errors no values are assigned, but only the operation and error types in order to allow use of the defined types also by other protocols (e.g. TS GSM 04.80). The values (operation codes and error codes) are defined in a separate module. The ASN.1 source lines are preceded by line-numbers at the left margin in order to enable the usage of the cross-reference in appendix A.

The module containing the definition of the operation packages for MAP is:
1. MAP-OperationPackages.

The module containing the definition of the application contexts for MAP is:

2. MAP-ApplicationContexts.

The module containing the data types for the Abstract Syntax to be used for TCAPMessages.DialoguePortion for MAP is:

3. MAP-DialogueInformation.

The module containing the operation codes and error codes for MAP is:

4. MAP-Protocol.

The modules containing all operation type definitions for MAP are:

5. MAP-MobileServiceOperations;
6. MAP-OperationAndMaintenanceOperations;
7. MAP-CallHandlingOperations;
8. MAP-SupplementaryServiceOperations;
9. MAP-ShortMessageServiceOperations.

The module containing all error type definitions for MAP is:

10. MAP-Errors.

Modules containing all data type definitions for MAP are:

11. MAP-MS-DataTypes;
12. MAP-OM-DataTypes;
13. MAP-CH-DataTypes;
14. MAP-SS-DataTypes;
15. MAP-SS-Code;
16. MAP-SM-DataTypes;
17. MAP-ER-DataTypes;
18. MAP-CommonDataTypes;
19. MAP-TS-Code;
20. MAP-BS-Code;
21. MAP-ExtensionDataTypes.

References are made also to modules defined outside of this ETS. They are defined in the technical specification Mobile Services Domain and technical specification Transaction Capability respectively:

MobileDomainDefinitions;
TCAPMessages;
DialoguePDUs.
### 14.1.6 Application Contexts

The following informative table lists the latest versions of the Application Contexts used in this specification, with the operations used by them and, where applicable, whether or not the operation description is exactly the same as for previous versions. Information in sections 14.6 & 14.7 relates only to the ACs in this table.

<table>
<thead>
<tr>
<th>AC Name</th>
<th>AC Version</th>
<th>Operations Used</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>locationCancellationContext</td>
<td>v2</td>
<td>cancelLocation</td>
<td></td>
</tr>
<tr>
<td>equipmentMngtContext</td>
<td>v2</td>
<td>checkIMEI</td>
<td></td>
</tr>
<tr>
<td>imsiRetrievalContext</td>
<td>v2</td>
<td>sendIMSI</td>
<td></td>
</tr>
<tr>
<td>infoRetrievalContext</td>
<td>v2</td>
<td>sendAuthenticationInfo</td>
<td></td>
</tr>
<tr>
<td>interVlrInfoRetrievalContext</td>
<td>v2</td>
<td>sendIdentification</td>
<td></td>
</tr>
<tr>
<td>handoverControlContext</td>
<td>v2</td>
<td>prepareHandover</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>forwardAccessSignalling</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>sendEndSignal</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>processAccessSignalling</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>prepareSubsequentHandover</td>
<td></td>
</tr>
<tr>
<td>mwdMngtContext</td>
<td>v2</td>
<td>readyForSM</td>
<td></td>
</tr>
<tr>
<td>msPurgingContext</td>
<td>v2</td>
<td>purgeMS</td>
<td></td>
</tr>
<tr>
<td>shortMsgAlertContext</td>
<td>v2</td>
<td>alertServiceCentre</td>
<td></td>
</tr>
<tr>
<td>resetContext</td>
<td>v2</td>
<td>reset</td>
<td></td>
</tr>
<tr>
<td>networkUnstructuredSsContext</td>
<td>v2</td>
<td>processUnstructuredSS-Request</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>unstructuredSS-Request</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>unstructuredSS-Notify</td>
<td></td>
</tr>
<tr>
<td>tracingContext</td>
<td>v3</td>
<td>activateTraceMode</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>deactivateTraceMode</td>
<td></td>
</tr>
<tr>
<td>networkFunctionalSsContext</td>
<td>v2</td>
<td>registerSS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>eraseSS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>activateSS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>deactivateSS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>registerPassword</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>interrogateSS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>getPassword</td>
<td></td>
</tr>
<tr>
<td>shortMsgMO-RelayContext</td>
<td>v3</td>
<td>mo-forwardSM</td>
<td></td>
</tr>
<tr>
<td>shortMsgMT-RelayContext</td>
<td>v3</td>
<td>mt-forwardSM</td>
<td></td>
</tr>
<tr>
<td>shortMsgGatewayContext</td>
<td>v3</td>
<td>sendRoutingInfoForSM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>reportSM-DeliveryStatus</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>InformServiceCentre</td>
<td></td>
</tr>
<tr>
<td>networkLocUpContext</td>
<td>v3</td>
<td>updateLocation</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>forwardCheckSs-Indication</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>restoreData</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>insertSubscriberData</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>activateTraceMode</td>
<td></td>
</tr>
<tr>
<td>subscriberDataMngtContext</td>
<td>v3</td>
<td>insertSubscriberData</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>deleteSubscriberData</td>
<td></td>
</tr>
<tr>
<td>roamingNumberEnquiryContext</td>
<td>v3</td>
<td>provideRoamingNumber</td>
<td></td>
</tr>
<tr>
<td>locationInfoRetrievalContext</td>
<td>v3</td>
<td>sendRoutingInfo</td>
<td></td>
</tr>
<tr>
<td>callControlTransferContext</td>
<td>v3</td>
<td>resumeCallHandling</td>
<td></td>
</tr>
<tr>
<td>subscriberInfoEnquiryContext</td>
<td>v3</td>
<td>provideSubscriberInfo</td>
<td></td>
</tr>
<tr>
<td>anyTimeEnquiryContext</td>
<td>v3</td>
<td>anyTimeInterrogation</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE (\*: The syntax of the operations is not the same in previous versions unless explicitly stated.**
14.2 Operation packages

14.2.1 General aspects

This subclause describes the operation-packages which are used to build the application-contexts defined in subclause 14.3.

Each operation-package is a specification of the roles of a pair of communicating objects (i.e. a pair of MAP-Providers), in terms of operations which they can invoke of each other.

The grouping of operations into one or several packages does not necessarily imply any grouping in terms of Application Service Elements.

The following ASN.1 MACRO is used to describe operation-packages in this subclause:

```
OPERATION-PACKAGE MACRO ::=
BEGIN
TYPE NOTATION ::= Symmetric | ConsumerInvokes SupplierInvokes | empty
VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)
Symmetric ::= "OPERATIONS" "{" OperationList "}"
ConsumerInvokes ::= "CONSUMER INVOKES" "{" OperationList "}"
SupplierInvokes ::= "SUPPLIER INVOKES" "{" OperationList "}" | empty
OperationList ::= Operation | OperationList "," Operation
Operation ::= value(OPERATION)
END
```

Since the application-context definitions provided in subclause 14.3 use only an informal description technique, only the type notation is used in the following subclauses to define operation-packages.

The following definitions are used throughout this subclause (n>=2):

- v1-only operation: An operation which shall be used only in v1 application-contexts;
- vn-only operation: An operation which shall be used only in vn application-contexts;
- v(n-1)-operation: An operation whose specification has not been modified since the MAP v(n-1) specifications or if the modifications are considered as not affecting v(n-1) implementations;
- v(n-1)-equivalent operation: The version of an operation which excludes all the information elements and errors which have been added since the MAP v(n-1) specification;
- vn-only package: An operation package which contains only vn-only operations;
- v(n-1)-package: An operation package which contains only v(n-1)-operations.

The names of vn-packages are suffixed by "-vn" where n>=2.

For each operation package which is not vn-only (n>=2) and which does not include only v(n-1)-operations, there is a v(n-1)-equivalent package. Except when a definition is explicitly provided in the following subclauses, the v(n-1)-equivalent package includes the v(n-1)-equivalent operations of the operations which belong to this package.
14.2.2 Packages specifications

14.2.2.1 Location updating

This operation package includes the operations required for location management procedures between HLR and VLR.

LocationUpdatingPackage-v3 ::= OPERATION-PACKAGE
  -- Supplier is HLR if Consumer is VLR
  CONSUMER INVOKES {
    updateLocation
  }
  SUPPLIER INVOKES {
    forwardCheckSs-Indication
  }

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 14.2.1.

14.2.2.2 Location cancellation

This operation package includes the operations required for location cancellation and MS purging procedures between HLR and VLR.

LocationCancellationPackage-v2 ::= OPERATION-PACKAGE
  -- Supplier is VLR if Consumer is HLR
  CONSUMER INVOKES {
    cancelLocation
  }

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.3 Roaming number enquiry

This operation package includes the operations required for roaming number enquiry procedures between HLR and VLR.

RoamingNumberEnquiryPackage-v3 ::= OPERATION-PACKAGE
  -- Supplier is VLR if Consumer is HLR
  CONSUMER INVOKES {
    provideRoamingNumber
  }

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 14.2.1.

14.2.2.4 Information retrieval

This operation package includes the operation required for the authentication information retrieval procedure between HLR and VLR.

InfoRetrievalPackage-v2 ::= OPERATION-PACKAGE
  -- Supplier is HLR if Consumer is VLR
  CONSUMER INVOKES {
    sendAuthenticationInfo
  }

The v1-equivalent package is defined as follows:

InfoRetrievalPackage-v1 ::= OPERATION-PACKAGE
  -- Supplier is HLR or VLR if Consumer is VLR
  CONSUMER INVOKES {
    sendParameters
  }

14.2.2.5 Inter-VLR information retrieval

This operation package includes the operations required for inter VLR information retrieval procedures.

InterVlrInfoRetrievalPackage-v2 ::= OPERATION-PACKAGE
  -- Supplier is VLR if Consumer is VLR
  CONSUMER INVOKES {
    sendIdentification
  }

The v1-equivalent package is : InfoRetrievalPackage-v1
14.2.2.6 IMSI retrieval

This operation package includes the operation required for the IMSI retrieval procedure between HLR and VLR.

```
IMSIRetrievalPackage-v2 ::= OPERATION-PACKAGE
   -- Supplier is HLR if Consumer is VLR
   CONSUMER INVOKES {
      sendIMSI
   }
```

This package is v2 only.

14.2.2.7 Call control transfer

This operation package includes the operation required for the call control transfer procedure between VMSC and GMSC.

```
CallControlTransferPackage-v3 ::= OPERATION-PACKAGE
   -- Supplier is GMSC if Consumer is VMSC
   CONSUMER INVOKES {
      resumeCallHandling
   }
```

This package is v3 only.

14.2.2.8 - 14.2.2.9 [Spare]

14.2.2.10 Interrogation

This operation package includes the operations required for interrogation procedures between MSC and HLR.

```
InterrogationPackage-v3 ::= OPERATION-PACKAGE
   -- Supplier is HLR if Consumer is MSC
   CONSUMER INVOKES {
      sendRoutingInfo
   }
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 14.2.1.

14.2.2.11 [Spare]

14.2.2.12 Handover Control

This operation package includes the operations required for handover procedures between MSCs.

```
HandoverControlPackage-v2 ::= OPERATION-PACKAGE
   -- Supplier is MSCB if Consumer is MSCA
   CONSUMER INVOKES {
      prepareHandover,
      forwardAccessSignalling
   }
   SUPPLIER INVOKES {
      sendEndSignal,
      processAccessSignalling,
      prepareSubsequentHandover
   }
```

The v1-equivalent package is defined as follows.

```
HandoverControlPackage-v1 ::= OPERATION-PACKAGE
   -- Supplier is MSCB if Consumer is MSCA
   CONSUMER INVOKES {
      performHandover,
      forwardAccessSignalling,
      traceSubscriberActivity
   }
   SUPPLIER INVOKES {
      sendEndSignal,
      noteInternalHandover,
      processAccessSignalling,
      performSubsequentHandover
   }
```
14.2.2.13  Subscriber Data management stand alone

This operation package includes the operations required for stand alone subscriber data management procedures between HLR and VLR.

```
SubscriberDataMngtStandalonePackage-v3 ::= OPERATION-PACKAGE
   -- Supplier is VLR if Consumer is HLR
   CONSUMER INVOKES {
      insertSubscriberData,
      deleteSubscriberData}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 14.2.1.

14.2.2.14  Equipment management

This operation package includes the operations required for equipment management procedures between EIR and MSC.

```
EquipmentMngtPackage-v2 ::= OPERATION-PACKAGE
   -- Supplier is EIR if Consumer is MSC
   CONSUMER INVOKES {
      checkIMEI}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.15  Subscriber data management

This operation package includes the operations required for subscriber data management procedures between HLR and VLR.

```
SubscriberDataMngtPackage-v3 ::= OPERATION-PACKAGE
   -- Supplier is VLR if Consumer is HLR
   CONSUMER INVOKES {
      insertSubscriberData}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 14.2.1.

14.2.2.16  Location register restart

This operation package includes the operations required for location register restart procedures between HLR and VLR.

```
ResetPackage-v2 ::= OPERATION-PACKAGE
   -- Supplier is VLR if Consumer is HLR
   CONSUMER INVOKES {
      reset}
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.17  Tracing stand-alone

This operation package includes the operations required for stand alone tracing procedures between HLR and VLR.

```
TracingStandAlonePackage-v3 ::= OPERATION-PACKAGE
   -- Supplier is VLR if Consumer is HLR
   CONSUMER INVOKES {
      activateTraceMode,
      deactivateTraceMode}
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 14.2.1.
14.2.2.18 Functional SS handling

This operation package includes the operations required for functional supplementary services procedures between VLR and HLR.

```
FunctionalSsPackage-v2 ::= OPERATION-PACKAGE
   -- Supplier is HLR if Consumer is VLR
   CONSUMER INVOKES {
      registerSS,
      eraseSS,
      activateSS,
      deactivateSS,
      registerPassword,
      interrogateSS
   };
   SUPPLIER INVOKES {
      getPassword
   }
```

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.19 Tracing

This operation package includes the operations required for tracing procedures between HLR and VLR.

```
TracingPackage-v3 ::= OPERATION-PACKAGE
   -- Supplier is VLR if Consumer is HLR
   CONSUMER INVOKES {
      activateTraceMode
   }
```

The v1-equivalent and v2-equivalent packages can be determined according to the rules described in subclause 14.2.1.

14.2.2.20 Binding

This operation package includes the operation required to initialize a supplementary service procedure between VLR and HLR.

```
BindingPackage-v1 ::= OPERATION-PACKAGE
   -- Supplier is HLR if Consumer is VLR
   CONSUMER INVOKES {
      beginSubscriberActivity
   }
```

This package is v1 only.

14.2.2.21 Unstructured SS handling

This operation package includes the operations required for unstructured supplementary services procedures between VLR and HLR.

```
UnstructuredSsPackage-v2 ::= OPERATION-PACKAGE
   -- Supplier is HLR if Consumer is VLR
   CONSUMER INVOKES {
      processUnstructuredSS-Request
   };
   SUPPLIER INVOKES {
      unstructuredSS-Request,
      unstructuredSS-Notify
   }
```

The v1-equivalent package is defined as follows:

```
UnstructuredSsPackage-v1 ::= OPERATION-PACKAGE
   -- Supplier is HLR if Consumer is VLR
   CONSUMER INVOKES {
      processUnstructuredSS-Data
   }
```
14.2.2.22  MO Short message relay services

This operation package includes the operations required for short message relay service procedures between IWMSC and VMSC or between GMSC and MSC.

| MOShortMsgRelayPackage-v3 ::= OPERATION-PACKAGE |
| --- | --- |
| --- Supplier is IWMSC if Consumer is MSC |
| CONSUMER INVOKES { |
| MO-forwardSM} |

The v2-equivalent package is defined as follows:

| ShortMsgRelayPackage-v2 ::= OPERATION-PACKAGE |
| --- | --- |
| --- Supplier is IWMSC if Consumer is MSC |
| -- Supplier is MSC if Consumer is GMSC |
| CONSUMER INVOKES { |
| forwardSM} |

The v1-equivalent package can be determined according to the rules described in subclause 14.2.1.

14.2.2.23  Short message gateway services

This operation package includes the operations required for short message service gateway procedures between MSC and HLR.

| ShortMsgGatewayPackage-v3 ::= OPERATION-PACKAGE |
| --- | --- |
| --- Supplier is HLR if Consumer is GMSC |
| CONSUMER INVOKES { |
| sendRoutingInfoForSM, |
| reportSM-DeliveryStatus} |
| SUPPLIER INVOKES { |
| informServiceCentre} |

The v2-equivalent package can be determined according to the rules described in subclause 14.2.1.

The v1-equivalent package is defined as follows:

| ShortMsgGatewayPackage-v1 ::= OPERATION-PACKAGE |
| --- | --- |
| --- Supplier is HLR if Consumer is GMSC |
| CONSUMER INVOKES { |
| sendRoutingInfoForSM |
| reportSM-DeliveryStatus} |

14.2.2.24  MT Short message relay services

This operation package includes the operations required for short message relay service procedures between GMSC and MSC.

| MTShortMsgRelayPackage-v3 ::= OPERATION-PACKAGE |
| --- | --- |
| --- Supplier is MSC if Consumer is GMSC |
| CONSUMER INVOKES { |
| MT-forwardSM} |

The v2-equivalent package is: ShortMsgRelayPackage-v2

14.2.2.25  [Spare]

14.2.2.26  Message waiting data management

This operation package includes the operations required for short message waiting data procedures between HLR and VLR.

| MwdMngtPackage-v2 ::= OPERATION-PACKAGE |
| --- | --- |
| --- Supplier is HLR if Consumer is VLR |
| CONSUMER INVOKES { |
| readyForSM} |
The v1-equivalent package is defined as follows:

```
MwdMngtPackage-v1 ::= OPERATION-PACKAGE
  -- Supplier is HLR if Consumer is VLR
  CONSUMER INVOKES {
    noteSubscriberPresent}
```

14.2.2.27 Alerting

This operation package includes the operations required for alerting between HLR and IWMSC.

```
AlertingPackage-v2 ::= OPERATION-PACKAGE
  -- Supplier is IWMSC if Consumer is HLR
  CONSUMER INVOKES {
    alertServiceCentre}
```

The v1-equivalent package is defined as follows.

```
AlertingPackage-v1 ::= OPERATION-PACKAGE
  -- Supplier is IWMSC if Consumer is HLR
  CONSUMER INVOKES {
    alertServiceCentreWithoutResult}
```

14.2.2.28 Data restoration

This operation package includes the operations required for VLR data restoration between HLR and VLR.

```
DataRestorationPackage-v3 ::= OPERATION-PACKAGE
  -- Supplier is HLR if Consumer is VLR
  CONSUMER INVOKES {
    restoreData}
```

The v2-equivalent package can be determined according to the rules described in subclause 14.2.1.

The v1-equivalent package is: InfoRetrievalPackage-v1

14.2.2.29 Purging

This operation package includes the operations required for purging between HLR and VLR.

```
PurgingPackage-v2 ::= OPERATION-PACKAGE
  -- Supplier is HLR if Consumer is VLR
  CONSUMER INVOKES {
    purgeMS}
```

This Package is v2 only.

14.2.2.30 Subscriber information enquiry

This operation package includes the operations required for subscriber information enquiry procedures between HLR and VLR.

```
SubscriberInformationEnquiryPackage-v3 ::= OPERATION-PACKAGE
  -- Supplier is VLR if Consumer is HLR
  CONSUMER INVOKES {
    provideSubscriberInfo}
```

This package is v3 only.
14.2.2.31 Any time information enquiry

This operation package includes the operations required for any time information enquiry procedures between gsmSCF and HLR.

AnyTimeInformationEnquiryPackage-v3 ::= OPERATION-PACKAGE
   -- Supplier is HLR if Consumer is gsmSCF
   CONSUMER INVOKES {
     anytimeInterrogation
   }

This package is v3 only.
14.3 Application contexts

14.3.1 General aspects

An application-context is assigned for each dialogue established by a MAP-user. In this ETS each application-context is assigned a name which is supplied in the MAP-OPEN Req primitive by the MAP-User and transmitted to the peer under certain circumstances.

The following ASN.1 MACRO is used to describe the main aspects of application-contexts in the following subclauses:

```
APPLICATION-CONTEXT MACRO ::= 
BEGIN
  TYPE NOTATION ::= Symmetric | InitiatorConsumerOf
                  ResponderConsumerOf | empty
  VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)
  Symmetric ::= "OPERATIONS OF" "{" PackageList "}" 
  InitiatorConsumerOf ::= "INITIATOR CONSUMER OF" "{" PackageList "}" 
  ResponderConsumerOf ::= "RESPONDER CONSUMER OF" "{" PackageList "}" 
                        | empty
  PackageList ::= Package | PackageList "," Package
  Package ::= value(OPERATION-PACKAGE)
               | type -- shall reference a package type
END
```

The following definitions are used throughout this subclause:

- `v1-application-context`: An application-context which contains only v1-packages and uses only TC v1 facilities;
- `v1 context set`: the set of v1-application-contexts defined in this ETS;
- `vn-application-context (n>=2)`: An application-context which contains only vn-packages;

The names of v1-application-contexts are suffixed by "-v1" while other names are suffixed by "-vn" where n>=2.

Application-contexts which do not belong to the v1 context set use v2 TC facilities.

The last component of each application-context-name (i.e. the last component of the object identifier value) assigned to an application-context which belongs to the v1 context set indicates explicitly "version1".

For each application-context which does not belong to the "v1 context set" there is a v1-equivalent application context. This is a v1-application-context which includes the v1-equivalents of the packages included in the original context.

Each application-context uses the abstract-syntax associated with the operation-packages it includes and uses the transfer-syntax derived from it by applying the encoding rules defined in subclause 14.1.1.

ACs which do not belong to the v1 context set require the support of the abstract-syntax identified by the object identifier value: MAP-DialogueInformation.map-Dialogue-AS defined in subclause 14.4.
14.3.2 Application context definitions

14.3.2.1 [Spare]

14.3.2.2 Location Updating

This application context is used between HLR and VLR for location updating procedures.

```

networkLocUpContext-v3 APPLICATION-CONTEXT
   -- Responder is HLR if Initiator is VLR
   INITIATOR CONSUMER OF {
      LocationUpdatingPackage-v3,
      DataRestorationPackage-v3
   }
   RESPONDER CONSUMER OF {
      SubscriberDataMngtPackage-v3
      TracingPackage-v3
   }::= {map-ac networkLocUp(1) version3(3)}

The following application-context-name is assigned to the v2-equivalent application-context:

{map-ac networkLocUp(1) version2(2)}

The following application-context-name is assigned to the v1-equivalent application-context:

{map-ac networkLocUp(1) version1(1)}

14.3.2.3 Location Cancellation

This application context is used between HLR and VLR for location cancellation procedures.

```

locationCancellationContext-v2 APPLICATION-CONTEXT
   -- Responder is VLR if Initiator is HLR
   INITIATOR CONSUMER OF {
      LocationCancellationPackage-v2
   }::= {map-ac locationCancel(2) version2(2)}

The following application-context-name is assigned to the v1-equivalent application-context:

map-ac locationCancel(2) version1(1)

14.3.2.4 Roaming number enquiry

This application context is used between HLR and VLR for roaming number enquiry procedures.

```

roamingNumberEnquiryContext-v3 APPLICATION-CONTEXT
   -- Responder is VLR if Initiator is HLR
   INITIATOR CONSUMER OF {
      RoamingNumberEnquiryPackage-v3
   }::= {map-ac roamingNbEnquiry(3) version3(3)}

The following application-context-name is assigned to the v2-equivalent application-context:

{map-ac roamingNbEnquiry(3) version2(2)}

The following application-context-name is assigned to the v1-equivalent application-context:

{map-ac roamingNbEnquiry(3) version1(1)}
14.3.2.5 [Spare]

14.3.2.6 Location Information Retrieval

This application-context is used between GMSC and HLR when retrieving location information.

```
locationInfoRetrievalContext-v3 APPLICATION-CONTEXT
   -- Responder is HLR if Initiator is GMSC
   INITIATOR CONSUMER OF {
      InterrogationPackage-v3
   }::= {map-ac locInfoRetrieval(5) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac locInfoRetrieval(5) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac locInfoRetrieval(5) version1(1)}
```

14.3.2.7 Call control transfer

This application context is used for the call control transfer procedure between the VMSC and the GMSC.

```
callControlTransferContext-v3 APPLICATION-CONTEXT
   -- Responder is GMSC if Initiator is VMSC
   INITIATOR CONSUMER OF {
      CallControlTransferPackage-v3
   }::= {map-ac callControlTransfer(6) version3(3)}
```

This application-context is v3 only.

14.3.2.8 - 14.3.2.10 [Spare]

14.3.2.11 Location registers restart

This application context is used for location register restart procedures.

```
resetContext-v2 APPLICATION-CONTEXT
   -- Responder is VLR if Initiator is HLR
   INITIATOR CONSUMER OF {
      ResetPackage-v2
   }::= {map-ac reset(10) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac reset(10) version1(1)}
```

14.3.2.12 Handover control

This application context is used for handover procedures between MSCs.

```
handoverControlContext-v2 APPLICATION-CONTEXT
   -- Responder is MSCB if Initiator is MSCA
   INITIATOR CONSUMER OF {
      HandoverControlPackage-v2
   }::= {map-ac handoverControl(11) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac handoverControl(11) version1(1)}
```
14.3.2.13  IMSI Retrieval

This application context is used for IMSI retrieval between HLR and VLR.

```
imsiRetrievalContext-v2 APPLICATION-CONTEXT
  -- Responder is HLR if Initiator is VLR
  INITIATOR CONSUMER OF {
    IMSIRetrievalPackage-v2
  } ::= {map-ac imsiRetrieval(26) version2(2)}
```

This application-context is v2 only.

14.3.2.14  Equipment Management

This application context is used for equipment checking between MSC and EIR:

```
equipmentMngtContext-v2 APPLICATION-CONTEXT
  -- Responder is EIR if Initiator is MSC
  INITIATOR CONSUMER OF {
    EquipmentMngtPackage-v2
  } ::= {map-ac equipmentMngt(13) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac equipmentMngt(13) version1(1)}
```

14.3.2.15  Information retrieval

This application context is used for authentication information retrieval between HLR and VLR.

```
infoRetrievalContext-v2 APPLICATION-CONTEXT
  -- Responder is HLR if Initiator is VLR
  INITIATOR CONSUMER OF {
    InfoRetrievalPackage-v2
  } ::= {map-ac infoRetrieval(14) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
  -- Responder is HLR if Initiator is VLR
  {map-ac infoRetrieval(14) version1(1)}
```

14.3.2.16  Inter-VLR information retrieval

This application context is used for information retrieval between VLRs.

```
interVlrInfoRetrievalContext-v2 APPLICATION-CONTEXT
  -- Responder is VLR if Initiator is VLR
  INITIATOR CONSUMER OF {
    InterVlrInfoRetrievalPackage-v2
  } ::= {map-ac interVlrInfoRetrieval(15) version2(2)}
```

The v1-equivalent application-context is:

```
  -- Responder is VLR if Initiator is VLR
  {map-ac infoRetrieval(14) version1(1)}
```

14.3.2.17  Stand Alone Subscriber Data Management

This application context is used for stand alone subscriber data management between HLR and VLR:

```
subscriberDataMngtContext-v3 APPLICATION-CONTEXT
  -- Responder is VLR if Initiator is HLR
  INITIATOR CONSUMER OF {
    SubscriberDataMngtStandAlonePackage-v3
  } ::= {map-ac subscriberDataMngt(16) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac subscriberDataMngt(16) version2(2)}
```
The following application-context-name is assigned to the v1-equivalent application-context:

{map-ac subscriberDataMngt(16) version1(1)}

14.3.2.18  Tracing

This application context is used for stand alone tracing control procedures:

```
tracingContext-v3 APPLICATION-CONTEXT
  -- Responder is VLR if Initiator is HLR
  INITIATOR CONSUMER OF {
    TracingStandAlonePackage-v3}
 ::= {map-ac tracing(17) version3(3)}
```

The following application-context-name is assigned to the v2-equivalent application-context:

{map-ac tracing(17) version2(2)}

The following application-context-name is assigned to the v1-equivalent application-context:

{map-ac tracing(17) version1(1)}

14.3.2.19  Network functional SS handling

This application context is used for functional-like SS handling procedures between VLR and HLR.

```
networkFunctionalSsContext-v2 APPLICATION-CONTEXT
  -- Responder is HLR, Initiator is VLR
  INITIATOR CONSUMER OF {
    FunctionalSsPackage-v2}
 ::= {map-ac networkFunctionalSs(18) version2(2)}
```

The v1-equivalent application-context is defined as follows:

```
networkFunctionalSsContext-v1 APPLICATION-CONTEXT
  -- Responder is HLR, Initiator is VLR
  -- Responder is VLR, Initiator is HLR
  OPERATIONS OF {
    FunctionalSsPackage-v1,
    UnstructuredSsPackage-v1,
    BindingPackage-v1}
 ::= {map-ac networkFunctionalSs(18) version1(1)}
```

14.3.2.20  Network unstructured SS handling

This application context is used for handling stimuli-like procedures between HLR and VLR.

```
networkUnstructuredSsContext-v2 APPLICATION-CONTEXT
  -- Responder is HLR, Initiator is VLR
  -- Responder is VLR, Initiator is HLR
  OPERATIONS OF {
    UnstructuredSsPackage-v2}
 ::= {map-ac networkUnstructuredSs(19) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

{map-ac networkFunctionalSs(18) version1(1)}

14.3.2.21  Short Message Gateway

This application context is used for short message gateway procedures.

```
shortMsgGatewayContext-v3 APPLICATION-CONTEXT
  -- Responder is HLR if Initiator is GMSC
  INITIATOR CONSUMER OF {
    ShortMsgGatewayPackage-v3}
 ::= {map-ac shortMsgGateway(20) version3(3)}
```
The following application-context-name is assigned to the v2-equivalent application-context:

\{(map-ac shortMsgGateway(20) version2(2))\}

The following application-context-name is assigned to the v1-equivalent application-context:

\{(map-ac shortMsgGateway(20) version1(1))\}

14.3.2.22 Mobile originating Short Message Relay

This application context is used for mobile originating short message relay procedures.

\texttt{shortMsgMO-RelayContext-v3 APPLICATION-CONTEXT}

\hspace{1em} -- Responder is IWMSC if Initiator is MSC

\hspace{1em} INITIATOR CONSUMER OF { MOShortMsgRelayPackage-v3}

\hspace{1em} ::= \{(map-ac shortMsgMO-Relay(21) version3(3))\}

The following application-context-name is assigned to the v2-equivalent application-context:

\{(map-ac shortMsgMO-Relay(21) version2(2))\}

The following application-context-name is assigned to the v1-equivalent application-context:

\{(map-ac shortMsg-Relay(21) version1(1))\}

14.3.2.23 [Spare]

14.3.2.24 Short message alert

This application context is used for short message alerting procedures.

\texttt{shortMsgAlertContext-v2 APPLICATION-CONTEXT}

\hspace{1em} -- Responder is IWMSC if Initiator is HLR

\hspace{1em} INITIATOR CONSUMER OF { AlertingPackage-v2}

\hspace{1em} ::= \{(map-ac shortMsgAlert(23) version2(2))\}

The following application-context-name is symbolically assigned to the v1-equivalent application-context:

\{(map-ac shortMsgAlert(23) version1(1))\}

14.3.2.25 Short message waiting data management

This application context is used for short message waiting data management procedures.

\texttt{mwdMngtContext-v2 APPLICATION-CONTEXT}

\hspace{1em} -- Responder is HLR if Initiator is VLR

\hspace{1em} INITIATOR CONSUMER OF { MwdMngtPackage-v2}

\hspace{1em} ::= \{(map-ac mwdMngt(24) version2(2))\}

The following application-context-name is assigned to the v1-equivalent application-context:

\{(map-ac mwdMngt(24) version1(1))\}

14.3.2.26 Mobile terminating Short Message Relay

This application context is used for mobile terminating short message relay procedures.

\texttt{shortMsgMT-RelayContext-v3 APPLICATION-CONTEXT}

\hspace{1em} -- Responder is MSC if Initiator is GMSC

\hspace{1em} INITIATOR CONSUMER OF { MTShortMsgRelayPackage-v3}

\hspace{1em} ::= \{(map-ac shortMsgMT-Relay(25) version3(3))\}
The following application-context-name is assigned to the v2-equivalent application-context:

```
{map-ac shortMsgMT-Relay(25) version2(2)}
```

The following application-context-name is assigned to the v1-equivalent application-context:

```
{map-ac shortMsgMO-Relay(21) version1(1)}
```

### 14.3.2.27 MS purging

This application context is used between HLR and VLR for MS purging procedures.

```
msPurgingContext-v2 APPLICATION-CONTEXT
  -- Responder is HLR if Initiator is VLR
  INITIATOR CONSUMER OF {
purgingPackage-v2
::= {map-ac msPurging(27) version2(2)}
}
```

This application-context is v2 only.

### 14.3.2.28 Subscriber information enquiry

This application context is used between HLR and VLR for subscriber information enquiry procedures.

```
subscriberInfoEnquiryContext-v3 APPLICATION-CONTEXT
  -- Responder is VLR if Initiator is HLR
  INITIATOR CONSUMER OF {
  SubscriberInformationEnquiryPackage-v3
::= {map-ac subscriberInfoEnquiry(28) version3(3)}
}
```

This application-context is v3 only.

### 14.3.2.29 Any time information enquiry

This application context is used between gsmSCF and HLR for any time information enquiry procedures.

```
anyTimeInfoEnquiryContext-v3 APPLICATION-CONTEXT
  -- Responder is HLR if Initiator is gsmSCF
  INITIATOR CONSUMER OF {
  AnyTimeInformationEnquiryPackage-v3
::= {map-ac anyTimeInfoEnquiry(29) version3(3)}
}
```

This application-context is v3 only.
14.3.3 ASN.1 Module for application-context-names

The following ASN.1 module summarizes the application-context-name assigned to MAP application-contexts.

```asn1
MAP-ApplicationContexts {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-ApplicationContexts (2) version3 (3)
}

DEFINITIONS
BEGIN

-- EXPORTS everything

IMPORTS
  gsm-NetworkId,
  ac-Id
FROM MobileDomainDefinitions {
  ccitt (0) identified-organization (4) etsi (0) mobileDomain (0)
mobileDomainDefinitions (0) version1 (1)};

-- application-context-names

map-ac OBJECT IDENTIFIER ::= {gsm-NetworkId ac-Id}

networkLocUpContext-v3 OBJECT IDENTIFIER ::= 
  {map-ac networkLocUp(1) version3(3)}

locationCancellationContext-v2 OBJECT IDENTIFIER ::= 
  {map-ac locationCancel(2) version2(2)}

roamingNumberEnquiryContext-v3 OBJECT IDENTIFIER ::= 
  {map-ac roamingNbEnquiry(3) version3(3)}

locationInfoRetrievalContext-v3 OBJECT IDENTIFIER ::= 
  {map-ac locInfoRetrieval(5) version3(3)}

resetContext-v2 OBJECT IDENTIFIER ::= 
  {map-ac reset(10) version2(2)}

handoverControlContext-v2 OBJECT IDENTIFIER ::= 
  {map-ac handoverControl(11) version2(2)}

equipmentMngtContext-v2 OBJECT IDENTIFIER ::= 
  {map-ac equipmentMngt(13) version2(2)}

infoRetrievalContext-v2 OBJECT IDENTIFIER ::= 
  {map-ac infoRetrieval(14) version2(2)}

interVlrInfoRetrievalContext-v2 OBJECT IDENTIFIER ::= 
  {map-ac interVlrInfoRetrieval(15) version2(2)}

subscriberDataMngtContext-v3 OBJECT IDENTIFIER ::= 
  {map-ac subscriberDataMngt(16) version3(3)}

tracingContext-v3 OBJECT IDENTIFIER ::= 
  {map-ac tracing(17) version3(3)}

networkFunctionalSsContext-v2 OBJECT IDENTIFIER ::= 
  {map-ac networkFunctionalSs(18) version2(2)}

networkUnstructuredSsContext-v2 OBJECT IDENTIFIER ::= 
  {map-ac networkUnstructuredSs(19) version2(2)}

shortMsgGatewayContext-v3 OBJECT IDENTIFIER ::= 
  {map-ac shortMsgGateway(20) version3(3)}

END
```
shortMsgMO-RelayContext-v3 OBJECT IDENTIFIER ::=  
{map-ac shortMsgMO-Relay(21) version3(3)}

shortMsgAlertContext-v2 OBJECT IDENTIFIER ::=  
{map-ac shortMsgAlert(23) version2(2)}

mwdMngtContext-v2 OBJECT IDENTIFIER ::=  
{map-ac mwdMngt(24) version2(2)}

shortMsgMT-RelayContext-v3 OBJECT IDENTIFIER ::=  
{map-ac shortMsgMT-Relay(25) version3(3)}

imsiRetrievalContext-v2 OBJECT IDENTIFIER ::=  
{map-ac imsiRetrieval(26) version2(2)}

msPurgingContext-v2 OBJECT IDENTIFIER ::=  
{map-ac msPurging(27) version2(2)}

subscriberInfoEnquiryContext-v3 OBJECT IDENTIFIER ::=  
{map-ac subscriberInfoEnquiry(28) version3(3)}

anyTimeInfoEnquiryContext-v3 OBJECT IDENTIFIER ::=  
{map-ac anyTimeInfoEnquiry(29) version3(3)}

callControlTransferContext-v3 OBJECT IDENTIFIER ::=  
{map-ac callControlTransfer(6) version3(3)}

-- The following Object Identifiers are reserved for application-
-- contexts existing in previous versions of the protocol

-- AC Name & Version Object Identifier

--- networkLocUpContext-v1 map-ac networkLocUp (1) version1 (1)
-- locationCancellationContext-v1 map-ac locationCancellation (2) version1 (1)
-- roamingNumberEnquiryContext-v1 map-ac roamingNumberEnquiry (3) version1 (1)
-- locationInfoRetrievalContext-v1 map-ac locationInfoRetrieval (5) version1 (1)
-- resetContext-v1 map-ac reset (10) version1 (1)
-- handoverControlContext-v1 map-ac handoverControl (11) version1 (1)
-- equipmentMngtContext-v1 map-ac equipmentMngt (13) version1 (1)
-- infoRetrievalContext-v1 map-ac infoRetrieval (14) version1 (1)
-- subscriberDataMngtContext-v1 map-ac subscriberDataMngt (16) version1 (1)
-- subscriberDataMngtContext-v2 map-ac subscriberDataMngt (16) version2 (2)
-- tracingContext-v1 map-ac tracing (17) version1 (1)
-- tracingContext-v2 map-ac tracing (17) version2 (2)
-- networkFunctionalSsContext-v1 map-ac networkFunctionalSs (18) version1 (1)
-- shortMsgGatewayContext-v1 map-ac shortMsgGateway (20) version1 (1)
-- shortMsgGatewayContext-v2 map-ac shortMsgGateway (20) version2 (2)
-- shortMsgRelayContext-v1 map-ac shortMsgRelay (21) version1 (1)
-- shortMsgAlertContext-v1 map-ac shortMsgAlert (23) version1 (1)
-- mwdMngtContext-v1 map-ac mwdMngt (24) version1 (1)
-- shortMsgMT-RelayContext-v2 map-ac shortMsgMT-Relay (25) version2 (2)

14.4 MAP Dialogue Information

MAP-DialogueInformation {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-DialogueInformation (3) version3 (3)}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS  
map-DialogueAS,  
MAP-DialoguePDU

;
IMPORTS
gsm-NetworkId,
as-Id
FROM MobileDomainDefinitions {
ccitt (0) identified-organization (4) etsi (0) mobileDomain (0)
mobileDomainDefinitions (0) version1 (1)}

AddressString
FROM MAP-CommonDataTypes {
ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}

ExtensionContainer
FROM MAP-ExtensionDataTypes {
ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}

-- abstract syntax name for MAP-DialoguePDU

map-DialogueAS OBJECT IDENTIFIER ::= {
gsm-NetworkId as-Id map-DialoguePDU (1) version1 (1)}

MAP-DialoguePDU ::= CHOICE {
  map-open [0] MAP-OpenInfo,
  map-accept [1] MAP-AcceptInfo,
  map-close [2] MAP-CloseInfo,
  map-refuse [3] MAP-RefuseInfo,
  map-userAbort [4] MAP-UserAbortInfo,
  map-providerAbort [5] MAP-ProviderAbortInfo}

MAP-OpenInfo ::= SEQUENCE {
  destinationReference [0] AddressString OPTIONAL,
  originationReference [1] AddressString OPTIONAL,
  extensionContainer ExtensionContainer OPTIONAL
  -- extensionContainer must not be used in version 2
}

MAP-AcceptInfo ::= SEQUENCE {
  extensionContainer ExtensionContainer OPTIONAL
  -- extensionContainer must not be used in version 2
}

MAP-CloseInfo ::= SEQUENCE {
  extensionContainer ExtensionContainer OPTIONAL
  -- extensionContainer must not be used in version 2
}

MAP-RefuseInfo ::= SEQUENCE {
  reason Reason,
  extensionContainer ExtensionContainer OPTIONAL
  -- extensionContainer must not be used in version 2
}

Reason ::= ENUMERATED {
  noReasonGiven (0),
  invalidDestinationReference (1),
  invalidOriginatingReference (2)}

MAP-UserAbortInfo ::= SEQUENCE {
  map-UserAbortChoice MAP-UserAbortChoice,
  extensionContainer ExtensionContainer OPTIONAL
  -- extensionContainer must not be used in version 2
}

MAP-UserAbortChoice ::= CHOICE {
  userSpecificReason [0] NULL,
  userResourceLimitation [1] NULL,
  resourceUnavailable [2] ResourceUnavailableReason,
  applicationProcedureCancellation [3] ProcedureCancellationReason}
ResourceUnavailableReason ::= ENUMERATED {
  shortTermResourceLimitation (0),
  longTermResourceLimitation (1)}

ProcedureCancellationReason ::= ENUMERATED {
  handoverCancellation (0),
  radioChannelRelease (1),
  networkPathRelease (2),
  callRelease (3),
  associatedProcedureFailure (4),
  tandemDialogueRelease (5),
  remoteOperationsFailure (6)}

MAP-ProviderAbortInfo ::= SEQUENCE {
  map-ProviderAbortReason MAP-ProviderAbortReason,
  ...,
  extensionContainer ExtensionContainer OPTIONAL
     -- extensionContainer must not be used in version 2
}

MAP-ProviderAbortReason ::= ENUMERATED {
  abnormalDialogue (0),
  invalidPDU (1)}

END
14.5 MAP operation and error codes

MAP-Protocol
{ ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-Protocol (4) version3 (3)}

DEFINITIONS
BEGIN

IMPORTS
UpdateLocation, CancelLocation, PurgeMS, SendIdentification, PrepareHandover, SendEndSignal, ProcessAccessSignalling, ForwardAccessSignalling, PrepareSubsequentHandover, SendAuthenticationInfo, CheckIMEI, InsertSubscriberData, DeleteSubscriberData, Reset, ForwardCheckSS-Indication, RestoreData, ProvideSubscriberInfo, AnyTimeInterrogation

FROM MAP-MobileServiceOperations {
ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-MobileServiceOperations (5) version3 (3)}

ActivateTraceMode, DeactivateTraceMode, SendMSI
FROM MAP-OperationAndMaintenanceOperations {
ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-OperationAndMaintenanceOperations (6) version3 (3)}

SendRoutingInfo, ProvideRoamingNumber, ResumeCallHandling
FROM MAP-CallHandlingOperations {
ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-CallHandlingOperations (7) version3 (3)}

RegisterSS, EraseSS, ActivateSS, DeactivateSS, InterrogateSS, ProcessUnstructuredSS-Request, UnstructuredSS-Request, UnstructuredSS-Notify, RegisterPassword, GetPassword
FROM MAP-SupplementaryServiceOperations {
ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-SupplementaryServiceOperations (8) version3 (3)}

SendRoutingInfoForSM, MO-ForwardSM, MT-ForwardSM, ReportSM-DeliveryStatus, AlertServiceCentre, InformServiceCentre, ReadyForSM
FROM MAP-ShortMessageServiceOperations {
ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-ShortMessageServiceOperations (9)}
77 version3 (3))
78 SystemFailure,
79 DataMissing,
80 UnexpectedDataValue,
81 FacilityNotSupported,
82 UnknownSubscriber,
83 NumberChanged,
84 UnknownMSC,
85 UnidentifiedSubscriber,
86 UnknownEquipment,
87 RoamingNotAllowed,
88 IllegalSubscriber,
89 IllegalEquipment,
90 BearerServiceNotProvisioned,
91 TeleserviceNotProvisioned,
92 NoHandoverNumberAvailable,
93 SubsequentHandoverFailure,
94 TracingBufferFull,
95 OR-NotAllowed,
96 NoRoamingNumberAvailable,
97 AbsentSubscriber,
98 BusySubscriber,
99 NoSubscriberReply,
100 CallBarred,
101 ForwardingViolation,
102 ForwardingFailed,
103 CUG-Reject,
104 ATI-NotAllowed,
105 IllegalSS-Operation,
106 SS-ErrorStatus,
107 SS-NotAvailable,
108 SS-Incompatibility,
109 UnknownAlphabet,
110 USSD-Busy,
111 PW-RegistrationFailure,
112 NegativePW-Check,
113 NumberOfPW-AttemptsViolation,
114 SubscriberBusyForMT-SMS,
115 SM-DeliveryFailure,
116 MessageWaitingListFull,
117 AbsentSubscriberSM
FROM MAP-Errors {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-Errors (10) version3 (3)}
;

-- location registration operation codes

updateLocation UpdateLocation ::= localValue 2
cancelLocation CancelLocation ::= localValue 3
purgeMS PurgeMS ::= localValue 67
sendIdentification SendIdentification ::= localValue 55

-- handover operation codes

prepareHandover PrepareHandover ::= localValue 68
sendEndSignal SendEndSignal ::= localValue 29
forwardAccessSignalling ForwardAccessSignalling ::= localValue 33
prepareSubsequentHandover PrepareSubsequentHandover ::= localValue 69

-- authentication operation codes

sendAuthenticationInfo SendAuthenticationInfo ::= localValue 56

-- IMEI MANAGEMENT operation codes

checkIMEI CheckIMEI ::= localValue 43

-- subscriber management operation codes
insertSubscriberData  InsertSubscriberData ::= localValue 7
deleteSubscriberData  DeleteSubscriberData ::= localValue 8

-- fault recovery operation codes
reset  Reset ::= localValue 37
forwardCheckSS-Indication  ForwardCheckSS-Indication ::= localValue 38
restoreData  RestoreData ::= localValue 57

-- operation and maintenance operation codes
activateTraceMode  ActivateTraceMode ::= localValue 50
deactivateTraceMode  DeactivateTraceMode ::= localValue 51
sendIMSI  SendIMSI ::= localValue 58

-- call handling operation codes
sendRoutingInfo  SendRoutingInfo ::= localValue 22
provideRoamingNumber  ProvideRoamingNumber ::= localValue 4
resumeCallHandling  ResumeCallHandling ::= localValue 6

-- supplementary service handling operation codes
registerSS  RegisterSS ::= localValue 10
eraseSS  EraseSS ::= localValue 11
activateSS  ActivateSS ::= localValue 12
deactivateSS  DeactivateSS ::= localValue 13
interrogateSS  InterrogateSS ::= localValue 14
processUnstructuredSS-Request  ProcessUnstructuredSS-Request ::= localValue 59
unstructuredSS-Request  UnstructuredSS-Request ::= localValue 60
unstructuredSS-Notify  UnstructuredSS-Notify ::= localValue 61
registerPassword  RegisterPassword ::= localValue 17
getPassword  GetPassword ::= localValue 18

-- short message service operation codes
sendRoutingInfoForSM  SendRoutingInfoForSM ::= localValue 45
mo-forwardSM  MO-ForwardSM ::= localValue 46
mt-forwardSM  MT-ForwardSM ::= localValue 44
reportSM-DeliveryStatus  ReportSM-DeliveryStatus ::= localValue 47
informServiceCentre  InformServiceCentre ::= localValue 63
alertServiceCentre  AlertServiceCentre ::= localValue 64
readyForSM  ReadyForSM ::= localValue 66

-- provide subscriber info operation codes
provideSubscriberInfo  ProvideSubscriberInfo ::= localValue 70

-- any time interrogation operation codes
anyTimeInterrogation  AnyTimeInterrogation ::= localValue 71

-- generic error codes
systemFailure  SystemFailure ::= localValue 34
dataMissing  DataMissing ::= localValue 35
unexpectedDataValue  UnexpectedDataValue ::= localValue 36
facilityNotSupported  FacilityNotSupported ::= localValue 21

-- identification and numbering error codes
unknownSubscriber  UnknownSubscriber ::= localValue 1
numberChanged  NumberChanged ::= localValue 44
unknownMSC  UnknownMSC ::= localValue 3
unidentifiedSubscriber  UnidentifiedSubscriber ::= localValue 5
unknownEquipment  UnknownEquipment ::= localValue 7

-- subscription error codes
roamingNotAllowed RoamingNotAllowed ::= localValue 8
illegalSubscriber IllegalSubscriber ::= localValue 9
illegalEquipment IllegalEquipment ::= localValue 12
bearerServiceNotProvisioned BearerServiceNotProvisioned ::= localValue 10
teleServiceNotProvisioned TeleServiceNotProvisioned ::= localValue 11

-- handover error codes

noHandoverNumberAvailable NoHandoverNumberAvailable ::= localValue 25
subsequentHandoverFailure SubsequentHandoverFailure ::= localValue 26

-- operation and maintenance error codes

tracingBufferFull TracingBufferFull ::= localValue 40

-- call handling error codes

noRoamingNumberAvailable NoRoamingNumberAvailable ::= localValue 39
absentSubscriber AbsentSubscriber ::= localValue 27
busySubscriber BusySubscriber ::= localValue 45
noSubscriberReply NoSubscriberReply ::= localValue 46
callBarred CallBarred ::= localValue 13
forwardingFailed ForwardingFailed ::= localValue 47
or-NotAllowed OR-NotAllowed ::= localValue 48
forwardingViolation ForwardingViolation ::= localValue 14
cug-Reject CUG-Reject ::= localValue 15

-- any time interrogation error codes

ati-NotAllowed ATI-NotAllowed ::= localValue 49

-- supplementary service error codes

illegalSS-Operation IllegalSS-Operation ::= localValue 16
ss-ErrorStatus SS-ErrorStatus ::= localValue 17
ss-NotAvailable SS-NotAvailable ::= localValue 18
ss-SubscriptionViolation SS-SubscriptionViolation ::= localValue 19
ss-Incompatibility SS-Incompatibility ::= localValue 20
unknownAlphabet UnknownAlphabet ::= localValue 71
ussd-Busy USSD-Busy ::= localValue 72
negativePW-Check NegativePW-Check ::= localValue 38
numberOFPW-AttemptsViolation NumberOfPW-AttemptsViolation ::= localValue 43

-- short message service error codes

subscriberBusyForMT-SMS SubscriberBusyForMT-SMS ::= localValue 31
sm-DeliveryFailure SM-DeliveryFailure ::= localValue 32
messageWaitingListFull MessageWaitingListFull ::= localValue 33

-- The following operation codes are reserved for operations
-- existing in previous versions of the protocol

-- Operation Name AC used Oper. Code
-- sendParameters map-ac infoRetrieval (14) version1 (1) localValue 9
-- performHandover map-ac networkFunctionalSs (18) version1 (1) localValue 19
-- performSubsequentHandover map-ac handoverControl (11) version1 (1) localValue 28
-- noteInternalHandover map-ac handoverControl (11) version1 (1) localValue 30
-- noteSubscriberPresent map-ac mwdMngt (24) version1 (1) localValue 48
-- alertServiceCentreWithoutResult map-ac shortMsgAlert (23) version1 (1) localValue 49
-- traceSubscriberActivity map-ac handoverControl (11) version1 (1) localValue 52
-- beginSubscriberActivity map-ac networkFunctionalSs (18) version1 (1) localValue 54

-- The following error codes are reserved for errors

ETSI
-- existing in previous versions of the protocol

<table>
<thead>
<tr>
<th>Error Name</th>
<th>AC used</th>
<th>Error Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- unknownBaseStation</td>
<td>map-ac handoverControl (11) version1 (1)</td>
<td>localValue 2</td>
</tr>
<tr>
<td>-- invalidTargetBaseStation</td>
<td>map-ac handoverControl (11) version1 (1)</td>
<td>localValue 23</td>
</tr>
<tr>
<td>-- noRadioResourceAvailable</td>
<td>map-ac handoverControl (11) version1 (1)</td>
<td>localValue 24</td>
</tr>
</tbody>
</table>

END
14.6  MAP operation and error types

14.6.1  Mobile Service Operations

MAP-MobileServiceOperations {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-MobileServiceOperations (5)
version3 (3)
}

DEFINITIONS ::=
BEGIN
EXPORTS
-- location registration operations
UpdateLocation,
CancelLocation,
PurgeMS,
SendIdentification,
-- subscriber information enquiry operations
ProvideSubscriberInfo,
-- any time information enquiry operations
AnyTimeInterrogation,
-- handover operations
PrepareHandover,
SendEndSignal,
ProcessAccessSignalling,
ForwardAccessSignalling,
PrepareSubsequentHandover,
-- authentication management operations
SendAuthenticationInfo,
-- IMEI management operations
CheckIMEI,
-- subscriber management operations
InsertSubscriberData,
DeleteSubscriberData,
-- fault recovery operations
Reset,
ForwardCheckSS-Indication,
RestoreData
;
IMPORTS
OPERATION
FROM TCAPMessages {
  ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
SystemFailure,
DataMissing,
UnexpectedDataValue,
UnknownSubscriber,
UnknownMSC,
UnidentifiedSubscriber,
UnknownEquipment,
RoamingNotAllowed,
ATI-NotAllowed,
NoHandoverNumberAvailable,
SubsequentHandoverFailure
FROM MAP-Errors {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-Errors (10) version3 (3)}
UpdateLocationArg,
UpdateLocationRes,
CancelLocationArg,
PurgeMS-Arg,
SendIdentificationRes,
PrepareHO-Arg,
PrepareHO-Res,
PrepareSubsequentHO-Arg,
SendAuthenticationInfoArg,
SendAuthenticationInfoRes,
EquipmentStatus,
InsertSubscriberDataArg,
InsertSubscriberDataRes,
DeleteSubscriberDataArg,
DeleteSubscriberDataRes,
ResetArg,
RestoreDataArg,
RestoreDataRes,
ProvideSubscriberInfoArg,
ProvideSubscriberInfoRes,
AnyTimeInterrogationArg,
AnyTimeInterrogationRes
FROM MAP-MS-DataTypes {
ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-MS-DataTypes (11) version3 (3)}
ExternalSignalInfo,
TMSI,
IMEI
FROM MAP-CommonDataTypes {
ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}
-- location registration operations

UpdateLocation ::= OPERATION --Timer m
ARGUMENT
updateLocationArg UpdateLocationArg
RESULT
updateLocationRes UpdateLocationRes
ERRORS {
SystemFailure,
DataMissing,
UnexpectedDataValue,
UnknownSubscriber,
RoamingNotAllowed"

CancelLocation ::= OPERATION --Timer m
ARGUMENT
cancelLocationArg CancelLocationArg
RESULT
ERRORS {
DataMissing,
UnexpectedDataValue"

PurgeMS ::= OPERATION --Timer m
ARGUMENT
purgeMS-Arg PurgeMS-Arg
RESULT

SendIdentification ::= OPERATION --Timer s
ARGUMENT
tmsi TMSI
RESULT
sendIdentificationRes SendIdentificationRes
ERRORS {
DataMissing,
UnidentifiedSubscriber"

-- subscriber information enquiry operations

ProvideSubscriberInfo ::= OPERATION --Timer m
ARGUMENT
provideSubscriberInfoArg ProvideSubscriberInfoArg
RESULT
provideSubscriberInfoRes ProvideSubscriberInfoRes
ERRORS {
DataMissing,
UnexpectedDataValue"

-- any time information enquiry operations
AnyTimeInterrogation ::= OPERATION  --Timer m
  ARGUMENT
  anyTimeInterrogationArg AnyTimeInterrogationArg
  RESULT
  anyTimeInterrogationRes AnyTimeInterrogationRes
  ERRORS {
    SystemFailure,
    ATI-NotAllowed,
    DataMissing,
    UnexpectedDataValue,
    UnknownSubscriber}

-- handover operations

PrepareHandover ::= OPERATION  --Timer m
  ARGUMENT
  prepareHO-Arg PrepareHO-Arg
  RESULT
  prepareHO-Res PrepareHO-Res
  ERRORS {
    SystemFailure,
    DataMissing,
    UnexpectedDataValue,
    NoHandoverNumberAvailable}

SendEndSignal ::= OPERATION  --Timer l
  ARGUMENT
  bss-APDU ExternalSignalInfo
  RESULT

ProcessAccessSignalling ::= OPERATION  --Timer s
  ARGUMENT
  bss-APDU ExternalSignalInfo

ForwardAccessSignalling ::= OPERATION  --Timer s
  ARGUMENT
  bss-APDU ExternalSignalInfo

PrepareSubsequentHandover ::= OPERATION  --Timer m
  ARGUMENT
  prepareSubsequentHO-Arg PrepareSubsequentHO-Arg
  RESULT
  bss-APDU ExternalSignalInfo
  ERRORS {
    UnexpectedDataValue,
    DataMissing,
    UnknownMSC,
    SubsequentHandoverFailure}

-- authentication management operations

SendAuthenticationInfo ::= OPERATION  --Timer m
  ARGUMENT
  sendAuthenticationInfoArg SendAuthenticationInfoArg
  RESULT
  sendAuthenticationInfoRes SendAuthenticationInfoRes
  ERRORS {
    SystemFailure,
    DataMissing,
    UnexpectedDataValue,
    UnknownSubscriber}

-- IMEI management operations

CheckIMEI ::= OPERATION  --Timer m
  ARGUMENT
  imei IMEI
  RESULT
  equipmentStatus EquipmentStatus
  ERRORS {
    SystemFailure,
    DataMissing,
    UnexpectedDataValue,
    UnknownEquipment}

-- subscriber management operations
InsertSubscriberData ::= OPERATION     --Timer m
ARGUMENT
insertSubscriberDataArg   InsertSubscriberDataArg
RESULT
insertSubscriberDataRes   InsertSubscriberDataRes
-- optional
ERRORS {
  DataMissing,
  UnexpectedDataValue,
  UnidentifiedSubscriber}

DeleteSubscriberData ::= OPERATION     --Timer m
ARGUMENT
deleteSubscriberDataArg   DeleteSubscriberDataArg
RESULT
deleteSubscriberDataRes   DeleteSubscriberDataRes
-- optional
ERRORS {
  DataMissing,
  UnexpectedDataValue,
  UnidentifiedSubscriber}

-- fault recovery operations

Reset ::= OPERATION     --Timer m
ARGUMENT
resetArg     ResetArg

ForwardCheckSS-Indication ::= OPERATION     --Timer s

RestoreData ::= OPERATION     --Timer m
ARGUMENT
restoreDataArg   RestoreDataArg
RESULT
restoreDataRes   RestoreDataRes
ERRORS {
  SystemFailure,
  DataMissing,
  UnexpectedDataValue,
  UnknownSubscriber}
14.6.2 Operation and Maintenance Operations

**MAP-OperationAndMaintenanceOperations**

```
correct character
```

**DEFINITIONS**

```
BEGIN

EXPORTS
ActivateTraceMode,
DeactivateTraceMode,
SendIMSI;

IMPORTS
OPERATION FROM TCAPMessages {
  ccitt recommendation q 773 modules (2) messages (1) version2 (2)}

FROM MAP-Errors {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-Errors (10) version3 (3)}

FROM MAP-OM-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-OM-DataTypes (12) version3 (3)}

FROM MAP-CommonDataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}

ActivateTraceModeArg, ActivateTraceModeRes,
DeactivateTraceModeArg, DeactivateTraceModeRes

ISDN-AddressString, IMSI

ActivateTraceModeArg, ActivateTraceModeRes

activateTraceModeArg ActivateTraceModeArg
activateTraceModeRes ActivateTraceModeRes

ERRORS {
  SystemFailure,
  DataMissing,
  UnexpectedDataValue,
  FacilityNotSupported,
  UnidentifiedSubscriber,
  TracingBufferFull}

DeactivateTraceModeArg, DeactivateTraceModeRes

deactivateTraceModeArg DeactivateTraceModeArg
deactivateTraceModeRes DeactivateTraceModeRes

ERRORS {
  SystemFailure,
  DataMissing,
  UnexpectedDataValue,
  FacilityNotSupported,
  UnidentifiedSubscriber
```

**ActivateTraceMode** ::= OPERATION --Timer m

ARGUMENT
activateTraceModeArg ActivateTraceModeArg
RESULT
activateTraceModeRes ActivateTraceModeRes

-- optional

ERRORS {
  SystemFailure,
  DataMissing,
  UnexpectedDataValue,
  FacilityNotSupported,
  UnidentifiedSubscriber,
  TracingBufferFull}

**DeactivateTraceMode** ::= OPERATION --Timer m

ARGUMENT
deactivateTraceModeArg DeactivateTraceModeArg
RESULT
deactivateTraceModeRes DeactivateTraceModeRes

-- optional

ERRORS {
  SystemFailure,
  DataMissing,
  UnexpectedDataValue,
  FacilityNotSupported,
  UnidentifiedSubscriber}
SendIMSI ::= OPERATION
   --Timer m
   ARGUMENT
     msisdn ISDN-AddressString
   RESULT
     imsi IMSI
   ERRORS {
     DataMissing,
     UnexpectedDataValue,
     UnknownSubscriber}
END
14.6.3 Call Handling Operations

MAP-CallHandlingOperations {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-CallHandlingOperations (7)
  version3 (3)
}

DEFINITIONS ::= BEGIN

EXPORTS
  SendRoutingInfo,
  ProvideRoamingNumber,
  ResumeCallHandling;

IMPORTS
  OPERATION FROM TCAPMessages {
    ccitt recommendation q 773 modules (2) messages (1) version2 (2)}

  SystemFailure,
  DataMissing,
  UnexpectedDataValue,
  FacilityNotSupported,
  OR-NotAllowed,
  UnknownSubscriber,
  NumberChanged,
  BearerServiceNotProvisioned,
  TeleserviceNotProvisioned,
  NoRoamingNumberAvailable,
  AbsentSubscriber,
  BusySubscriber,
  NoSubscriberReply,
  CallBarred,
  ForwardingViolation,
  ForwardingFailed,
  CUG-Reject

FROM MAP-Errors {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-Errors (10) version3 (3)}

  SendRoutingInfoArg,
  SendRoutingInfoRes,
  ProvideRoamingNumberArg,
  ProvideRoamingNumberRes,
  ResumeCallHandlingArg,
  ResumeCallHandlingRes

FROM MAP-CH-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-CH-DataTypes (13) version3 (3)}

SendRoutingInfo ::= OPERATION --Timer m
  ARGUMENT
    sendRoutingInfoArg SendRoutingInfoArg
  RESULT
    sendRoutingInfoRes SendRoutingInfoRes
  ERRORS
    SystemFailure,
    DataMissing,
    UnexpectedDataValue,
    FacilityNotSupported,
    OR-NotAllowed,
    UnknownSubscriber,
    NumberChanged,
    BearerServiceNotProvisioned,
    TeleserviceNotProvisioned,
    AbsentSubscriber,
    BusySubscriber,
    NoSubscriberReply,
    CallBarred,
    CUG-Reject,
    ForwardingViolation}
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>77</td>
<td><strong>ProvideRoamingNumber</strong> ::= OPERATION --Timer m</td>
</tr>
<tr>
<td>78</td>
<td>ARGUMENT</td>
</tr>
<tr>
<td>79</td>
<td>provideRoamingNumberArg ProvideRoamingNumberArg</td>
</tr>
<tr>
<td>80</td>
<td>RESULT</td>
</tr>
<tr>
<td>81</td>
<td>provideRoamingNumberRes ProvideRoamingNumberRes</td>
</tr>
<tr>
<td>82</td>
<td>ERRORS {</td>
</tr>
<tr>
<td>83</td>
<td>SystemFailure,</td>
</tr>
<tr>
<td>84</td>
<td>DataMissing,</td>
</tr>
<tr>
<td>85</td>
<td>UnexpectedDataValue,</td>
</tr>
<tr>
<td>86</td>
<td>FacilityNotSupported,</td>
</tr>
<tr>
<td>87</td>
<td>OR-NotAllowed,</td>
</tr>
<tr>
<td>88</td>
<td>AbsentSubscriber,</td>
</tr>
<tr>
<td>89</td>
<td>NoRoamingNumberAvailable}</td>
</tr>
<tr>
<td>90</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td><strong>ResumeCallHandling</strong> ::= OPERATION --Timer m</td>
</tr>
<tr>
<td>92</td>
<td>ARGUMENT</td>
</tr>
<tr>
<td>93</td>
<td>resumeCallHandlingArg ResumeCallHandlingArg</td>
</tr>
<tr>
<td>94</td>
<td>RESULT</td>
</tr>
<tr>
<td>95</td>
<td>resumeCallHandlingRes ResumeCallHandlingRes</td>
</tr>
<tr>
<td>96</td>
<td>ERRORS {</td>
</tr>
<tr>
<td>97</td>
<td>ForwardingFailed,</td>
</tr>
<tr>
<td>98</td>
<td>OR-NotAllowed,</td>
</tr>
<tr>
<td>99</td>
<td>UnexpectedDataValue}</td>
</tr>
<tr>
<td>100</td>
<td>END</td>
</tr>
</tbody>
</table>
14.6.4 Supplementary service operations

```
MAP-SupplementaryServiceOperations {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-SupplementaryServiceOperations (8)
version3 (3)}

DEFINITIONS
BEGIN
EXPORTS
  RegisterSS,
  EraseSS,
  ActivateSS,
  DeactivateSS,
  InterrogateSS,
  ProcessUnstructuredSS-Request,
  UnstructuredSS-Request,
  UnstructuredSS-Notify,
  RegisterPassword,
  GetPassword;
IMPORTS
OPERATION
FROM TCAPMessages {
  ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
SystemFailure,
DataMissing,
UnexpectedDataValue,
BearerServiceNotProvisioned,
TeleserviceNotProvisioned,
CallBarred,
IllegalSS-Operation,
SS-ErrorStatus,
SS-NotAvailable,
SS-SubscriptionViolation,
SS-Incompatibility,
PW-RegistrationFailure,
NegativePW-Check,
NumberOfPW-AttemptsViolation,
UnknownPW-Check,
IllegalPassword,
IllegalSubscriber;
FROM MAP-Errors {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-Errors (10) version3 (3)}
RegisterSS-Arg,
SS-Info,
SS-ForBS-Code,
InterrogateSS-Res,
USSD-Arg,
USSD-Res,
Password,
GuidanceInfo;
FROM MAP-SS-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-SS-DataTypes (14) version3 (3)}
SS-Code
FROM MAP-SS-Code {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-SS-Code (15) version3 (3)};
-- supplementary service handling operations
```
### RegistersSS

- **Operation**: Timer m
- **Argument**: registerSS-Arg
- **Result**: ss-Info
- **Errors**:
  - SystemFailure
  - DataMissing
  - UnexpectedDataValue
  - BearerServiceNotProvisioned
  - TeleserviceNotProvisioned
  - CallBarred
  - IllegalSS-Operation
  - SS-ErrorStatus
  - SS-Incompatibility

### EraseSS

- **Operation**: Timer m
- **Argument**: ss-ForBS
- **Result**: ss-Info
- **Errors**:
  - SystemFailure
  - DataMissing
  - UnexpectedDataValue
  - BearerServiceNotProvisioned
  - TeleserviceNotProvisioned
  - CallBarred
  - IllegalSS-Operation
  - SS-ErrorStatus

### ActivateSS

- **Operation**: Timer m
- **Argument**: ss-ForBS
- **Result**: ss-Info
- **Errors**:
  - SystemFailure
  - DataMissing
  - UnexpectedDataValue
  - BearerServiceNotProvisioned
  - TeleserviceNotProvisioned
  - CallBarred
  - IllegalSS-Operation
  - SS-ErrorStatus
  - SS-SubscriptionViolation
  - SS-Incompatibility
  - NegativePW-Check
  - NumberOfPW-AttemptsViolation

### DeactivateSS

- **Operation**: Timer m
- **Argument**: ss-ForBS
- **Result**: ss-Info
- **Errors**:
  - SystemFailure
  - DataMissing
  - UnexpectedDataValue
  - BearerServiceNotProvisioned
  - TeleserviceNotProvisioned
  - CallBarred
  - IllegalSS-Operation
  - SS-ErrorStatus
  - SS-SubscriptionViolation
  - NegativePW-Check
  - NumberOfPW-AttemptsViolation
InterrogateSS ::= OPERATION          --Timer m
   ARGUMENT
   ss-ForBS           SS-ForBS-Code
   RESULT
   InterrogateSS-Res InterrogateSS-Res
   ERRORS (  
     SystemFailure,  
     DataMissing,     
     UnexpectedDataValue,  
     TeleserviceNotProvisioned,  
     CallBarred,  
     IllegalSS-Operation,  
     SS-NotAvailable) 

ProcessUnstructuredSS-Request ::= OPERATION       --Timer 10 minutes
   ARGUMENT
   ussd-Arg           USSD-Arg
   RESULT
   USSD-Res
   ERRORS (  
     SystemFailure,  
     DataMissing,     
     UnexpectedDataValue,  
     UnknownAlphabet,     
     CallBarred) 

UnstructuredSS-Request ::= OPERATION      --Timer ml
   ARGUMENT
   ussd-Arg           USSD-Arg
   RESULT
   USSD-Res
   ERRORS (  
     SystemFailure,  
     DataMissing,     
     UnexpectedDataValue,  
     AbsentSubscriber,  
     IllegalSubscriber,  
     IllegalEquipment,  
     UnknownAlphabet,  
     USSD-Busy) 

UnstructuredSS-Notify ::= OPERATION       --Timer ml
   ARGUMENT
   ussd-Arg           USSD-Arg
   RESULT
   ERRORS (  
     SystemFailure,  
     DataMissing,     
     UnexpectedDataValue,  
     AbsentSubscriber,  
     IllegalSubscriber,  
     IllegalEquipment,  
     UnknownAlphabet,  
     USSD-Busy) 

RegisterPassword ::= OPERATION          --Timer ml
   ARGUMENT
   ss-Code            SS-Code
   RESULT
   newPassword        Password
   ERRORS (  
     SystemFailure,  
     DataMissing,     
     UnexpectedDataValue,  
     CallBarred,  
     SS-SubscriptionViolation,  
     PW-RegistrationFailure,  
     NegativePW-Check,  
     NumberOfPWAttemptsViolation)  
   LINKED (  
     GetPassword)
GetPassword ::= OPERATION --Timer m
ARGUMENT
  guidanceInfo GuidanceInfo
RESULT
  currentPassword Password
END
14.6.5 Short message service operations

```plaintext
MAP-ShortMessageServiceOperations {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-ShortMessageServiceOperations (9)
version3 (3)}

DEFINITIONS
::=
BEGIN
EXPORTS
SendRoutingInfoForSM,
MO-ForwardSM,
MT-ForwardSM,
ReportSM-DeliveryStatus,
AlertServiceCentre,
InformServiceCentre,
ReadyForSM;

IMPORTS
OPERATION
FROM TCAPMessages {
ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
SystemFailure,
DataMissing,
UnexpectedDataValue,
FacilityNotSupported,
UnknownSubscriber,
UnidentifiedSubscriber,
IllegalSubscriber,
IllegalEquipment,
TeleserviceNotProvisioned,
AbsentSubscriber,
CallBarred,
SubscriberBusyForMT-SMS,
SM-DeliveryFailure,
MessageWaitingListFull,
AbsentSubscriberSM
FROM MAP-Errors {
ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-Errors (10) version3 (3)}
RoutingInfoForSM-Arg,
RoutingInfoForSM-Res,
MO-ForwardSM-Arg,
MO-ForwardSM-Res,
MT-ForwardSM-Arg,
MT-ForwardSM-Res,
ReportSM-DeliveryStatusArg,
ReportSM-DeliveryStatusRes,
AlertServiceCentreArg,
InformServiceCentreArg,
ReadyForSM-Arg
FROM MAP-SM-DataTypes {
ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-SM-DataTypes (16) version3 (3)}
;
```
SendRoutingInfoForSM ::= OPERATION --Timer m
   ARGUMENT
   routingInfoForSM-Arg RoutingInfoForSM-Arg
   RESULT
   routingInfoForSM-Res RoutingInfoForSM-Res
   ERRORS {SystemFailure, DataMissing, UnexpectedDataValue, FacilityNotSupported, UnknownSubscriber, TeleserviceNotProvisioned, CallBarred, AbsentSubscriberSM}

MO-ForwardSM ::= OPERATION --Timer ml
   ARGUMENT
   mo-forwardSM-Arg MO-ForwardSM-Arg
   RESULT
   mo-forwardSM-Res MO-ForwardSM-Res
   -- optional
   ERRORS {SystemFailure, UnexpectedDataValue, FacilityNotSupported, SM-DeliveryFailure}

MT-ForwardSM ::= OPERATION --Timer ml
   ARGUMENT
   mt-forwardSM-Arg MT-ForwardSM-Arg
   RESULT
   mt-forwardSM-Res MT-ForwardSM-Res
   -- optional
   ERRORS {SystemFailure, DataMissing, UnexpectedDataValue, FacilityNotSupported, UnidentifiedSubscriber, IllegalSubscriber, IllegalEquipment, SubscriberBusyForMT-SMS, SM-DeliveryFailure, AbsentSubscriberSM}

ReportSM-DeliveryStatus ::= OPERATION --Timer s
   ARGUMENT
   reportSM-DeliveryStatusArg ReportSM-DeliveryStatusArg
   RESULT
   reportSM-DeliveryStatusRes ReportSM-DeliveryStatusRes
   -- optional
   ERRORS {DataMissing, UnexpectedDataValue, UnknownSubscriber, MessageWaitingListFull}

AlertServiceCentre ::= OPERATION --Timer s
   ARGUMENT
   alertServiceCentreArg AlertServiceCentreArg
   RESULT
   ERRORS {SystemFailure, DataMissing, UnexpectedDataValue}

InformServiceCentre ::= OPERATION --Timer s
   ARGUMENT
   informServiceCentreArg InformServiceCentreArg
ReadyForSM ::= OPERATION

ARGUMENT

readyForSM-Arg

RESULT

ERRORS {

DataMissing,
UnexpectedDataValue,
FacilityNotSupported,
UnknownSubscriber}

END
14.6.6 Errors

MAP-Errors {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-Errors (10) version3 (3)}
DEFINITIONS ::=
BEGIN
EXPORTS

-- generic errors
SystemFailure,
DataMissing,
UnexpectedDataValue,
FacilityNotSupported,

-- identification and numbering errors
UnknownSubscriber,
NumberChanged,
UnknownMSC,
UnidentifiedSubscriber,
UnknownEquipment,

-- subscription errors
RoamingNotAllowed,
IllegalSubscriber,
IllegalEquipment,
BearerServiceNotProvisioned,
TeleserviceNotProvisioned,

-- handover errors
NoHandoverNumberAvailable,
SubsequentHandoverFailure,

-- operation and maintenance errors
TracingBufferFull,

-- call handling errors
OR-NotAllowed,
NoRoamingNumberAvailable,
BusySubscriber,
NoSubscriberReply,
AbsentSubscriber,
CallBarred,
ForwardingViolation,
ForwardingFailed,
CUG-Reject,

-- any time interrogation errors
ATI-NotAllowed,

-- supplementary service errors
IllegalSS-Operation,
SS-ErrorStatus,
SS-NotAvailable,
SS-SubscriptionViolation,
SS-Incompatibility,
UnknownAlphabet,
USSD-Busy,
PW-RegistrationFailure,
NegativePW-Check,
NumberOfPW-AttemptsViolation,

-- short message service errors
SubscriberBusyForMT-SMS,
SM-DeliveryFailure,
MessageWaitingListFull,
AbsentSubscriberSM
;
IMPORTS ERROR FROM TCAPMessages {
  ccitt recommendation q 773 modules (2) messages (1) version2 (2)}
FROM MAP-SS-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-SS-DataTypes (14) version3 (3)}

SS-IncompatibilityCause,
PW-RegistrationFailureCause,
SM-DeliveryFailureCause,
SystemFailureParam,
DataMissingParam,
UnexpectedDataParam,
FacilityNotSupParam,
UnknownSubscriberParam,
NumberChangedParam,
UnidentifiedSubParam,
RoamingNotAllowedParam,
IllegalSubscriberParam,
IllegalEquipmentParam,
BearerServNotProvParam,
TeleservNotProvParam,
TracingBufferFullParam,
NoRoamingNbParam,
OR-NotAllowedParam,
AbsentSubscriberParam,
BusySubscriberParam,
NoSubscriberReplyParam,
CallBarredParam,
ForwardingViolationParam,
ForwardingFailedParam,
CUG-RejectParam,
ATI-NotAllowedParam,
SubBusyForMT-SMS-Param,
MessageWaitListFullParam,
AbsentSubscriberSM-Param

FROM MAP-ER-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-ER-DataTypes (17) version3 (3)}

-- generic errors

SystemFailure ::= ERROR
PARAMETER
  systemFailureParam SystemFailureParam
  -- optional

DataMissing ::= ERROR
PARAMETER
  dataMissingParam DataMissingParam
  -- optional
  -- dataMissingParam must not be used in version <3

UnexpectedDataValue ::= ERROR
PARAMETER
  unexpectedDataParam UnexpectedDataParam
  -- optional
  -- unexpectedDataParam must not be used in version <3

FacilityNotSupported ::= ERROR
PARAMETER
  facilityNotSupParam FacilityNotSupParam
  -- optional
  -- facilityNotSupParam must not be used in version <3

-- identification and numbering errors

UnknownSubscriber ::= ERROR
PARAMETER
  unknownSubscriberParam UnknownSubscriberParam
  -- optional
  -- unknownSubscriberParam must not be used in version <3

NumberChanged ::= ERROR
PARAMETER
  numberChangedParam NumberChangedParam
  -- optional
UnknownMSC ::= ERROR

UnidentifiedSubscriber ::= ERROR
  PARAMETER
    unidentifiedSubParam UnidentifiedSubParam
    -- optional
    -- unidentifiedSubParam must not be used in version <3

UnknownEquipment ::= ERROR

-- subscription errors

RoamingNotAllowed ::= ERROR
  PARAMETER
    roamingNotAllowedParam RoamingNotAllowedParam

IllegalSubscriber ::= ERROR
  PARAMETER
    illegalSubscriberParam IllegalSubscriberParam
    -- optional
    -- illegalSubscriberParam must not be used in version <3

IllegalEquipment ::= ERROR
  PARAMETER
    illegalEquipmentParam IllegalEquipmentParam
    -- optional
    -- illegalEquipmentParam must not be used in version <3

BearerServiceNotProvisioned ::= ERROR
  PARAMETER
    bearerServNotProvParam BearerServNotProvParam
    -- optional
    -- bearerServNotProvParam must not be used in version <3

TeleserviceNotProvisioned ::= ERROR
  PARAMETER
    teleservNotProvParam TeleservNotProvParam
    -- optional
    -- teleservNotProvParam must not be used in version <3

NoHandoverNumberAvailable ::= ERROR

-- handover errors

SubsequentHandoverFailure ::= ERROR

-- operation and maintenance errors

TracingBufferFull ::= ERROR
  PARAMETER
    tracingBufferFullParam TracingBufferFullParam
    -- optional

NoRoamingNumberAvailable ::= ERROR
  PARAMETER
    noRoamingNbParam NoRoamingNbParam
    -- optional

AbsentSubscriber ::= ERROR
  PARAMETER
    absentSubscriberParam AbsentSubscriberParam
    -- optional
    -- absentSubscriberParam must not be used in version <3

BusySubscriber ::= ERROR
  PARAMETER
    busySubscriberParam BusySubscriberParam
    -- optional

NoSubscriberReply ::= ERROR
  PARAMETER
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Description</th>
<th>Parameter Name</th>
<th>Optional</th>
</tr>
</thead>
<tbody>
<tr>
<td>noSubscriberReplyParam</td>
<td>NoSubscriberReplyParam</td>
<td>CallBarred</td>
<td>ERROR</td>
</tr>
<tr>
<td>CallBarred : error</td>
<td>PARAMETER</td>
<td>callBarredParam</td>
<td>optional</td>
</tr>
<tr>
<td>ForwardingViolation</td>
<td>ForwardingViolation</td>
<td>ForwardingViolationParam</td>
<td>ERROR</td>
</tr>
<tr>
<td>ForwardingFailed</td>
<td>ForwardingFailed</td>
<td>ForwardingFailedParam</td>
<td>ERROR</td>
</tr>
<tr>
<td>CUG-Reject</td>
<td>CUG-Reject</td>
<td>CUG-RejectParam</td>
<td>ERROR</td>
</tr>
<tr>
<td>OR-NotAllowed</td>
<td>OR-NotAllowed</td>
<td>OR-NotAllowedParam</td>
<td>ERROR</td>
</tr>
<tr>
<td>ATI-NotAllowed</td>
<td>ATI-NotAllowed</td>
<td>ATI-NotAllowedParam</td>
<td>ERROR</td>
</tr>
<tr>
<td>IllegalSS-Operation</td>
<td>IllegalSS-Operation</td>
<td>IllegalSS-Operation</td>
<td>ERROR</td>
</tr>
<tr>
<td>SS-ErrorStatus</td>
<td>SS-ErrorStatus</td>
<td>SS-Status</td>
<td>ERROR</td>
</tr>
<tr>
<td>SS-NotAvailable</td>
<td>SS-NotAvailable</td>
<td>SS-NotAvailable</td>
<td>ERROR</td>
</tr>
<tr>
<td>SS-SubscriptionViolation</td>
<td>SS-SubscriptionViolation</td>
<td>SS-SubscriptionViolation</td>
<td>ERROR</td>
</tr>
<tr>
<td>SS-Incompatibility</td>
<td>SS-Incompatibility</td>
<td>SS-IncompatibilityCause</td>
<td>ERROR</td>
</tr>
<tr>
<td>UnknownAlphabet</td>
<td>UnknownAlphabet</td>
<td>UnknownAlphabet</td>
<td>ERROR</td>
</tr>
<tr>
<td>USSD-Busy</td>
<td>USSD-Busy</td>
<td>USSD-Busy</td>
<td>ERROR</td>
</tr>
<tr>
<td>PW-RegistrationFailure</td>
<td>PW-Registration Failure</td>
<td>PW-RegistrationFailureCause</td>
<td>ERROR</td>
</tr>
<tr>
<td>NegativePW-Check</td>
<td>NegativePW-Check</td>
<td>NegativePW-Check</td>
<td>ERROR</td>
</tr>
<tr>
<td>NumberOfPW-AttemptsViolation</td>
<td>NumberOfPW-AttemptsViolation</td>
<td>NumberOfPW-AttemptsViolation</td>
<td>ERROR</td>
</tr>
<tr>
<td>SubscriberBusyForMT-SMS</td>
<td>SubscriberBusyForMT-SMS</td>
<td>SubscriberBusyForMT-SMS-Param</td>
<td>ERROR</td>
</tr>
</tbody>
</table>

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---
SM-DeliveryFailure ::= ERROR
   PARAMETER
   sm-DeliveryFailureCause SM-DeliveryFailureCause

MessageWaitingListFull ::= ERROR
   PARAMETER
   messageWaitListFullParam MessageWaitListFullParam
   -- optional

AbsentSubscriberSM ::= ERROR
   PARAMETER
   absentSubscriberSM-Param AbsentSubscriberSM-Param
   -- optional

END
14.7 MAP constants and data types

14.7.1 Mobile Service data types

```plaintext
MAP-MS-DataTypes {  
ccti: identified-organization (4) etsi (0) mobileDomain (0)  
gsm-Network (1) modules (3) map-MS-DataTypes (11) version3 (3)}

DEFINITIONS
IMPLICIT TAGS
::=
BEGIN
EXPORTS

-- location registration types
UpdateLocationArg,  
UpdateLocationRes,  
CancelLocationArg,  
PurgeMS-Arg,  
SendIdentificationRes,

-- handover types
PrepareHO-Arg,  
PrepareHO-Res,  
PrepareSubsequentHO-Arg,

-- authentication management types
SendAuthenticationInfoArg,  
SendAuthenticationInfoRes,

-- security management types
EquipmentStatus,

-- subscriber management types
InsertSubscriberDataArg,  
InsertSubscriberDataRes,  
DeleteSubscriberDataArg,  
DeleteSubscriberDataRes,  
SubscriberData,  
ODB-Data,  
SubscriberStatus,  
ZoneCodeList,  
maxNumOfZoneCodes,  
O-CSI,  
ServiceKey,  
DefaultCallHandling,  
SupportedCamelPhases,  
maxNumOfCamelTDPData,  
CUG-Index,  
CUG-Interlock,  
InterCUG-Restrictions,  
IntraCUG-Options,

-- fault recovery types
ResetArg,  
RestoreDataArg,  
RestoreDataRes,

-- subscriber information enquiry types
ProvideSubscriberInfoArg,  
ProvideSubscriberInfoRes,  
SubscriberInfo,  
LocationInformation,  
SubscriberState,

-- any time information enquiry types
AnyTimeInterrogationArg,  
AnyTimeInterrogationRes

;  
IMPORTS
maxNumOfSS,
```
FROM MAP-SS-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-SS-DataTypes (14) version3 (3)}
FROM MAP-SS-Code {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-SS-Code (15) version3 (3)}
FROM MAP-BS-Code {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-BS-Code (20) version3 (3)}
FROM MAP-TS-Code {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-TS-Code (19) version3 (3)}
FROM MAP-CommonDataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}
FROM MAP-ExtensionDataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}

-- location registration types

UpdateLocationArg ::= SEQUENCE {
  imsi IMSI,
  msc-Number [1] ISDN-AddressString,
  vlr-Number ISDN-AddressString,
  lmsi [10] LMSI OPTIONAL,
  extensionContainer ExtensionContainer OPTIONAL,
...}

UpdateLocationRes ::= SEQUENCE {
  hlr-Number ISDN-AddressString,
  extensionContainer ExtensionContainer OPTIONAL,
...}

CancelLocationArg ::= CHOICE {
  imsi IMSI,
  imsi-WithLMSI IMSI-WithLMSI
}

PurgeMS-Arg ::= SEQUENCE {
  imsi IMSI,
  vlr-Number ISDN-AddressString,
...}

IMSI-WithLMSI ::= SEQUENCE {
  imsi IMSI,
  lmsi LMSI,
  -- a special value 00000000 indicates that the LMSI is not in use
...}
SendIdentificationRes ::= SEQUENCE {
  imsi           IMSI,
  authenticationSetList AuthenticationSetList  OPTIONAL,
  ...
}

AuthenticationSetList ::= SEQUENCE SIZE (1..5) OF
  AuthenticationSet

AuthenticationSet ::= SEQUENCE {
  rand           RAND,
  sres           SRES,
  kc             Kc,
  ...
}

RAND ::= OCTET STRING (SIZE (16))

SRES ::= OCTET STRING (SIZE (4))

Kc ::= OCTET STRING (SIZE (8))

-- handover types

PrepareHO-Arg ::= SEQUENCE {
  targetCellId  GlobalCellId OPTIONAL,
  ho-NumberNotRequired NULL OPTIONAL,
  bss-APDU      ExternalSignalInfo OPTIONAL,
  ...
}

PrepareHO-Res ::= SEQUENCE {
  handoverNumber ISDN-AddressString OPTIONAL,
  bss-APDU      ExternalSignalInfo OPTIONAL,
  ...
}

PrepareSubsequentHO-Arg ::= SEQUENCE {
  targetCellId  GlobalCellId,
  targetMSC-Number ISDN-AddressString,
  bss-APDU      ExternalSignalInfo,
  ...
}

-- authentication management types

SendAuthenticationInfoArg ::= IMSI

SendAuthenticationInfoRes ::= AuthenticationSetList

-- security management types

EquipmentStatus ::= ENUMERATED {
  whitelisted (0),
  blacklisted (1),
  greylisted (2)
}

-- subscriber management types

InsertSubscriberDataArg ::= SEQUENCE {
  imsi           IMSI [0] OPTIONAL,
  COMPONENTS OF SubscriberData,
  extensionContainer ExtensionContainer [14] OPTIONAL,
  ...
}
SubscriberData ::= SEQUENCE {
  msisdn  [1] ISDN-AddressString OPTIONAL,
  category  [2] Category OPTIONAL,
  subscriberStatus  [3] SubscriberStatus OPTIONAL,
  bearerServiceList  [4] BearerServiceList OPTIONAL,
  -- The exception handling for reception of unsupported / not allocated
  -- bearerServiceCodes is defined in section 6.8.1
  teleserviceList  [5] TeleserviceList OPTIONAL,
  -- The exception handling for reception of unsupported / not allocated
  -- teleserviceCodes is defined in section 6.8.1
  provisionedSS  [7] Ext-SS-InfoList OPTIONAL,
  odbData  [8] ODB-Data OPTIONAL,
  roamingRestrictionDueToUnsupportedFeature  [9] NULL OPTIONAL,
  regionalSubscriptionData  [10] ZoneCodeList OPTIONAL,
  vbsSubscriptionData  [11] VBSDataList OPTIONAL,
  vgsSubscriptionData  [12] VGCSDataList OPTIONAL,
  vlrCamelSubscriptionInfo  [13] VlrCamelSubscriptionInfo OPTIONAL,
}

Category ::= OCTET STRING (SIZE (1))
  -- The internal structure is defined in CCITT Rec Q.763.

SubscriberStatus ::= ENUMERATED {
  serviceGranted  (0),
  operatorDeterminedBarring  (1)
}

BearerServiceList ::= SEQUENCE SIZE (1..maxNumOfBearerServices) OF Ext-BearerServiceCode

maxNumOfBearerServices INTEGER ::= 50

TeleserviceList ::= SEQUENCE SIZE (1..maxNumOfTeleservices) OF Ext-TeleserviceCode

maxNumOfTeleservices INTEGER ::= 20

ODB-Data ::= SEQUENCE {
  odbGeneralData  ODB-GeneralData,
  odbHPLMNData  ODB-HPLMN-Data OPTIONAL,
  extensionContainer  ExtensionContainer OPTIONAL,
  ...
}

ODB-GeneralData ::= BIT STRING {
  allOG-CallsBarred  (0),
  internationalOGCallsBarred  (1),
  internationalOGCallsNotToHPLMN-CountryBarred  (2),
  interzonalOGCallsBarred  (3),
  interzonalOGCallsNotToHPLMN-CountryBarred  (4),
  premiumRateInformationOGCallsBarred  (5),
  allECT-Barred  (6),
  chargeableECT-Barred  (7),
  internationalECT-Barred  (8),
  interzonalECT-Barred  (9),
  doublyChargeableECT-Barred  (10),
  multipleECT-Barred  (11) (SIZE (15..32))
  -- exception handling: reception of unknown bit assignments in the
  -- ODB-GeneralData type shall be treated like unsupported ODB-GeneralData
}

ODB-HPLMN-Data ::= BIT STRING {
  plmnSpecificBarringType1  (0),
  plmnSpecificBarringType2  (1),
  plmnSpecificBarringType3  (2),
  plmnSpecificBarringType4  (3) (SIZE (4..32))
  -- exception handling: reception of unknown bit assignments in the
  -- ODB-HPLMN-Data type shall be treated like unsupported ODB-HPLMN-Data
}

Ext-SS-InfoList ::= SEQUENCE SIZE (1..maxNumOfSS) OF Ext-SS-Info
Ext-SS-Info ::= CHOICE {
  forwardingInfo  [0] Ext-ForwInfo,
  callBarringInfo [1] Ext-CallBarInfo,
  cug-Info       [2] CUG-Info,
  ss-Data        [3] Ext-SS-Data,
}

EMLPP-Info ::= SEQUENCE {
  maximumentitledPriority  EMLPP-Priority,
  defaultPriority         EMLPP-Priority,
  extensionContainer      ExtensionContainer OPTIONAL,
  ...}

EMLPP-Priority ::= INTEGER (0..15)
-- The mapping from the values A,B,0,1,2,3,4 to the integer-value is
-- specified as follows where A is the highest and 4 is the lowest
-- priority level
-- the integer values 7-15 are spare and shall be mapped to value 4
priorityLevelA EMLPP-Priority ::= 6
priorityLevelB EMLPP-Priority ::= 5
priorityLevel0 EMLPP-Priority ::= 0
priorityLevel1 EMLPP-Priority ::= 1
priorityLevel2 EMLPP-Priority ::= 2
priorityLevel3 EMLPP-Priority ::= 3
priorityLevel4 EMLPP-Priority ::= 4

Ext-ForwInfo ::= SEQUENCE {
  ss-Code               SS-Code,
  forwardingFeatureList Ext-ForwFeatureList,
  extensionContainer    [0] ExtensionContainer OPTIONAL,
  ...}

Ext-ForwFeatureList ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF Ext-ForwFeature

Ext-ForwFeature ::= SEQUENCE {
  basicService       Ext-BasicServiceCode OPTIONAL,
  ss-Status          [4] Ext-SS-Status,
  forwardedToNumber  [5] ISDN-AddressString OPTIONAL,
  forwardedToSubaddr ISDN-SubaddressString OPTIONAL,
  forwardingOptions  [6] Ext-ForwOptions OPTIONAL,
  noReplyCondTime    [7] Ext-NoRepCondTime OPTIONAL,
  extensionContainer [9] ExtensionContainer OPTIONAL,
  ...}

Ext-SS-Status ::= OCTET STRING (SIZE (1..5))
-- OCTET 1:
-- bits 8765: 0000 (unused)
-- bits 4321: Used to convey the "P bit", "R bit", "A bit" and "Q bit",
-- representing supplementary service state information
-- as defined in TS GSM 03.11
-- bit 4: "Q bit"
-- bit 3: "P bit"
-- bit 2: "R bit"
-- bit 1: "A bit"
-- OCTETS 2-5: reserved for future use. They shall be discarded if
-- received and not understood.
**Ext-ForwOptions** ::= OCTET STRING (SIZE (1..5))

-- OCTET 1:
-- bit 8: notification to forwarding party
-- 0  no notification
-- 1  notification
-- bit 7: 0 (unused)
-- bit 6: notification to calling party
-- 0  no notification
-- 1  notification
-- bit 5: 0 (unused)
-- bits 43: forwarding reason
-- 00  ms not reachable
-- 01  ms busy
-- 10  no reply
-- 11  unconditional
-- bits 21: 00 (unused)
-- OCTETS 2-5: reserved for future use. They shall be discarded if
-- received and not understood.

**Ext-NoRepCondTime** ::= INTEGER (1..100)

-- only values 5-30 are used.
-- values in the ranges 1-4 and 31-100 are reserved for future use
-- if received:
-- values 1-4 shall be mapped on to value 5
-- values 31-100 shall be mapped on to value 30

**Ext-CallBarInfo** ::= SEQUENCE {
  ss-Code SS-Code,
  callBarringFeatureList Ext-CallBarFeatureList,
  extensionContainer ExtensionContainer OPTIONAL,
  ...
}

**Ext-CallBarFeatureList** ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF Ext-CallBarringFeature

**Ext-CallBarringFeature** ::= SEQUENCE {
  basicService Ext-BasicServiceCode OPTIONAL,
  ss-Status [4] Ext-SS-Status,
  extensionContainer ExtensionContainer OPTIONAL,
  ...
}

**CUG-Info** ::= SEQUENCE {
  cug-SubscriptionList CUG-SubscriptionList,
  cug-FeatureList CUG-FeatureList OPTIONAL,
  extensionContainer [0] ExtensionContainer OPTIONAL,
  ...
}

**CUG-SubscriptionList** ::= SEQUENCE SIZE (0..maxNumOfCUG) OF CUG-Subscription

**CUG-Subscription** ::= SEQUENCE {
  cug-Index CUG-Index,
  cug-Interlock CUG-Interlock,
  intraCUG-Options IntraCUG-Options,
  basicServiceGroupList Ext-BasicServiceGroupList OPTIONAL,
  extensionContainer [0] ExtensionContainer OPTIONAL,
  ...
}

**CUG-Index** ::= INTEGER (0..32767)

-- the internal structure is defined in ETS 300 138.

**CUG-Interlock** ::= OCTET STRING (SIZE (4))

**IntraCUG-Options** ::= ENUMERATED {
  noCUG-Restrictions (0),
  cugIC-CallBarred (1),
  cugOG-CallBarred (2)
}

maxNumOfCUG INTEGER ::= 10
CUG-FeatureList ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF CUG-Feature

Ext-BasicServiceGroupList ::= SEQUENCE SIZE (1..maxNumOfExt-BasicServiceGroups) OF Ext-BasicServiceCode

maxNumOfExt-BasicServiceGroups INTEGER ::= 32

CUG-Feature ::= SEQUENCE {
    basicService Ext-BasicServiceCode OPTIONAL,
    preferentialCUG-Indicator CUG-Index OPTIONAL,
    interCUG-Restrictions InterCUG-Restrictions,
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

InterCUG-Restrictions ::= OCTET STRING (SIZE (1))
    -- bits 876543: 000000 (unused)
    -- Exception handling:
    -- bits 876543 shall be ignored if received and not understood
    -- bits 21
    -- 00 CUG only facilities
    -- 01 CUG with outgoing access
    -- 10 CUG with incoming access
    -- 11 CUG with both outgoing and incoming access

Ext-SS-Data ::= SEQUENCE {
    ss-Code SS-Code,
    ss-Status [4] Ext-SS-Status,
    ss-SubscriptionOption SS-SubscriptionOption OPTIONAL,
    basicServiceGroupList Ext-BasicServiceGroupList OPTIONAL,
    extensionContainer [5] ExtensionContainer OPTIONAL,
    ...}

ZoneCodeList ::= SEQUENCE SIZE (1..maxNumOfZoneCodes) OF ZoneCode

ZoneCode ::= OCTET STRING (SIZE (2))
    -- internal structure is defined in TS GSM 03.03

maxNumOfZoneCodes INTEGER ::= 10

InsertSubscriberDataRes ::= SEQUENCE {
    teleserviceList [1] TeleserviceList OPTIONAL,
    bearerServiceList [2] BearerServiceList OPTIONAL,
    ss-List [3] SS-List OPTIONAL,
    odb-GeneralData [4] ODB-GeneralData OPTIONAL,
    regionalSubscriptionResponse [5] RegionalSubscriptionResponse OPTIONAL,
    supportedCamelPhases [6] SupportedCamelPhases OPTIONAL,
    extensionContainer [7] ExtensionContainer OPTIONAL,
    ...}

RegionalSubscriptionResponse ::= ENUMERATED {
    mac-AreaRestricted (0),
    tooManyZoneCodes (1),
    zoneCodesConflict (2),
    regionalSubscNotSupported (3)}

DeleteSubscriberDataArg ::= SEQUENCE {
    imsi [0] IMSI,
    basicServiceList [1] BasicServiceList OPTIONAL,
    ss-List [2] SS-List OPTIONAL,
    roamingRestrictionDueToUnsupportedFeature [4] NULL OPTIONAL,
    regionalSubscriptionIdentifier [5] ZoneCode OPTIONAL,
    vbsGroupIndication [7] NULL OPTIONAL,
    vgcGroupIndication [8] NULL OPTIONAL,
    camelSubscriptionInfoWithdraw [9] NULL OPTIONAL,
    extensionContainer [6] ExtensionContainer OPTIONAL,
    ...}
BasicServiceList ::= SEQUENCE SIZE (1..maxNumOfBasicServices) OF Ext-BasicServiceCode

maxNumOfBasicServices INTEGER ::= 70

DeleteSubscriberDataRes ::= SEQUENCE {
    regionalSubscriptionResponse [0] RegionalSubscriptionResponse OPTIONAL,
    extensionContainer ExtensionContainer OPTIONAL,
    ...
}

VlrCamelSubscriptionInfo ::= SEQUENCE {
    o-CSI [0] O-CSI OPTIONAL,
    extensionContainer [1] ExtensionContainer OPTIONAL,
    ...
}

O-CSI ::= SEQUENCE {
    o-BcsmCamelTDPDataList O-BcsmCamelTDPDataList, 
    extensionContainer ExtensionContainer OPTIONAL,
    ...
}

O-BcsmCamelTDPData ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF O-BcsmCamelTDPData

maxNumOfCamelTDPData INTEGER ::= 10

O-BcsmTriggerDetectionPoint ::= ENUMERATED {
   collectedInfo (2),
   ...
}

DefaultCallHandling ::= ENUMERATED {
   continueCall (0),
   releaseCall (1),
   ...
}

SupportedCamelPhases ::= BIT STRING {
   phase1 (0) } (SIZE (1..16))

ResetArg ::= SEQUENCE {
   hlr-Number ISDN-AddressString,
   hlr-List HLR-List OPTIONAL,
   ...
}

RestoreDataArg ::= SEQUENCE {
   imsi IMSI,
   lmsi LMSI OPTIONAL,
   extensionContainer ExtensionContainer OPTIONAL,
   ...
}

RestoreDataRes ::= SEQUENCE {
   hlr-Number ISDN-AddressString,
   msNotReachable NULL OPTIONAL,
   extensionContainer ExtensionContainer OPTIONAL,
   ...
}

-- VBS/VGCS types

ETS
VBSDataList ::= SEQUENCE SIZE (1..maxNumOfVBSGroupIds)

OF VoiceBroadcastData

VGCSDataList ::= SEQUENCE SIZE (1..maxNumOfVGCSGroupIds)

OF VoiceGroupCallData

maxNumOfVBSGroupIds INTEGER ::= 50

maxNumOfVGCSGroupIds INTEGER ::= 50

VoiceGroupCallData ::= SEQUENCE {
  groupId  GroupId,
  extensionContainer ExtensionContainer OPTIONAL,
  ...
}

VoiceBroadcastData ::= SEQUENCE {
  groupId  GroupId,
  broadcastInitEntitlement NULL OPTIONAL,
  extensionContainer ExtensionContainer OPTIONAL,
  ...
}

GroupId ::= OCTET STRING (SIZE (3))

-- Refers to the Group Identification as specified in GSM TS 03.03
-- and 03.68/ 03.69
-- provide subscriber info types

ProvideSubscriberInfoArg ::= SEQUENCE {
  imsi  [0] IMSI,
  lmsi  [1] LMSI OPTIONAL,
  requestedInfo [2] RequestedInfo,
  extensionContainer [3] ExtensionContainer OPTIONAL,
  ...
}

ProvideSubscriberInfoRes ::= SEQUENCE {
  subscriberInfo SubscriberInfo,
  extensionContainer ExtensionContainer OPTIONAL,
  ...
}

SubscriberInfo ::= SEQUENCE {
  locationInformation [0] LocationInformation OPTIONAL,
  subscriberState [1] SubscriberState OPTIONAL,
  extensionContainer [2] ExtensionContainer OPTIONAL,
  ...
}

RequestedInfo ::= SEQUENCE {
  locationInformation [0] NULL OPTIONAL,
  subscriberState [1] NULL OPTIONAL,
  extensionContainer [2] ExtensionContainer OPTIONAL,
  ...
}

LocationInformation ::= SEQUENCE {
  ageOfLocationInformation AgeOfLocationInformation OPTIONAL,
  geographicalInformation [0] GeographicalInformation OPTIONAL,
  vlr-number [1] ISDN-AddressString OPTIONAL,
  locationNumber [2] LocationNumber OPTIONAL,
  cellIdOrLAI [3] CellIdOrLAI OPTIONAL,
  extensionContainer [4] ExtensionContainer OPTIONAL,
  ...
}

AgeOfLocationInformation ::= INTEGER (0..32767)

-- the value represents the elapsed time in minutes since the last
-- network contact of the mobile station (i.e. the actuality of the
-- location information).
-- value “0” indicates that the MS is currently in contact with the
-- network
-- value “32767” indicates that the location information is at least
-- 32767 minutes old

GeographicalInformation ::= OCTET STRING (SIZE (8))

-- Refers to geographical Information defined in GSM 03.32.
-- Only the description of an ellipsoid point with uncertainty circle
-- as specified in GSM 03.32 is allowed to be used
-- The internal structure according to GSM 03.32 is as follows:
-- Type of shape (ellipsoid point with uncertainty circle) 1 octet
-- Degrees of Latitude 3 octets
-- Degrees of Longitude 3 octets
-- Uncertainty code 1 octet
LocationNumber ::= OCTET STRING (SIZE (2..10))
-- the internal structure is defined in CCITT Rec Q.763

SubscriberState ::= CHOICE {
    assumedIdle [0] NULL,
    camelBusy [1] NULL,
    netDetNotReachable NotReachableReason,
    notProvidedFromVLR [2] NULL
}

NotReachableReason ::= ENUMERATED {
    msPurged (0),
    imsiDetached (1),
    restrictedArea (2),
    notRegistered (3)
}
-- any time interrogation info types

AnyTimeInterrogationArg ::= SEQUENCE {
    subscriberIdentity [0] SubscriberIdentity,
    requestedInfo [1] RequestedInfo,
    gsmSCF-Address [3] ISDN-AddressString,
    extensionContainer [2] ExtensionContainer OPTIONAL,
    ...
}

AnyTimeInterrogationRes ::= SEQUENCE {
    subscriberInfo SubscriberInfo,
    extensionContainer ExtensionContainer OPTIONAL,
    ...
}

SubscriberIdentity ::= CHOICE {
    imsi [0] IMSI,
    msisdn [1] ISDN-AddressString
}
14.7.2 Operation and maintenance data types

MAP-OM-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-OM-DataTypes (12) version3 (3)}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS
  ActivateTraceModeArg,
  ActivateTraceModeRes,
  DeactivateTraceModeArg,
  DeactivateTraceModeRes

;

IMPORTS
  AddressString,
  IMSI
FROM MAP-CommonDataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}

ExtensionContainer
FROM MAP-ExtensionDataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}

;

ActivateTraceModeArg ::= SEQUENCE {
  imsi   [0] IMSI  OPTIONAL,
  traceReference [1] TraceReference,
  traceType      [2] TraceType,
  omc-Id         [3] AddressString OPTIONAL,
  extensionContainer [4] ExtensionContainer OPTIONAL,
  ...}

TraceReference ::= OCTET STRING (SIZE (1..2))

-- Trace types are fully defined in TS GSM 12.08.

TraceType ::= OCTET STRING (SIZE (1..2))

    ::= INTEGER
      (0..255)

ActivateTraceModeRes ::= SEQUENCE {
  extensionContainer [0] ExtensionContainer OPTIONAL,
  ...
}

DeactivateTraceModeArg ::= SEQUENCE {
  imsi   [0] IMSI  OPTIONAL,
  traceReference [1] TraceReference,
  extensionContainer [2] ExtensionContainer OPTIONAL,
  ...
}

DeactivateTraceModeRes ::= SEQUENCE {
  extensionContainer [0] ExtensionContainer OPTIONAL,
  ...
}

END
14.7.3 Call handling data types

```plaintext
MAP-CH-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-CH-DataTypes (13) version3 (3)
}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS
SendRoutingInfoArg,
SendRoutingInfoRes,
ProvideRoamingNumberArg,
ProvideRoamingNumberRes,
ResumeCallHandlingArg,
ResumeCallHandlingRes,
NumberOfForwarding,
SuppressionOfAnnouncement,
CallReferenceNumber;

IMPORTS
maxNumOfCamelTDPData,
SubscriberInfo,
ServiceKey,
DefaultCallHandling,
SupportedCamelPhases,
CUG-Interlock,
O-CSI
FROM MAP-MS-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-MS-DataTypes (11) version3 (3)
}
ForwardingOptions,
SS-List
FROM MAP-SS-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-SS-DataTypes (14) version3 (3)
}
ISDN-AddressString,
ISDN-SubaddressString,
ExternalSignalInfo,
IMSI,
LMSI,
Ext-BasicServiceCode
FROM MAP-CommonDataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)
}
ExtensionContainer
FROM MAP-ExtensionDataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)
};

CUG-CheckInfo ::= SEQUENCE {
cug-Interlock CUG-Interlock,
cug-OutgoingAccess NULL   OPTIONAL,
extensionContainer ExtensionContainer OPTIONAL,
...}

NumberOfForwarding ::= INTEGER (1..5)
```
SendRoutingInfoArg ::= SEQUENCE {
    msisdn [0] ISDN-AddressString,
    cug-CheckInfo [1] CUG-CheckInfo OPTIONAL,
    numberOfForwarding [2] NumberOfForwarding OPTIONAL,
    interrogationType [3] InterrogationType,
    or-Interrogation [4] NULL OPTIONAL,
    or-Capability [5] OR-Phase OPTIONAL,
    gmsc-Address [6] ISDN-AddressString,
    callReferenceNumber [7] CallReferenceNumber OPTIONAL,
    forwardingReason [8] ForwardingReason OPTIONAL,
    basicServiceGroup [9] Ext-BasicServiceCode OPTIONAL,
    networkSignalInfo [10] ExternalSignalInfo OPTIONAL,
    camelInfo [11] CamelInfo OPTIONAL,
    suppressionOfAnnouncement [12] SuppressionOfAnnouncement OPTIONAL,
    extensionContainer [13] ExtensionContainer OPTIONAL,
    ...
}

SuppressionOfAnnouncement ::= NULL

InterrogationType ::= ENUMERATED {
    basicCall (0),
    forwarding (1)
}

OR-Phase ::= INTEGER (1..127)

CallReferenceNumber ::= OCTET STRING (SIZE (1..8))

ForwardingReason ::= ENUMERATED {
    notReachable (0),
    busy (1),
    noReply (2)
}

SendRoutingInfoRes ::= [3] SEQUENCE {
    imsi [0] IMSI OPTIONAL,
    -- IMSI must be present if SendRoutingInfoRes is not segmented.
    -- If the TC-Result-NL segmentation option is taken the IMSI must be
    -- present in one segmented transmission of SendRoutingInfoRes.
    extendedRoutingInfo [1] ExtendedRoutingInfo OPTIONAL,
    cug-CheckInfo [2] CUG-CheckInfo OPTIONAL,
    gmsc-Address [3] ISDN-AddressString,
    callReferenceNumber [4] CallReferenceNumber OPTIONAL,
    forwardingInterrogationRequired [5] NULL OPTIONAL,
    vmsc-Address [6] ISDN-AddressString OPTIONAL,
    extensionContainer [7] ExtensionContainer OPTIONAL,
    ...
}

RoutingInfo ::= CHOICE {
    roamingNumber ISDN-AddressString,
    forwardingData ForwardingData
}

ForwardingData ::= SEQUENCE {
    forwardedToNumber [0] ISDN-AddressString OPTIONAL,
    forwardedToSubaddress [1] ISDN-SubaddressString OPTIONAL,
    forwardingOptions [2] ForwardingOptions OPTIONAL,
    extensionContainer [3] ExtensionContainer OPTIONAL,
    ...
}

ProvideRoamingNumberArg ::= SEQUENCE {
    imsi [0] IMSI,
    msc-Number [1] ISDN-AddressString,
    msisdn [2] ISDN-AddressString OPTIONAL,
    lmsi [3] LMSI OPTIONAL,
    gsm-BearerCapability [4] ExternalSignalInfo OPTIONAL,
    networkSignalInfo [5] ExternalSignalInfo OPTIONAL,
    suppressionOfAnnouncement [6] SuppressionOfAnnouncement OPTIONAL,
    gmsc-Address [7] ISDN-AddressString OPTIONAL,
    callReferenceNumber [8] CallReferenceNumber OPTIONAL,
    or-Interrogation [9] NULL OPTIONAL,
    extensionContainer [10] ExtensionContainer OPTIONAL,
    ...
}

ProvideRoamingNumberRes ::= SEQUENCE {
    roamingNumber ISDN-AddressString,
    extensionContainer ExtensionContainer OPTIONAL,
    ...
}
ResumeCallHandlingArg ::= SEQUENCE {
    callReferenceNumber [0] CallReferenceNumber,
    basicServiceGroup [1] Ext-BasicServiceCode,
    forwardingData [2] ForwardingData,
    imsi [3] IMSI,
    cug-CheckInfo [4] CUG-CheckInfo OPTIONAL,
    o-CSI [5] O-CSI OPTIONAL,
    extensionContainer [7] ExtensionContainer OPTIONAL,
    ...
}

ResumeCallHandlingRes ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...
}

CamelInfo ::= SEQUENCE {
    supportedCamelPhases SupportedCamelPhases,
    suppress-T-CSI NULL OPTIONAL,
    extensionContainer ExtensionContainer OPTIONAL,
    ...
}

ExtendedRoutingInfo ::= CHOICE {
    routingInfo RoutingInfo,
    camelRoutingInfo [8] CamelRoutingInfo
}

CamelRoutingInfo ::= SEQUENCE {
    forwardingData ForwardingData OPTIONAL,
    gmscCamelSubscriptionInfo [0] GmscCamelSubscriptionInfo,
    extensionContainer [1] ExtensionContainer OPTIONAL,
    ...
}

GmscCamelSubscriptionInfo ::= SEQUENCE {
    t-CSI [0] T-CSI OPTIONAL,
    o-CSI [1] O-CSI OPTIONAL,
    extensionContainer [2] ExtensionContainer OPTIONAL,
    ...
}

T-CSI ::= SEQUENCE {
    t-BcsmCamelTDPDataList T-BcsmCamelTDPDataList,
    extensionContainer ExtensionContainer OPTIONAL,
    ...
}

T-BcsmCamelTDPDataList ::= SEQUENCE SIZE (1..maxNumOfCamelTDPData) OF
    T-BcsmCamelTDPData

T-BcsmCamelTDPData ::= SEQUENCE {
    t-BcsmTriggerDetectionPoint T-BcsmTriggerDetectionPoint,
    serviceKey ServiceKey,
    gsmSCF-Address [0] ISDN-AddressString,
    defaultCallHandling [1] DefaultCallHandling,
    extensionContainer [2] ExtensionContainer OPTIONAL,
    ...
}

T-BcsmTriggerDetectionPoint ::= ENUMERATED {
    termAttemptAuthorized(12),
    ...
} -- exception handling:

-- For T-BcsmCamelTDPData sequences containing this parameter with any other
-- value than the ones listed the receiver shall ignore the whole
-- T-BcsmCamelTDPData sequence.
14.7.4 Supplementary service data types

MAP-SS-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-SS-DataTypes (14) version3 (3)
} DEFINITIONS
IMPLICIT TAGS ::= BEGIN EXPORTS
  RegisterSS-Arg,
  SS-Info,
  SS-Status,
  SS-SubscriptionOption,
  SS-ForBS-Code,
  InterrogateSS-Res,
  USSD-Arg,
  USSD-Res,
  Password,
  GuidanceInfo,
  SS-List,
  SS-InfoList,
  OverrideCategory,
  CliRestrictionOption,
  NoReplyConditionTime,
  ForwardingOptions,
  maxNumOfSS,
  SS-Data,
  USSD-DataCodingScheme,
  USSD-String ;
IMPORTS
  AddressString,
  ISDN-AddressString,
  ISDN-SubaddressString,
  BasicServiceCode
FROM MAP-CommonDataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)
} SS-Code
FROM MAP-SS-Code {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-SS-Code (15) version3 (3)
} ;

RegisterSS-Arg ::= SEQUENCE{
  ss-Code  SS-Code,
  basicService BasicServiceCode OPTIONAL,
  forwardedToNumber [4] AddressString OPTIONAL,
  forwardedToSubaddress [6] ISDN-SubaddressString OPTIONAL,
  noReplyConditionTime [5] NoReplyConditionTime OPTIONAL,
  ...}

NoReplyConditionTime ::= INTEGER (5..30)

SS-Info ::= CHOICE {
  forwardingInfo [0] ForwardingInfo,
  callBarringInfo [1] CallBarringInfo,
  ss-Data [3] SS-Data
}

ForwardingInfo ::= SEQUENCE {
  ss-Code  SS-Code  OPTIONAL,
  forwardedFeatureList ForwardingFeatureList,
  ...}

ForwardingFeatureList ::= SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF
  ForwardingFeature
ForwardingFeature ::= SEQUENCE {
  basicService BasicServiceCode OPTIONAL,
  ss-Status [4] SS-Status OPTIONAL,
  forwardedToNumber [5] ISDN-AddressString OPTIONAL,
  forwardedToSubaddress [8] ISDN-SubaddressString OPTIONAL,
  forwardingOptions [6] ForwardingOptions OPTIONAL,
  noReplyConditionTime [7] NoReplyConditionTime OPTIONAL,
  ...
}

SS-Status ::= OCTET STRING (SIZE (1))
-- bits 8765: 0000 (unused)
-- bits 4321: Used to convey the "P bit", "R bit", "A bit" and "Q bit",
-- representing supplementary service state information
-- as defined in TS GSM 03.11
-- bit 4: "Q bit"
-- bit 3: "P bit"
-- bit 2: "R bit"
-- bit 1: "A bit"

ForwardingOptions ::= OCTET STRING (SIZE (1))
-- bit 8: notification to forwarding party
-- 0  no notification
-- 1  notification
-- bit 7: 0 (unused)
-- bit 6: notification to calling party
-- 0  no notification
-- 1  notification
-- bit 5: 0 (unused)
-- bits 43: forwarding reason
-- 00  ms not reachable
-- 01  ms busy
-- 10  no reply
-- 11 unconditional
-- bits 21: 00 (unused)

CallBarringInfo ::= SEQUENCE {
  ss-Code SS-Code OPTIONAL,
  callBarringFeatureList CallBarringFeatureList,
  ...
}

CallBarringFeatureList ::= SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF
  CallBarringFeature

CallBarringFeature ::= SEQUENCE {
  basicService BasicServiceCode OPTIONAL,
  ss-Status [4] SS-Status OPTIONAL,
  ...
}

SS-Data ::= SEQUENCE {
  ss-Code SS-Code OPTIONAL,
  ss-Status [4] SS-Status OPTIONAL,
  ss-SubscriptionOption SS-SubscriptionOption OPTIONAL,
  basicServiceGroupList BasicServiceGroupList OPTIONAL,
  ...
}

SS-SubscriptionOption ::= CHOICE {
  cliRestrictionOption [2] CliRestrictionOption,
  overrideCategory [1] OverrideCategory
}

CliRestrictionOption ::= ENUMERATED {
  permanent (0),
  temporaryDefaultRestricted (1),
  temporaryDefaultAllowed (2)
}

OverrideCategory ::= ENUMERATED {
  overrideEnabled (0),
  overrideDisabled (1)
}
SS-ForBS-Code ::= SEQUENCE {
    ss-Code SS-Code,
    basicService BasicServiceCode OPTIONAL,
    ...
}

Cli-RestrictionInfo ::= SEQUENCE {
    ss-Status SS-Status,
    cliRestrictionOption CliRestrictionOption OPTIONAL,
    ...
}

InterrogateSS-Res ::= CHOICE {
    ss-Status [0] SS-Status,
    basicServiceGroupList [2] BasicServiceGroupList,
    forwardingFeatureList [3] ForwardingFeatureList,
    cli-RestrictionInfo [4] Cli-RestrictionInfo
}

USSD-Arg ::= SEQUENCE {
    ussd-DataCodingScheme USSD-DataCodingScheme,
    ussd-String USSD-String,
    ...
}

USSD-Res ::= SEQUENCE {
    ussd-DataCodingScheme USSD-DataCodingScheme,
    ussd-String USSD-String,
    ...
}

USSD-DataCodingScheme ::= OCTET STRING (SIZE (1))
-- The structure of the USSD-DataCodingScheme is defined by
-- the Cell Broadcast Data Coding Scheme as described in
-- TS GSM 03.38

USSD-String ::= OCTET STRING (SIZE (1..maxUSSD-StringLength))
-- The structure of the contents of the USSD-String is dependent
-- on the USSD-DataCodingScheme as described in TS GSM 03.38.

maxUSSD-StringLength INTEGER ::= 160

Password ::= NumericString
    (FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9"))
    (SIZE (4))

GuidanceInfo ::= ENUMERATED {
    enterPW (0),
    enterNewPW (1),
    enterNewPW-Again (2)
-- How this information is really delivered to the subscriber
-- (display, announcement, ...) is not part of this
-- specification.

SS-List ::= SEQUENCE SIZE (1..maxNumOfSS) OF
    SS-Code

maxNumOfSS INTEGER ::= 30

SS-InfoList ::= SEQUENCE SIZE (1..maxNumOfSS) OF
    SS-Info

BasicServiceGroupList ::= SEQUENCE SIZE (1..maxNumOfBasicServiceGroups) OF
    BasicServiceCode

maxNumOfBasicServiceGroups INTEGER ::= 13

END
14.7.5 Supplementary service codes

MAP-SS-Code {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-SS-Code (15) version3 (3)}

DEFINITIONS
::=
BEGIN

SS-Code ::= OCTET STRING (SIZE (1))
  -- This type is used to represent the code identifying a single
  -- supplementary service, a group of supplementary services, or
  -- all supplementary services. The services and abbreviations
  -- used are defined in TS GSM 02.04. The internal structure is
  -- defined as follows:
  --
  -- bits 87654321: group (bits 8765), and specific service
  -- (bits 4321)

allSS    SS-Code ::= '00000000'B
  -- reserved for possible future use
  -- all SS

allLineIdentificationSS  SS-Code ::= '00010000'B
  -- reserved for possible future use
  -- all line identification SS
clip     SS-Code ::= '00010001'B
  -- calling line identification presentation
clir     SS-Code ::= '00010010'B
  -- calling line identification restriction
colp     SS-Code ::= '00010011'B
  -- connected line identification presentation
colr     SS-Code ::= '00010100'B
  -- connected line identification restriction
mci      SS-Code ::= '00010101'B
  -- reserved for possible future use
  -- malicious call identification

allForwardingSS  SS-Code ::= '00100000'B
  -- all forwarding SS
 cfu       SS-Code ::= '00100001'B
  -- call forwarding unconditional

allCondForwardingSS  SS-Code ::= '00101000'B
  -- all conditional forwarding SS
 cfb       SS-Code ::= '00101001'B
  -- call forwarding on mobile subscriber busy
 cfnry     SS-Code ::= '00101010'B
  -- call forwarding on no reply
 cfnrc     SS-Code ::= '00101011'B
  -- call forwarding on mobile subscriber not reachable

allCallOfferingSS  SS-Code ::= '01000000'B
  -- reserved for possible future use
  -- all call offering SS includes also all forwarding SS
 ect       SS-Code ::= '01000001'B
  -- explicit call transfer
 mah       SS-Code ::= '01000010'B
  -- reserved for possible future use
  -- mobile access hunting

allCallCompletionSS  SS-Code ::= '01000011'B
  -- reserved for possible future use
  -- all call completion SS
 cw        SS-Code ::= '01000100'B
  -- call waiting
  hold      SS-Code ::= '01000101'B
  -- call hold
 ccbv      SS-Code ::= '01000110'B
  -- reserved for possible future use
  -- completion of call to busy subscribers
allMultiPartySS  SS-Code ::= '01010000'B
-- reserved for possible future use
-- all multiparty SS
multiParty  SS-Code ::= '01010001'B
-- multiparty

allCommunityOfInterest-SS  SS-Code ::= '01100000'B
-- reserved for possible future use
-- all community of interest SS
cug  SS-Code ::= '01100001'B
-- closed user group

allChargingSS  SS-Code ::= '01110000'B
-- reserved for possible future use
-- all charging SS
aoci  SS-Code ::= '01110001'B
-- advice of charge information
acc  SS-Code ::= '01110010'B
-- advice of charge charging

allAdditionalInfoTransferSS  SS-Code ::= '10000000'B
-- reserved for possible future use
-- all additional information transfer SS
uus  SS-Code ::= '10000001'B
-- UUS user-to-user signalling

allBarringSS  SS-Code ::= '10010000'B
-- all barring SS
barringOfOutgoingCalls  SS-Code ::= '10010001'B
-- barring of outgoing calls
baoc  SS-Code ::= '10010010'B
-- barring of all outgoing calls
boic  SS-Code ::= '10010011'B
-- barring of outgoing international calls
boicExHC  SS-Code ::= '10010100'B
-- barring of outgoing international calls except those directed
-- to the home PLMN
barringOfIncomingCalls  SS-Code ::= '10011001'B
-- barring of incoming calls
baic  SS-Code ::= '10011010'B
-- barring of all incoming calls
bicRoam  SS-Code ::= '10011011'B
-- barring of incoming calls when roaming outside home PLMN
-- Country

allPLMN-specificSS  SS-Code ::= '11110000'B
plmn-specificSS-1  SS-Code ::= '11110001'B
plmn-specificSS-2  SS-Code ::= '11110010'B
plmn-specificSS-3  SS-Code ::= '11110011'B
plmn-specificSS-4  SS-Code ::= '11111000'B
plmn-specificSS-5  SS-Code ::= '11111001'B
plmn-specificSS-6  SS-Code ::= '11111010'B
plmn-specificSS-7  SS-Code ::= '11111011'B
plmn-specificSS-8  SS-Code ::= '11111100'B
plmn-specificSS-9  SS-Code ::= '11111101'B
plmn-specificSS-A  SS-Code ::= '11111110'B
plmn-specificSS-B  SS-Code ::= '11111111'B
plmn-specificSS-C  SS-Code ::= '111111100'B
plmn-specificSS-D  SS-Code ::= '111111101'B
plmn-specificSS-E  SS-Code ::= '111111110'B
plmn-specificSS-F  SS-Code ::= '111111111'B

allCallPrioritySS  SS-Code ::= '10100000'B
-- reserved for possible future use
-- all call priority SS
emlpp  SS-Code ::= '10100001'B
-- enhanced Multilevel Precedence Pre-emption (EMLPP) service
END
14.7.6 Short message data types

MAP-SM-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-SM-DataTypes (16) version3 (3)}

DEFINITIONS

IMPLICIT TAGS

::=
BEGIN

EXPORTS
RoutingInfoForSM-Arg,
RoutingInfoForSM-Res,
MO-ForwardSM-Arg,
MO-ForwardSM-Res,
MT-ForwardSM-Arg,
MT-ForwardSM-Res,
ReportSM-DeliveryStatusArg,
ReportSM-DeliveryStatusRes,
AlertServiceCentreArg,
InformServiceCentreArg,
ReadyForSM-Arg,
SM-DeliveryOutcome,
AlertReason;

IMPORTS
AddressString,
ISDN-AddressString,
SignalInfo,
IMSI,
LMSI
FROM MAP-CommonDataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}

AbsentSubscriberDiagnosticSM
FROM MAP-ER-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-ER-DataTypes (17) version3 (3)}

ExtensionContainer
FROM MAP-ExtensionDataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
  gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}

RoutingInfoForSM-Arg ::= SEQUENCE {
  msisdn[0] ISDN-AddressString,
  sm-RP-PRI[1] BOOLEAN,
  serviceCentreAddress[2] AddressString,
  extensionContainer[6] ExtensionContainer OPTIONAL,
  ...}

RoutingInfoForSM-Res ::= SEQUENCE {
  imsi IMSI,
  locationInfoWithLMSI[0] LocationInfoWithLMSI,
  extensionContainer[4] ExtensionContainer OPTIONAL,
  ...}

LocationInfoWithLMSI ::= SEQUENCE {
  msc-Number[1] ISDN-AddressString,
  lmsi LMSI OPTIONAL,
  extensionContainer ExtensionContainer OPTIONAL,
  ...}

MO-ForwardSM-Arg ::= SEQUENCE {
  sm-RP-DA SM-RP-DA,
  sm-RP-OA SM-RP-OA,
  sm-RP-UI SignalInfo,
  extensionContainer ExtensionContainer OPTIONAL,
  ...}
MO-ForwardSM-Res ::= SEQUENCE {
  sm-RP-UI  SignalInfo, OPTIONAL,
  extensionContainer ExtensionContainer OPTIONAL,
  ...
}

MT-ForwardSM-Arg ::= SEQUENCE {
  sm-RP-DA  SM-RP-DA,
  sm-RP-OA  SM-RP-OA,
  sm-RP-UI  SignalInfo, OPTIONAL,
  moreMessagesToSend NULL OPTIONAL,
  extensionContainer ExtensionContainer OPTIONAL,
  ...
}

MT-ForwardSM-Res ::= SEQUENCE {
  sm-RP-UI  SignalInfo OPTIONAL,
  extensionContainer ExtensionContainer OPTIONAL,
  ...
}

SM-RP-DA ::= CHOICE {
  imsi [0] IMSI,
  lmsi [1] LMSI,
  serviceCentreAddressDA [4] AddressString,
}

SM-RP-OA ::= CHOICE {
  msisdn [2] ISDN-AddressString,
  serviceCentreAddressOA [4] AddressString,
}

ReportSM-DeliveryStatusArg ::= SEQUENCE {
  msisdn ISDN-AddressString,
  serviceCentreAddress AddressString,
  sm-DeliveryOutcome SM-DeliveryOutcome,
  absentSubscriberDiagnosticSM [0] AbsentSubscriberDiagnosticSM OPTIONAL,
  extensionContainer [1] ExtensionContainer OPTIONAL,
  ...
}

SM-DeliveryOutcome ::= ENUMERATED {
  memoryCapacityExceeded (0),
  absentSubscriber (1),
  successfulTransfer (2)
}

ReportSM-DeliveryStatusRes ::= SEQUENCE {
  storedMSISDN ISDN-AddressString OPTIONAL,
  extensionContainer ExtensionContainer OPTIONAL,
  ...
}

AlertServiceCentreArg ::= SEQUENCE {
  msisdn ISDN-AddressString,
  serviceCentreAddress AddressString,
  ...
}

InformServiceCentreArg ::= SEQUENCE {
  storedMSISDN ISDN-AddressString OPTIONAL,
  mw-Status MW-Status OPTIONAL,
  extensionContainer ExtensionContainer OPTIONAL,
  ...
}

MW-Status ::= BIT STRING {
  sc-AddressNotIncluded (0),
  mnrf-Set (1),
  mcef-Set (2) (SIZE (6..16))
-- exception handling:
-- bits 3 to 15 shall be ignored if received and not understood
}

ReadyForSM-Arg ::= SEQUENCE {
  imsi [0] IMSI,
  alertReason AlertReason,
  ...
}

AlertReason ::= ENUMERATED {
  ms-Present (0),
  memoryAvailable (1)
}
14.7.7 Error data types

```plaintext
MAP-ER-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-ER-DataTypes (17) version3 (3)

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS
  RoamingNotAllowedParam,
  CallBarredParam,
  CUG-RejectParam,
  SS-IncompatibilityCause,
  PW-RegistrationFailureCause,
  SM-DeliveryFailureCause,
  SystemFailureParam,
  DataMissingParam,
  UnexpectedDataParam,
  OR-NotAllowedParam,
  UnknownSubscriberParam,
  NumberChangedParam,
  UnidentifiedSubParam,
  IllegalSubscriberParam,
  IllegalEquipmentParam,
  BearerServNotProvParam,
  TeleservNotProvParam,
  TracingBufferFullParam,
  NoRoamingNbParam,
  AbsentSubscriberParam,
  BusySubscriberParam,
  NoSubscriberReplyParam,
  ForwardingViolationParam,
  ForwardingFailedParam,
  ATI-NotAllowedParam,
  SubBusyForMT-SMS-Param,
  MessageWaitListFullParam,
  AbsentSubscriberSM-Param,
  AbsentSubscriberDiagnosticSM

; IMPORTS
  SS-Status
FROM MAP-SS-DataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-SS-DataTypes (14) version3 (3)

  SignalInfo,
  BasicServiceCode,
  NetworkResource
FROM MAP-CommonDataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)

  SS-Code
FROM MAP-SS-Code {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-SS-Code (15) version3 (3)

  ExtensionContainer
FROM MAP-ExtensionDataTypes {
  ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)

  }

RoamingNotAllowedParam ::= SEQUENCE {
  roamingNotAllowedCause RoamingNotAllowedCause,
  extensionContainer ExtensionContainer OPTIONAL,
  ...}
```

ETS
RoamingNotAllowedCause ::= ENUMERATED {
    plmnRoamingNotAllowed (0),
    operatorDeterminedBarring (3)}

CallBarredParam ::= CHOICE {
    callBarringCause CallBarringCause,
    -- call BarringCause must not be used in version 3
    extensibleCallBarredParam ExtensibleCallBarredParam
    -- extensibleCallBarredParam must not be used in version <3
}

CallBarringCause ::= ENUMERATED {
    barringServiceActive (0),
    operatorBarring (1)}

ExtensibleCallBarredParam ::= SEQUENCE {
    callBarringCause CallBarringCause OPTIONAL,
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

CUG-RejectParam ::= SEQUENCE {
    cug-RejectCause CUG-RejectCause OPTIONAL,
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

CUG-RejectCause ::= ENUMERATED {
    incomingCallsBarredWithinCUG (0),
    subscriberNotMemberOfCUG (1),
    requestedBasicServiceViolatesCUG-Constraints (5),
    calledPartySS-InteractionViolation (7)}

SS-IncompatibilityCause ::= SEQUENCE {
    ss-Code [1] SS-Code OPTIONAL,
    basicService BasicServiceCode OPTIONAL,
    ss-Status [4] SS-Status OPTIONAL,
    ...}

PW-RegistrationFailureCause ::= ENUMERATED {
    undetermined (0),
    invalidFormat (1),
    newPasswordsMismatch (2)}

SM-EnumeratedDeliveryFailureCause ::= ENUMERATED {
    memoryCapacityExceeded (0),
    equipmentProtocolError (1),
    equipmentNotSM-Equipped (2),
    unknownServiceCentre (3),
    sc-Congestion (4),
    invalidSME-Address (5),
    subscriberNotSC-Subscriber (6)}

SM-DeliveryFailureCause ::= SEQUENCE {
    sm-EnumeratedDeliveryFailureCause SM-EnumeratedDeliveryFailureCause,
    diagnosticInfo SignalInfo OPTIONAL,
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

AbsentSubscriberSM-Param ::= SEQUENCE {
    absentSubscriberDiagnosticSM AbsentSubscriberDiagnosticSM OPTIONAL,
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

AbsentSubscriberDiagnosticSM ::= INTEGER (0..255)
    -- AbsentSubscriberDiagnosticSM values are defined in ETS 300 536 (GSM 03.40)

SystemFailureParam ::= CHOICE {
    networkResource NetworkResource,
    -- networkResource must not be used in version 3
    extensibleSystemFailureParam ExtensibleSystemFailureParam
    -- extensibleSystemFailureParam must not be used in version <3
    )
ExtensibleSystemFailureParam ::= SEQUENCE {
    networkResource NetworkResource OPTIONAL,
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

DataMissingParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

UnexpectedDataParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

FacilityNotSupParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

OR-NotAllowedParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

UnknownSubscriberParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

NumberChangedParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

UnidentifiedSubParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

IllegalSubscriberParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

IllegalEquipmentParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

BearerServNotProvParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

TeleservNotProvParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

TracingBufferFullParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

NoRoamingNbParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

AbsentSubscriberParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

BusySubscriberParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

NoSubscriberReplyParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

ForwardingViolationParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}

ForwardingFailedParam ::= SEQUENCE {
    extensionContainer ExtensionContainer OPTIONAL,
    ...}
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>227</td>
<td><strong>ATI-NotAllowedParam</strong> ::= SEQUENCE {</td>
<td>extensionContainer ExtensionContainer</td>
</tr>
<tr>
<td>228</td>
<td>extensionContainer ExtensionContainer OPTIONAL,</td>
<td></td>
</tr>
<tr>
<td>229</td>
<td>...}</td>
<td></td>
</tr>
<tr>
<td>230</td>
<td><strong>SubBusyForMT-SMS-Param</strong> ::= SEQUENCE {</td>
<td>extensionContainer ExtensionContainer</td>
</tr>
<tr>
<td>231</td>
<td>extensionContainer ExtensionContainer OPTIONAL,</td>
<td></td>
</tr>
<tr>
<td>232</td>
<td>...}</td>
<td></td>
</tr>
<tr>
<td>233</td>
<td><strong>MessageWaitListFullParam</strong> ::= SEQUENCE {</td>
<td>extensionContainer ExtensionContainer</td>
</tr>
<tr>
<td>234</td>
<td>extensionContainer ExtensionContainer OPTIONAL,</td>
<td></td>
</tr>
<tr>
<td>235</td>
<td>...}</td>
<td></td>
</tr>
<tr>
<td>236</td>
<td>END</td>
<td></td>
</tr>
</tbody>
</table>
14.7.8 Common data types

```plaintext
MAP-CommonDataTypes {
    ccitt identified-organization (4) etsi (0) mobileDomain (0)
    gsm-Network (1) modules (3) map-CommonDataTypes (18) version3 (3)}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS

-- general data types and values
AddressString,
ISDN-AddressString,
ISDN-SubaddressString,
ExternalSignalInfo,
SignalInfo,
maxSignalInfoLength,

-- data types for numbering and identification
IMSI,
TMSI,
SubscriberId,
IMEI,
HLR-List,
LMSI,
GlobalCellId,
NetworkResource,

-- data types for CAMEL
CellIdOrLAI,

-- data types for subscriber management
BasicServiceCode,
Ext-BasicServiceCode

; }

IMPORTS

TeleserviceCode,
Ext-TeleserviceCode

FROM MAP-TS-Code {
    ccitt identified-organization (4) etsi (0) mobileDomain (0)
    gsm-Network (1) modules (3) map-TS-Code (19) version3 (3)}

BearerServiceCode,
Ext-BearerServiceCode

FROM MAP-BS-Code {
    ccitt identified-organization (4) etsi (0) mobileDomain (0)
    gsm-Network (1) modules (3) map-BS-Code (20) version3 (3)}

ExtensionContainer

FROM MAP-ExtensionDataTypes {
    ccitt identified-organization (4) etsi (0) mobileDomain (0)
    gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}

;

-- general data types

TBCD-STRING ::= OCTET STRING

-- This type (Telephony Binary Coded Decimal String) is used to
-- represent several digits from 0 through 9, *, #, a, b, c, two
-- digits per octet, each digit encoded 0000 to 1001 (0 to 9),
-- 1010 (*), 1011 (#), 1100 (a), 1101 (b) or 1110 (c); 1111 used
-- as filler when there is an odd number of digits.
-- bits 8765 of octet n encoding digit 2n
-- bits 4321 of octet n encoding digit 2(n-1) +1
```
AddressString ::= OCTET STRING (SIZE (1..maxAddressLength))
-- This type is used to represent a number for addressing purposes. It is composed of:
-- a) one octet for nature of address, and numbering plan indicator.
-- b) digits of an address encoded as TBCD-String.
-- a) The first octet includes a one bit extension indicator, a
-- 3 bits nature of address indicator and a 4 bits numbering plan indicator, encoded as follows:
-- bit 8: 1 (no extension)
-- bits 765: nature of address indicator
-- 000 unknown
-- 001 international number
-- 010 national significant number
-- 011 network specific number
-- 100 subscriber number
-- 101 reserved
-- 110 abbreviated number
-- 111 reserved for extension
-- bits 4321: numbering plan indicator
-- 0000 unknown
-- 0001 ISDN/Telephony Numbering Plan (Rec CCITT E.164)
-- 0010 spare
-- 0011 data numbering plan (CCITT Rec X.121)
-- 0100 telex numbering plan (CCITT Rec F.69)
-- 0101 spare
-- 0110 land mobile numbering plan (CCITT Rec E.212)
-- 0111 spare
-- 1000 national numbering plan
-- 1001 private numbering plan
-- 1111 reserved for extension
-- all other values are reserved.
-- b) The following octets representing digits of an address encoded as a TBCD-STRING.

maxAddressLength INTEGER ::= 20

ISDN-AddressString ::= AddressString (SIZE (1..maxISDN-AddressLength))
-- This type is used to represent ISDN numbers.

maxISDN-AddressLength INTEGER ::= 9
ISDN-SubaddressString ::= OCTET STRING (SIZE (1..maxISDN-SubaddressLength))

-- This type is used to represent ISDN subaddresses.
-- It is composed of:
--  a) one octet for type of subaddress and odd/even indicator.
--  b) 20 octets for subaddress information.
-- a) The first octet includes a one bit extension indicator, a
--  3 bits type of subaddress and a one bit odd/even indicator,
--  encoded as follows:
--  bit 8: 1 (no extension)
--  bits 765: type of subaddress
--  000  NSAP (X.213/ISO 8348 AD2)
--  010  User Specified
--  All other values are reserved
-- bit 4: odd/even indicator
--  0  even number of address signals
--  1  odd number of address signals
--  The odd/even indicator is used when the type of subaddress
-- is "user specified" and the coding is BCD.
-- bits 321: 000 (unused)
-- b) Subaddress information.
--  The NSAP X.213/ISO8348AD2 address shall be formatted as specified
-- by octet 4 which contains the Authority and Format Identifier
-- (AFI). The encoding is made according to the "preferred binary
-- encoding" as defined in X.213/ISO834AD2. For the definition
-- of this type of subaddress, see CCITT Rec I.334.
-- For User-specific subaddress, this field is encoded according
-- to the user specification, subject to a maximum length of 20
-- octets. When interworking with X.25 networks BCD coding should
-- be applied.

maxISDN-SubaddressLength INTEGER ::= 21

ExternalSignalInfo ::= SEQUENCE {
  protocolId ProtocolId,
  signalInfo SignalInfo,
  -- Information about the internal structure is given in
  -- subclause 5.6.9.
  extensionContainer ExtensionContainer OPTIONAL,
  -- extensionContainer must not be used in version 2
  ...}

SignalInfo ::= OCTET STRING (SIZE (1..maxSignalInfoLength))

maxSignalInfoLength INTEGER ::= 200

-- This NamedValue represents the theoretical maximum number of
-- octets which are available to carry a single data type,
-- without requiring segmentation to cope with the network layer
-- service. However, the actual maximum size available for a data
-- type may be lower, especially when other information elements
-- have to be included in the same component.

ProtocolId ::= ENUMERATED {
  gsm-0408  (1),
  gsm-0806  (2),
  gsm-BSSMAP (3),
  -- Value 3 is reserved and must not be used
  ets-300102-1 (4)}

-- data types for numbering and identification

IMSI ::= TBCD-STRING (SIZE (3..8))
-- digits of MCC, MNC, MSIN are concatenated in this order.

TMSI ::= OCTET STRING (SIZE (1..4))

SubscriberId ::= CHOICE {
  imsi [0] IMSI,
  tmsi [1] TMSI}
IMEI ::= TBCD-STRING (SIZE (8))
   -- Refers to International Mobile Station Equipment Identity
   -- and Software Version Number (SVN) defined in TS GSM 03.03.
   -- If the SVN is not present the last octet shall contain the
eight digit 0 and a filler.
   -- If present the SVN shall be included in the last octet.

HLR-Id ::= IMSI
   -- leading digits of IMSI, i.e. (MCC, MNC, leading digits of
   -- MSIN) forming HLR Id defined in TS GSM 03.03.

HLR-List ::= SEQUENCE SIZE (1..maxNumOfHLR-Id) OF
   HLR-Id

maxNumOfHLR-Id INTEGER ::= 50

IMSI ::= OCTET STRING (SIZE (4))

GlobalCellId ::= OCTET STRING (SIZE (.7))
   -- Refers to Cell Global Identification defined in TS GSM 03.03.
   -- Octets are coded according to TS GSM 04.08.
   -- The internal structure is defined as follows:
   -- Mobile Country Code: 3 digits according to CCITT Rec E.212
   -- Mobile Network Code: 2 digits according to CCITT Rec E.212
   -- Location Area Code: 2 octets according to TS GSM 04.08
   -- Cell Identity: 2 octets (CI) according to TS GSM 04.08

NetworkResource ::= ENUMERATED {
   plmn (0),
   hlr (1),
   vlr (2),
   pvlr (3),
   controllingMSC (4),
   vmsc (5),
   elr (6),
   rrs (7)}

   -- data types for CAMEL

CellIdOrLAI ::= CHOICE {
   cellIdFixedLength [0] CellIdFixedLength,
   laiFixedLength [1] LAIFixedLength}

CellIdFixedLength ::= OCTET STRING (SIZE (7))
   -- Refers to Cell Global Identification defined in TS GSM 03.03.
   -- Octets are coded according to TS GSM 04.08.
   -- The internal structure is defined as follows:
   -- Mobile Country Code: 3 digits according to CCITT Rec E.212
   -- Mobile Network Code: 2 digits according to CCITT Rec E.212
   -- Location Area Code: 2 octets according to TS GSM 04.08
   -- Cell Identity: 2 octets (CI) according to TS GSM 04.08

LAIFixedLength ::= OCTET STRING (SIZE (5))
   -- Refers to Location Area Identification defined in TS GSM 03.03.
   -- Octets are coded according to TS GSM 04.08.
   -- The internal structure is defined as follows:
   -- Mobile Country Code: 3 digits according to CCITT Rec E.212
   -- Mobile Network Code: 2 digits according to CCITT Rec E.212
   -- Location Area Code: 2 octets according to TS GSM 04.08

   -- data types for subscriber management

BasicServiceCode ::= CHOICE {
   bearerService [2] BearerServiceCode,

Ext-BasicServiceCode ::= CHOICE {
   ext-BearerService [2] Ext-BearerServiceCode,
14.7.9 Teleservice Codes

MAP-TS-Code

-- ccitt identified-organization (4) etsi (0) mobileDomain (0)
gsm-Network (1) modules (3) map-TS-Code (19) version3 (3)

DEFINITIONS

::=

BEGIN

TeleserviceCode ::= OCTET STRING (SIZE (1))
-- This type is used to represent the code identifying a single
-- teleservice, a group of teleservices, or all teleservices. The
-- services are defined in TS GSM 02.03.
-- The internal structure is defined as follows:
-- bits 87654321: group (bits 8765) and specific service
-- (bits 4321)

Ext-TeleserviceCode ::= OCTET STRING (SIZE (1..5))
-- This type is used to represent the code identifying a single
-- teleservice, a group of teleservices, or all teleservices. The
-- services are defined in TS GSM 02.03.
-- The internal structure is defined as follows:
-- OCTET 1:
-- bits 87654321: group (bits 8765) and specific service
-- (bits 4321)
-- OCTETS 2-5: reserved for future use. If received the
-- Ext-TeleserviceCode shall be
-- treated according to the exception handling defined for the
-- operation that uses this type.
-- Ext-TeleserviceCode includes all values defined for TeleserviceCode.

allTeleservices  TeleserviceCode ::= '00000000'B

allSpeechTransmissionServices TeleserviceCode ::= '00010000'B
telephony          TeleserviceCode ::= '00010001'B
emergencyCalls     TeleserviceCode ::= '00010010'B

allShortMessageServices TeleserviceCode ::= '00100000'B
shortMessageMT-PP    TeleserviceCode ::= '00100001'B
shortMessageMO-PP    TeleserviceCode ::= '00100010'B

allFacsimileTransmissionServices TeleserviceCode ::= '01100000'B
facsimileGroup3AndAlterSpeech TeleserviceCode ::= '01100001'B
automaticFacsimileGroup3  TeleserviceCode ::= '01100010'B
facsimileGroup4        TeleserviceCode ::= '01100011'B

-- The following non-hierarchical Compound Teleservice Groups
-- are defined in TS GSM 02.30:
-- covers Teleservice Groups 'allFacsimileTransmissionServices'
-- and 'allShortMessageServices'
allDatTeleservices    TeleserviceCode ::= '01110000'B

allTeleservices-ExceptSMS TeleserviceCode ::= '10000000'B
-- covers Teleservice Groups 'allSpeechTransmissionServices' and
-- 'allFacsimileTransmissionServices'

-- Compound Teleservice Group Codes are only used in call
-- independent supplementary service operations, i.e. they
-- are not used in InsertSubscriberData or in
-- DeleteSubscriberData messages.

allVoiceGroupCallServices TeleserviceCode ::= '10010000'B
voiceGroupCall        TeleserviceCode ::= '10010001'B
voiceBroadcastCall    TeleserviceCode ::= '10010010'B
<table>
<thead>
<tr>
<th>PLMN-specificTS</th>
<th>TeleserviceCode</th>
</tr>
</thead>
<tbody>
<tr>
<td>allPLMN-specificTS</td>
<td>'11010000'B</td>
</tr>
<tr>
<td>plmn-specificTS-1</td>
<td>'11010001'B</td>
</tr>
<tr>
<td>plmn-specificTS-2</td>
<td>'11010010'B</td>
</tr>
<tr>
<td>plmn-specificTS-3</td>
<td>'11010011'B</td>
</tr>
<tr>
<td>plmn-specificTS-4</td>
<td>'11010100'B</td>
</tr>
<tr>
<td>plmn-specificTS-5</td>
<td>'11010101'B</td>
</tr>
<tr>
<td>plmn-specificTS-6</td>
<td>'11010110'B</td>
</tr>
<tr>
<td>plmn-specificTS-7</td>
<td>'11010111'B</td>
</tr>
<tr>
<td>plmn-specificTS-8</td>
<td>'11011000'B</td>
</tr>
<tr>
<td>plmn-specificTS-9</td>
<td>'11011001'B</td>
</tr>
<tr>
<td>plmn-specificTS-A</td>
<td>'11011010'B</td>
</tr>
<tr>
<td>plmn-specificTS-B</td>
<td>'11011011'B</td>
</tr>
<tr>
<td>plmn-specificTS-C</td>
<td>'11011100'B</td>
</tr>
<tr>
<td>plmn-specificTS-D</td>
<td>'11011101'B</td>
</tr>
<tr>
<td>plmn-specificTS-E</td>
<td>'11011110'B</td>
</tr>
<tr>
<td>plmn-specificTS-F</td>
<td>'11011111'B</td>
</tr>
</tbody>
</table>
14.7.10 Bearer Service Codes

MAP-BS-Code

```plaintext
BEGIN

BearerServiceCode ::= OCTET STRING (SIZE (1))
-- This type is used to represent the code identifying a single
-- bearer service, a group of bearer services, or all bearer
-- services. The services are defined in TS GSM 02.02.
-- The internal structure is defined as follows:
-- plmn-specific bearer services:
-- bits 87654321: defined by the HPLMN operator
-- rest of bearer services:
-- bit 8: 0 (unused)
-- bits 7654321: group (bits 7654), and rate, if applicable
-- (bits 321)
```

Ext-BearerServiceCode ::= OCTET STRING (SIZE (1..5))
-- This type is used to represent the code identifying a single
-- bearer service, a group of bearer services, or all bearer
-- services. The services are defined in TS GSM 02.02.
-- The internal structure is defined as follows:
-- OCTET 1:
-- plmn-specific bearer services:
-- bits 87654321: defined by the HPLMN operator
-- rest of bearer services:
-- bit 8: 0 (unused)
-- bits 7654321: group (bits 7654), and rate, if applicable
-- (bits 321)
-- OCTETS 2-5: reserved for future use. If received the
-- Ext-TeleServiceCode shall be
-- treated according to the exception handling defined for the
-- operation that uses this type.
-- Ext-BearerServiceCode includes all values defined for BearerServiceCode.

allBearerServices BearerServiceCode ::= '00000000'B
allDataCDA-Services BearerServiceCode ::= '00010000'B
dataCDA-300bps BearerServiceCode ::= '00010001'B
dataCDA-1200bps BearerServiceCode ::= '00010010'B
dataCDA-2400bps BearerServiceCode ::= '00010011'B
dataCDA-4800bps BearerServiceCode ::= '00010100'B
dataCDA-9600bps BearerServiceCode ::= '00010101'B
general-dataCDA BearerServiceCode ::= '00010110'B
allDataCDS-Services BearerServiceCode ::= '00011000'B
dataCDS-1200bps BearerServiceCode ::= '00011010'B
dataCDS-2400bps BearerServiceCode ::= '00011100'B
dataCDS-4800bps BearerServiceCode ::= '00011101'B
dataCDS-9600bps BearerServiceCode ::= '00011110'B
general-dataCDS BearerServiceCode ::= '00011111'B
allPadAccessCA-Services BearerServiceCode ::= '00100000'B
padAccessCA-300bps BearerServiceCode ::= '00100001'B
padAccessCA-1200bps BearerServiceCode ::= '00100010'B
padAccessCA-1200-75bps BearerServiceCode ::= '00100011'B
padAccessCA-2400bps BearerServiceCode ::= '00100100'B
padAccessCA-4800bps BearerServiceCode ::= '00100101'B
padAccessCA-9600bps BearerServiceCode ::= '00100110'B
general-padAccessCA BearerServiceCode ::= '00100111'B
allDataPDS-Services BearerServiceCode ::= '00101000'B

dataPDS-2400bps BearerServiceCode ::= '00101100'B

dataPDS-4800bps BearerServiceCode ::= '00101101'B

dataPDS-9600bps BearerServiceCode ::= '00101110'B

general-dataPDS BearerServiceCode ::= '00101111'B

allAlternateSpeech-DataCDA BearerServiceCode ::= '00110000'B

allAlternateSpeech-DataCDS BearerServiceCode ::= '00111000'B

allSpeechFollowedByDataCDA BearerServiceCode ::= '01000000'B

allSpeechFollowedByDataCDS BearerServiceCode ::= '01001000'B

-- The following non-hierarchical Compound Bearer Service
-- Groups are defined in TS GSM 02.30:

allDataCircuitAsynchronous BearerServiceCode ::= '01010000'B

-- covers "allDataCDA-Services", "allAlternateSpeech-DataCDA" and

allAsyncronousServices BearerServiceCode ::= '01100000'B

-- covers "allDataCDA-Services", "allAlternateSpeech-DataCDA", "allSpeechFollowedByDataCDA" and "allPadAccessCDA-Services"

allDataCircuitSynchronous BearerServiceCode ::= '01011000'B

-- covers "allDataCDS-Services", "allAlternateSpeech-DataCDS" and

allSynchronousServices BearerServiceCode ::= '01101000'B

-- covers "allDataCDS-Services", "allAlternateSpeech-DataCDS", "allSpeechFollowedByDataCDS" and "allDataPDS-Services"

-- Compound Bearer Service Group Codes are only used in call
-- independent supplementary service operations, i.e. they
-- are not used in InsertSubscriberData or in
-- DeleteSubscriberData messages.

allPLMN-specificBS BearerServiceCode ::= '11010000'B

plmn-specificBS-1 BearerServiceCode ::= '11010001'B

plmn-specificBS-2 BearerServiceCode ::= '11010010'B

plmn-specificBS-3 BearerServiceCode ::= '11010011'B

plmn-specificBS-4 BearerServiceCode ::= '11010100'B

plmn-specificBS-5 BearerServiceCode ::= '11010101'B

plmn-specificBS-6 BearerServiceCode ::= '11010110'B

plmn-specificBS-7 BearerServiceCode ::= '11010111'B

plmn-specificBS-8 BearerServiceCode ::= '11011000'B

plmn-specificBS-9 BearerServiceCode ::= '11011001'B

plmn-specificBS-A BearerServiceCode ::= '11011010'B

plmn-specificBS-B BearerServiceCode ::= '11011011'B

plmn-specificBS-C BearerServiceCode ::= '11011100'B

plmn-specificBS-D BearerServiceCode ::= '11011101'B

plmn-specificBS-E BearerServiceCode ::= '11011110'B

plmn-specificBS-F BearerServiceCode ::= '11011111'B

END
14.7.11 Extension data types

MAP-ExtensionDataTypes { ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3) map-ExtensionDataTypes (21) version3 (3)}

DEFINITIONS

IMPLICIT TAGS

::=

BEGIN

EXPORTS

PrivateExtension,
ExtensionContainer;

-- IOC for private MAP extensions

MAP-EXTENSION ::= CLASS {
  &ExtensionType OPTIONAL,
  &extensionId OBJECT IDENTIFIER )
-- The length of the Object Identifier shall not exceed 16 octets and the
-- number of components of the Object Identifier shall not exceed 16

-- data types

ExtensionContainer ::= SEQUENCE {
  privateExtensionList [0]PrivateExtensionList OPTIONAL,
  pcs-Extensions [1]PCS-Extensions OPTIONAL,
  ...}

PrivateExtensionList ::= SEQUENCE SIZE (1..maxNumOfPrivateExtensions) OF PrivateExtension

PrivateExtension ::= SEQUENCE {
  extId MAP-EXTENSION.&extensionId
  ((ExtensionSet)),
  extType MAP-EXTENSION.&ExtensionType
  ((ExtensionSet){@extId}) OPTIONAL}

maxNumOfPrivateExtensions INTEGER ::= 10

ExtensionSet MAP-EXTENSION ::= {
  (...,
  -- ExtensionSet is the set of all defined private extensions
  )
-- Unsupported private extensions shall be discarded if received.

PCS-Extensions ::= SEQUENCE {
  ...}

END
15 General on MAP user procedures

15.1 Introduction

Clauses 15 to 21 describe the use of MAP services for GSM signalling procedures. GSM signalling procedures may involve one or several interfaces running one or several application protocols. This ETS addresses only the signalling procedures which require at least the use of one MAP service.

When a signalling procedure takes place in the network, an application process invocation is created in each system component involved. Part of the application process invocation acts as a MAP user and handles one or several MAP dialogues. For each dialogue it employs an instance of the MAP service provider. It may also use other communication services to exchange information on other interfaces, but detailed description of these aspects is outside the scope of this ETS.

15.2 Common aspects of user procedure descriptions

15.2.1 General conventions

For each signalling procedure this ETS provides a brief textual overview accompanied by a flow diagram which represent the functional interactions between system components. Functional interactions are labelled using the MAP service name when the interaction results from a service request or by this service name followed by the symbol "ack" when this interaction results from a service response.

For each of the system components involved, this ETS also provides a detailed textual description of the application process behaviour as well as an SDL diagram. SDL diagrams describe the sequence of events, as seen by the MAP-User, which occurs at MAP service provider boundaries as well as external events which occur at other interfaces and which impact on the previous sequence.

External events do not necessarily correspond to the messages of other protocols used in the system component. The MAP-user procedures are described as if a set of interworking functions (IWF) between the MAP-user and the other protocol entities was implemented (see figure 15.2/1). Such interworking functions are assumed to perform either an identity mapping or some processing or translation as required to eliminate information irrelevant to the MAP-user.

The mapping of service primitives on to protocol elements is described in clauses 11 to 14.

GSM signalling procedures are built from one or more sub-procedures (e.g. authentication, ciphering, ...). Sub-procedures from which signalling procedures are built are represented using SDL MACRO descriptions.

In case of any discrepancy between the textual descriptions and the SDL descriptions, the latter take precedence.

15.2.2 Naming conventions

Events related to MAP are represented by MAP service primitives. The signal names used in the SDL diagrams are derived from the service primitive names defined in clauses 5 to 10, with some lexical transformations for readability and parsability purposes (blanks between words are replaced by underscores, the first letter of each word is capitalized).

Events received and sent on other interfaces are named by appending the message or signal name to a symbol representing the interface type, with some lexical transformations for readability and parsability purposes (blanks between words are replaced by underscores, the first letter of each word is capitalized).
The following symbols are used to represent the interface types:

"I": For interfaces to the fixed network. "I" stands for ISUP interface.

"A": For interfaces to BSS (i.e. A-interfaces);

"OM": For network management interfaces (communication with OMC, MML interface, ...);

"SC": For interfaces to a Service Centre;

"HO_CA": For internal interfaces to the Handover Control Application.

"US": For a local USSD application.

These naming conventions can be summarized by the following BNF description:

\[
\text{<Event\_Name> ::= <MAP\_Primitive> | <External\_Event>}
\]

\[
\text{<MAP\_Primitive> ::= <MAP\_Open> | <MAP\_Close> | <MAP\_U\_Abort> | <MAP\_P\_Abort> |}
\]

\[
\text{<MAP\_Specific> | <MAP\_Notice>}
\]

\[
\text{<MAP\_Open> ::= MAP\_Open\_Req | MAP\_Open\_Ind | MAP\_Open\_Rsp | MAP\_Open\_Cnf}
\]

\[
\text{<MAP\_Close> ::= MAP\_Close\_Req | MAP\_Close\_Ind}
\]

\[
\text{<MAP\_U\_Abort> ::= MAP\_U\_Abort\_Req | MAP\_U\_Abort\_Ind}
\]

\[
\text{<MAP\_P\_Abort> ::= MAP\_P\_Abort\_Ind}
\]

\[
\text{<MAP\_Notice> ::= MAP\_Notice\_Ind}
\]

\[
\text{<MAP\_Specific> ::= <MAP\_Req> | <MAP\_Ind> | <MAP\_Rsp> | <MAP\_Cnf>}
\]

\[
\text{<MAP\_Req> ::= MAP\_<Service\_Name>_Req}
\]

\[
\text{<MAP\_Ind> ::= MAP\_<Service\_Name>_Ind}
\]

\[
\text{<MAP\_Rsp> ::= MAP\_<Service\_Name>_Rsp}
\]

\[
\text{<MAP\_Cnf> ::= MAP\_<Service\_Name>_Cnf}
\]

\[
\text{<External\_Event> ::= <Interface\_Type>_<External\_Signal>}
\]

\[
\text{<Interface\_Type> ::= I | A | OM | SC | HO AC | US}
\]

\[
\text{<External\_Signal> ::= <Lexical\_Unit>}
\]

\[
\text{<Service\_Name> ::= <Lexical\_Unit>}
\]

\[
\text{<Lexical\_Unit> ::= <Lexical\_Component> | <Lexical\_Unit>_ <Lexical\_Component>}
\]

\[
\text{<Lexical\_Component> ::= <Upper\_Case\_Letter><Letter\_Or\_Digit\_List>}
\]

\[
\text{<Letter\_Or\_Digit\_List> ::= <Letter\_Or\_Digit> | <Letter\_Or\_Digit\_List><Letter\_Or\_Digit>}
\]

\[
\text{<Letter\_Or\_Digit> ::= <Letter> | <Digit>}
\]

\[
\text{<Letter> ::= <Lower\_Case\_Letter> | <Upper\_Case\_Letter>}
\]

\[
\text{<Upper\_Case\_Letter> ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z}
\]

\[
\text{<Lower\_Case\_Letter> ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z}
\]

\[
\text{<Digit> ::= 1|2|3|4|5|6|7|8|9|0}
\]

ETS
15.2.3 Convention on primitives parameters

15.2.3.1 Open service

When the originating and destination reference parameters shall be included in the MAP-OPEN request primitive, their value are indicated as a comment to the signal which represents this primitive.

15.2.3.2 Close service

When a pre-arranged released is requested, a comment is attached to the signal which represents the MAP-CLOSE request primitive. In the absence of comment, a normal release is assumed.

15.2.4 Version handling at dialogue establishment

Unless explicitly indicated in subsequent subclauses, the following principles regarding version handling procedures at dialogue establishment are applied by the MAP-user:

15.2.4.1 Behaviour at the initiating side

When a MAP user signalling procedure has to be executed, the MAP-user issues a MAP-OPEN request primitive with an appropriate application-context-name. If several names are supported (i.e. several versions) a suitable one is selected using the procedures described in clause 3.

If version 2 is selected and a MAP-OPEN Confirm primitive in response to the MAP-OPEN request is received with a result parameter set to "refused" and a diagnostic parameter indicating "application-context-not-supported" or "potential incompatibility problem", the MAP-User issues a new MAP-OPEN request primitive with the equivalent version one context. This is informally represented in the SDL diagrams by a task symbol indicating "Perform Vr procedure".

If version 3 is selected and a MAP-OPEN Confirm primitive in response to the MAP-OPEN request is received with a result parameter set to "refused" and a diagnostic parameter indicating "application-context-not-supported" or "potential incompatibility problem", the MAP-User issues a new MAP-OPEN request primitive with the equivalent version one or version two context. This is informally represented in the SDL diagrams by task symbols indicating "Perform Vr procedure".

15.2.4.2 Behaviour at the responding side

On receipt of a MAP-OPEN indication primitive, the MAP-User analyses the application-context-name.
If it refers to a version one context, the associated V1 procedure is executed; if it refers to a version two context, the associated V2 procedure is executed, otherwise the associated V3 procedure is executed.

### 15.2.5 Abort Handling

Unless explicitly indicated in subsequent subclauses, the following principles are applied by the MAP-user regarding abort handling procedures.

On receipt of a MAP-P-ABORT indication or MAP-U-ABORT Indication primitive from any MAP-provider invocation, the MAP-User issues a MAP-U-ABORT Request primitive to each MAP-provider invocation associated with the same user procedure.

If applicable a decision is made to decide if the affected user procedure has to be retried or not.

### 15.2.6 SDL conventions

The MAP SDLs make use of a number of SDL concepts and conventions, where not all of them may be widely known. Therefore, this subclause outlines the use of a few concepts and conventions to improve understanding of the MAP SDLs.

The MAP User SDLs make use of SDL Processes, Procedures and Macros. Processes are independent from each other even if one process starts another one: The actions of both of them have no ordering in time. SDL Procedures and Macros are just used to ease writing of the specification: They contain parts of a behaviour used in several places, and the corresponding Procedure/Macro definition has to be expanded at the position of the Procedure/Macro call.

All Processes are started at system initialization and live forever, unless process creation/termination is indicated explicitly (i.e. a process is created by some other process).

The direction of Input/Output Signals in the SDL graphs is used to indicate the entity to which/from which communication is directed. If a process A communicates in parallel with processes B and C, all Inputs/Outputs to/from B are directed to one side, whereas communication with C is directed to the other side. However, there has been no formal convention used that communication to a certain entity (e.g. a HLR) will always be directed to a certain side (e.g. right).

In each state all those Input Signals are listed, which result in an action and/or state change. If an Input Signal is not listed in a state, receipt of this input should lead to an implicit consumption without any action or state change (according to the SDL rules). This implicit consumption is mainly used for receipt of the MAP DELIMITER indication and for receipt of a MAP CLOSE indication, except for a premature MAP CLOSE.

### 15.3 Interaction between MAP Provider and MAP Users

Each MAP User is defined by at least one SDL process. On the dialogue initiating side the MAP User will create a new instance of a MAP Provider implicit by issuing a MAP-OPEN request. This instance corresponds to a TC Dialogue and lives as long as the dialogue exists (see also subclause 11.3). There is a fix relation between MAP User and this Provider instance, i.e. all MAP service primitives from the MAP User for this dialogue are sent to this instance and all TC components received by this MAP Provider are mapped onto service primitives sent to this MAP User.

On the receiving side a MAP Provider instance is created implicit by receipt of a TC BEGIN indication. The corresponding MAP User is determined by the Application Context name included in this primitive, i.e. each Application Context is associated with one and only one MAP User. An instance of this User will be created implicit by receiving a MAP-OPEN indication. Note that in some cases there exist several SDL Processes for one MAP User (Application Context), e.g. the processes Register_SS_HLR, Erase_SS_HLR, Activate_SS_HLR, Deactivate_SS_HLR, Interrogate_SS_HLR, and Register>Password for the AC Network_Functional_SS_Handling. In these cases, a coordinator process is introduced acting as a MAP User, which in turn starts a sub-process depending on the first MAP service primitive received.
16 Mobility procedures

16.1 Location management Procedures

This subclause comprises a number of processes to handle the mobile nature of the subscriber. The processes will be addressed by SCCP Sub-System Number (MSC, VLR or HLR) and the Application Context. The following processes are defined in this subclause:

- **Process Update Location Area:**
  - Initiator: Update_Location_Area_MSC, subclause 16.1.1.2;
  - Responder: Update_Location_Area_VLR, subclause 16.1.1.3;

- **Process Update Location:**
  - Initiator: Update_Location_Area_VLR, subclause 16.1.1.3, or Update_Location_VLR, subclause 16.1.1.6;
  - Responder: Update_Location_HLR, subclause 16.1.1.4;

- **Process Send Identification:**
  - Initiator: Update_Location_Area_VLR, subclause 16.1.1.3;
  - Responder: Send_Identification_VLR, subclause 16.1.1.5;

- **Process Subscriber Present HLR:**
  - Initiator: Subscriber_Present_HLR, subclause 16.1.1.7;
  - Responder: Short_Message_Alert_IWMSC, subclause 20.4.3;

- **Process Cancel Location:**
  - Initiator: Cancel_Location_HLR, subclause 16.1.2.2;
  - Responder: Cancel_Location_VLR, subclause 16.1.2.3;

- **Process Detach IMSI:**
  - Initiator: Detach_IMSI_MSC, subclause 16.3.2;
  - Responder: Detach_IMSI_VLR, subclause 16.3.3.

As both the Update Location Area and the Detach IMSI processes use the same application context name, the MAP Provider cannot distinguish between them. Therefore, a Location Management Coordinator Process will act as one user for this application context. This process (one in MSC, one in VLR) will create the Update Location Area or the Detach IMSI process, depending on the first service primitive received in the respective dialogue.

Additionally, a Location Management Coordinator process in the HLR coordinates the two application processes "Update Location HLR" (subclause 16.1.1.4) and "RESTORE_DATA_HLR" (subclause 16.3.3) that are addressed by the same application context.

**Location Management Coordinator MSC**

On receipt of a request for location updating from the A-interface, the Location Management Coordinator in the MSC will:

- create the process Update_Location_Area_MSC in case the updating type indicated in the A-interface primitive indicates normal updating, periodic updating or IMSI Attach;
- create the process Detach_IMSI_MSC in case the updating type indicated in the A-interface primitive indicates IMSI Detach.

The respective primitive is then forwarded to the created process. Henceforth, the coordinator will relay all service primitives from provider to the user and vice versa, until a request or indication for dialogue termination is received. This last primitive will be relayed, too, before the Coordinator process returns to idle state.

**Location Management Coordinator VLR**

On receipt of a dialogue request for the Location Management Application Context (see Receive_Open_Ind macro in subclause 21.1), the Location_Management_Coordinator will:

- terminate the procedure in case of parameter problems or if the MSC indicated version Vr protocol; or
- continue as below, if the dialogue is accepted.

Depending on the first service primitive received from the MAP Provider in this dialogue, the user process is created:

- Update_Location_Area_VLR in case the primitive is a MAP_UPDATE_LOCATION_AREA indication;
- Detach_IMSI_VLR in case the primitive is a MAP_DETACH IMSI indication.

In case a MAP_U_ABORT, MAP_P_ABORT or a premature MAP_CLOSE indication is received instead, the process returns to idle state. If a MAP_NOTICE indication is received, the dialogue towards the MSC is aborted and the process returns to idle state.

After creation of the user process the service primitive received from the provider is passed to the user process. Henceforth, the coordinator will relay all service primitives from provider to the user and vice versa, until a request or indication for dialogue termination is received. This last primitive will be relayed, too, before the Coordinator process returns to idle state.

**Location Management Coordinator HLR**

On receipt of a dialogue request for the Location Management Application Context (see Receive_Open_Ind macro in subclause 21.1), the Location_Management_Coordinator will:

- terminate the process in case of parameter problems; or
- revert to MAP version Vr protocol if the VLR requests version Vr protocol; or
- continue as described in the following, if the dialogue is accepted.

The user process is created depending on the first service primitive received from the MAP service provider within this dialogue:

- Update_Location_HLR if the primitive is a MAP_UPDATE_LOCATION indication;
- RESTORE_DATA_HLR if the primitive is a MAP_RESTORE_DATA indication.

If a MAP_NOTICE indication is received instead, the dialogue towards the MSC is terminated and the process returns to idle state.

After creation of the user process the service primitive received from the MAP service-provider is passed to the user process. Henceforth, the coordinator will relay all service primitives from MAP service-provider to the MAP service-user and vice versa, until a request or indication for dialogue termination is received. This last primitive will be relayed, too, before the Coordinator process returns to idle state.
Figure 16.1/1: Process Location_Management_Coordinator_MSC
Figure 16.1/2: Process Location_Management_Coordinator_VLR
Process Location_Management_Coordinator_HLR

Figure 16.1/3: Location management coordination process in the HLR

Figure 21.1/1: Figure 16.1/3: Process Location_Management_Coordinator_HLR
16.1.1 Location updating

16.1.1.1 General

The location updating procedure is used to update the location information held in the network. This location information is used to route incoming calls, short messages and unstructured supplementary service data to the roaming subscriber. Additionally, this procedure is used to provide the VLR with the information that a subscriber already registered, but being detached, is reachable again (IMSI Attach, see GSM 03.12). The use of this Detach / Attach feature is optional for the network operator.

To minimize the updates of the subscriber’s HLR, the HLR holds only information about the VLR and MSC the subscriber is attached to. The VLR contains more detailed location information, i.e. the location area the subscriber is actually roaming in. Therefore, the VLR needs to be updated at each location area change (see figure 16.1.1/1 for this procedure), whereas the HLR needs updating only in the following cases:

- when the subscriber registers in a new VLR, i.e. the VLR has no data for that subscriber;
- when the subscriber registers in a new location area of the same VLR and new routing information is to be provided to the HLR (change of MSC area);
- if the indicator "Confirmed by HLR" or the indicator "Location Information Confirmed in HLR" is set to "Not Confirmed" because of HLR or VLR restoration, and the VLR receives an indication that the subscriber is present.

If a mobile subscriber registers in a visitor location register (VLR) not holding any information about this subscriber and is identified by a temporary mobile subscriber identity (TMSI) allocated by a previous visitor location register (PVLR), if the PVLR identity can be derived from LAI the new VLR must obtain the IMSI from PVLR to identify the HLR to be updated (see figure 16.1.1/2). If the IMSI cannot be retrieved from PVLR, it is requested from the MS (see figure 16.1.1/3).

The following MAP services are invoked by the location update procedure:

- **MAP_UPDATE_LOCATION_AREA** (see subclause 6.1);
- **MAP_UPDATE_LOCATION** (see subclause 6.1);
- **MAP_CANCEL_LOCATION** (see subclause 6.1);
- **MAP_INSERT_SUBSCRIBER_DATA** (see subclause 6.8);
- **MAP_SEND_IDENTIFICATION** (see subclause 6.1);
- **MAP_PROVIDE_IMSI** (see subclause 6.9);
- **MAP_AUTHENTICATE** (see subclause 6.5);
- **MAP_SET_CIPHERING_MODE** (see subclause 6.6);
- **MAP_FORWARD_NEW_TMSI** (see subclause 6.9);
- **MAP_CHECK_IMEI** (see subclause 6.7);
- **MAP_ACTIVATE_TRACE_MODE** (see subclause 7.2);
- **MAP_TRACE_SUBSCRIBER_ACTIVITY** (see subclause 7.2).
NOTE 1: For details of the procedure on the radio path, see GSM 04.08. The services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: Optional services are printed in italics.

Figure 16.1.1/1: Interface and services for location updating when roaming within an visitor location registers area (without need to update HLR)
Figure 16.1.1/2: Interface and services for location updating when changing the VLR area

NOTE: The optional procedures in figure 16.1.1/1 apply here respectively.
NOTE: The optional procedures in figure 16.1.1/1 apply here respectively.

Figure 16.1.1/3: Interface and services for location updating involving both a VLR and an HLR, when IMSI cannot be retrieved from the previous VLR.
16.1.1.2 Detailed procedure in the MSC

Figure 16.1.1/4 shows the MSC process for location register updating, containing macro calls for:

- Receive_Open_Cnf subclause 21.1;
- Authenticate_MSC subclause 21.5;
- Check_IMEI_MSC subclause 21.6;
- Obtain_IMSI_MSC subclause 21.8;
- Trace_Subscriber_Activity_MSC subclause 21.9.

For structuring purposes, the second part of the process is placed into the macro Update Location Completion MSC, which is specific to this process (see figure 16.1.1/5).

When the MSC receives an A_LU_REQUEST (normal location updating, periodic location updating or IMSI attach) for a subscriber via the radio path, the MSC opens a dialogue to the VLR (MAP_OPEN request without any user specific parameters) and sends a MAP_UPDATE_LOCATION_AREA request, containing the parameters provided in the A_LU_REQUEST by the MS or BSS (for the parameter mapping see GSM 09.10).

If the dialogue is rejected or the VLR indicates a fallback to the version Vr procedure (see Receive_Open_Cnf macro in subclause 21.1), the MSC will send an A_LU_Rej towards the MS and terminate the procedure.

If the dialogue is accepted, the VLR will process this updating request, invoking optionally the MAP_PROVIDE_IMSI, MAP_TRACE_SUBSCRIBER_ACTIVITY, MAP_CHECK_IMEI or the MAP_AUTHENTICATE services first (see subclause 16.1.1.3 for initiation conditions, subclause 21 for macros defining the handling of services in the MSC). For these macros there are two possible outcomes:

- a positive outcome, in which case the process continues waiting for the MAP_UPDATE_LOCATION_AREA confirmation; or
- an error is reported, in which case the process terminates (not applicable for Trace_Subscriber_Activity_MSC, which has only a positive outcome).

After receiving the MAP_UPDATE_LOCATION_AREA indication and handling these optional services, the VLR will decide whether a new TMSI need to be allocated to the subscriber or not.

Updating without TMSI reallocation

If the VLR does not reallocate the TMSI, the MSC will receive a MAP_UPDATE_LOCATION_AREA confirmation next (figure 16.1.1/4).

- if there are no parameters with this primitive, updating was successful and a confirmation will be sent to the MS;
- if there is an error cause contained in the received primitive, this cause will be mapped to the corresponding cause in the confirmation sent to the MS (see GSM 09.10 for the mapping of messages and causes).

Updating including TMSI reallocation

This case is covered by the macro Update Location Completion MSC given in figure 16.1.1/5. The MSC will upon receipt of a MAP_SET_CIPHERING_MODE request send a ciphering command towards BSS/MS. Thereafter, the MAP_FORWARD_NEW_TMSI indication and the MAP_UPDATE_LOCATION_AREA confirmation are received in arbitrary order, causing a confirmation on the radio path containing both new LAI and new TMSI. If the MAP_UPDATE_LOCATION_AREA confirmation contains any error, the updating request is rejected towards the MS:

- the MS will confirm receipt of the new TMSI, resulting in an empty MAP_FORWARD_NEW_TMSI response terminating the dialogue;
- if there is no confirmation received from the A-interface, the dialogue is terminated locally.

Before receiving a MAP_UPDATE_LOCATION_AREA confirmation, the MSC may receive a MAP_CHECK_IMEI indication. Handling of this indication, comprising IMEI request towards the MS and IMEI checking request towards
the EIR, is given in the macro description in subclause 21.6. The result may either be to return to the state Wait for TMSI or to return to terminate.

**Forwarding the Check SS Indication**

When the VLR receives a MAP_FORWARD_CHECK_SS_INDICATION_Ind during the Update LOCATION Area process, this indication is relayed to the MS (see GSM 09.11 for detailed interworking) and the MSC remains in the current state.

**Abort handling**

If the VLR receives a MAP_U_ABORT, a MAP_P_ABORT or a premature MAP_CLOSE indication from the VLR during the location update process, the MSC terminates the process by sending an A_LU_CONFIRM containing the error cause Updating Failure to the MS. If the MSC had already confirmed the location update towards the MS, the process terminates without notification towards the A-interface.

If the MSC receives a MAP_NOTICE indication, it issues a MAP_CLOSE and terminates the A-interface dialogue, and the process terminates.

When the procedure is terminated abnormally on the radio path, the dialogue towards the VLR is aborted with the appropriate diagnostic information, and the procedure terminates.
Figure 16.1.1/4: Process Update_Location_Area_MSC
Figure 16.1.1/5: Macro Update_Location_Completion_MSC
16.1.1.3 Detailed procedure in the VLR

Figure 16.1.1/6 shows the process for location updating in the VLR. The following general macros are used:

- Receive_Open_Ind subclause 21.1;
- Receive_Open_Cnf subclause 21.1;
- Authenticate_VLR subclause 21.5;
- Check_IMEI_VLR subclause 21.6;
- Insert_Subscriber_Data_VLR subclause 21.7;
- Obtain_IMSI_VLR to request the IMSI for the subscriber subclause 21.8;
- Activate_Tracing_VLR and Trace_Subscriber_Activity_VLR subclause 21.9;
- Subscriber_Present_VLR subclause 21.10.

Additionally, the process specific macro

Location_Update_Completion_VLR, for optional initiation of Ciphering and TMSI reallocation as for acknowledgement of the MAP_UPDATE_LOCATION_AREA service, see figure 16.1.1/7,

and the optional process specific macro

VLR_Update_HLR to update the HLR and download subscriber data from there, see figure 16.1.1/8,

are invoked by this process.

Process Initiation

The location area updating process will be activated by receiving a MAP_UPDATE_LOCATION_AREA indication from the MSC. If there are parameter errors in the indication, the process is terminated with the appropriate error sent in the MAP_UPDATE_LOCATION_AREA response to the MSC. Else, the behaviour will depend on the subscriber identity received, either an IMSI or an TMSI.

Updating using IMSI

If the subscriber identity is an IMSI, the VLR checks whether the subscriber is unknown (i.e. no IMSI record). If so, the indicator "Location Information Confirmed in HLR" is set to "Not Confirmed" to initiate HLR updating later on. If the IMSI is known, the VLR checks whether the previous location area identification (LAI) provided in the primitive received from the MSC belongs to this VLR. If it does not, the indicator "Location Information Confirmed in HLR" is set to "Not Confirmed" to initiate HLR updating later on. The process may continue in both cases with the authentication check (see below).

Updating using TMSI

If the subscriber identity is a TMSI, the VLR checks whether the previous location area identification (LAI) provided in the primitive received from MSC belongs to an area of this VLR:

- if so, the TMSI will be checked. In case of location area change within a VLR, the TMSI should be known and the process may continue with the authentication check. Additionally, the indicator "Location Information Confirmed in HLR" is set to "Not confirmed" and the trace activity status is checked in case the target Location Area Id belongs to a new MSC.

- if the TMSI is not known or the subscriber data stored are incomplete, e.g. because the new LA belongs to a different VLR or due to VLR restoration, the indicator "Confirmed by VLR" is set to "Not Confirmed" to initiate HLR updating later on.
If the subscriber has not already been registered in the VLR, i.e. the previous LAI belongs to a different VLR, the indicators "Confirmed by HLR" and "Location Information Confirmed in HLR" are set to "Not Confirmed" and the VLR checks whether the identity of the Previous VLR (PVLR) is derivable from the previous LAI:

- if so, the IMSI and authentication parameters are requested from that VLR using the MAP_SEND_IDENTIFICATION service (see sheet 3 of figure 16.1.1/6), containing the subscriber's TMSI.

- if the dialogue is rejected by the PVLR, the process continues requesting the IMSI from the MS. In case the PVLR reverts to the MAP version Vr dialogue, the VLR will perform the respective procedure of version Vr, too, with outcomes as for the current MAP version dialogue. Else, the process waits the for the respective MAP_SEND_IDENTIFICATION response from the PVLR:
  - if the IMSI is received in that primitive, the process continues with the authentication check;
  - if the IMSI is not received from the previous VLR for any reason, the dialogue to the PVLR is terminated and the IMSI will be requested from the MS;
  - if a MAP_NOTICE indication is received from the PVLR, the dialogue will be terminated by sending a MAP_CLOSE indication, and the process continues requesting the IMSI from the MS;
  - if a MAP_P_ABORT or MAP_U_ABORT indication is received from the MSC while waiting for the MAP_SEND_IDENTIFICATION response, the process is terminated;
  - if a MAP_NOTICE indication is received from the MSC while waiting for the MAP_SEND_IDENTIFICATION response, the dialogue with the PVLR will be aborted by sending a MAP_U_ABORT indication (Remote Operations Failure), the dialogue with the MSC will be terminated by sending a MAP_CLOSE and the process terminates;
  - if the identity of the previous VLR cannot be derived, the process continues by requesting the IMSI from the MS.

**Requesting IMSI from the MS**

For requesting the IMSI from the MS, the macro Obtain.IMSI_VLR described in subclause 21.8 is invoked (see figure 16.1.1/6 sheet 3). The outcome will be:

- OK, i.e. receipt of IMSI, in which case the process continues with the authentication check described below; or
- receipt of an Absent Subscriber error, indicating that the MS did not respond. In this case the System Failure error is reported in the MAP_UPDATE_LOCATION_AREA response towards the MSC and the updating process is terminated;
- aborted, i.e. the MSC dialogue has been released while waiting for the IMSI. In this case the updating process is terminated, too.

**Authentication check**

After a subscriber identity has been received, either in the service indication or by an explicit request procedure, the VLR checks whether authentication of this identity is required (see figure 16.1.1/6 sheet 2). If so, the authentication macro described in subclause 21.5 is invoked. The outcome of this macro can be:

- OK, i.e. the subscriber has been authenticated successfully, in which case the process is continued by setting the indicator "Confirmed by Radio Contact" to "Confirmed" and updating the location information held in the register. Thereafter,
  - if one or both of the indicators "Confirmed by HLR" and "Location Information Confirmed in HLR" is set to "Not Confirmed", HLR updating is invoked first;
  - otherwise the process continues with the Location Update Completion VLR macro described below, and the register is updated after successful completion of this macro.
- Illegal subscriber, i.e. there was a mismatch between expected and received SRES. The VLR checks whether authentication had been performed using the TMSI, in which case a new authentication attempt with IMSI may be started (VLR operator option).
  - if so, the process continues by requesting the IMSI from the MS;
- else, the Illegal Subscriber error is reported in the MAP_UPDATE_LOCATION_AREA response.
- Unknown Subscriber, i.e. the IMSI given is unknown in the HLR. In this case, the subscriber data are deleted in the VLR and the same error is returned in the MAP_UPDATE_LOCATION_AREA response.
- Procedure error, i.e. the authentication process was unsuccessful for some other reason, e.g. because of a failure while requesting authentication information from the HLR. In this case the System Failure error is reported in the MAP_UPDATE_LOCATION_AREA response.
- Null, indicating impossible dialogue continuation (e.g. termination of the radio path), and leading to procedure termination without any further action.

Updating the HLR

If the HLR is to be updated, the VLR_Update_HLR macro described below is performed, with one of the following results (see sheet 4 of figure 16.1.1/6):

- OK, if HLR updating has been completed successfully. The response will contain the HLR number as parameter. Next, the Location_Update_Completion VLR macro is invoked (checking amongst others the roaming restrictions and regional subscription data), and upon successful outcome of this macro the register is updated and the process terminates.
- Roaming Not Allowed, qualified by PLMN Roaming Not Allowed if the location information indicates a PLMN for which the subscriber has no subscription or if the subscribers HLR cannot be reached (e.g. SS7 links to the subscribers HPLMN do not yet exist). In this case, the error Roaming Not Allowed qualified by PLMN Roaming Not Allowed is sent in the MAP_UPDATE_LOCATION_AREA response. The Subscriber Data are deleted in the VLR.
- if Roaming Not Allowed was qualified by the parameter Operator Determined Barring, the same value is sent in the MAP_UPDATE_LOCATION_AREA response to the MSC. The subscriber data are deleted in the VLR.
- Unknown Subscriber, if the subscriber is not known in the HLR. In this case, the subscriber data are deleted in the VLR, and the same error is sent in the MAP_UPDATE_LOCATION_AREA response.
- Procedure error, if there occurs some other error during HLR updating (e.g. abort of the connection to HLR):
  - if the VLR can proceed in stand alone mode (VLR operator option), the Location Update Completion VLR macro is invoked to complete the VLR updating, and the indicator "Confirmed by HLR" remains unchanged;
  - otherwise, the System Failure error is sent in the MAP_UPDATE_LOCATION_AREA response.
- Aborted, indicating that during HLR updating the MSC dialogue has been terminated. In this case, the updating process terminates without any further action.

The macro Location Update Completion VLR

This macro completes the VLR updating process. First, the VLR checks whether there is a roaming restriction for the subscriber (see figure 16.1.1/7):

- if the target LA is not allowed for the subscriber due to national roaming restrictions, the error Roaming Not Allowed with cause National Roaming Not Allowed is returned in the MAP_UPDATE_LOCATION_AREA response towards the MSC.

The subscriber data are not deleted from VLR, to avoid unnecessary HLR updating when roaming into other LAs of the same MSC. An indication that the subscriber is not allowed to roam is set in the VLR (LA Not Allowed Flag set to not allowed). As a consequence the subscriber is not reachable (checked for MTC, SMS and MT USSD) and cannot perform outgoing actions (checked in Access Management).

- if the target LA is not allowed for the subscriber because of regional subscription data (Zone Code List) or Roaming Restriction Due To Unsupported Feature stored in the VLR, the error Roaming Not Allowed with cause Location Area Not Allowed is returned towards the MSC in the MAP_UPDATE_LOCATION_AREA response.

Also in this case the subscriber data are not deleted from VLR, to avoid unnecessary HLR updating when roaming into other LAs of the same MSC. The LA Not Allowed Flag is set to not allowed in the VLR.
- if, after check of possible roaming restrictions, the subscriber is allowed to roam in the target LA, the LA Not Allowed Flag is set to allowed (if necessary), the IMSI Detached Flag is set to attached and the process SUBSCRIBER_PRESENT_VLR is started; this may inform the HLR that the subscriber is present again to retry an SMS delivery (see subclause 16.1.1.7). Thereafter, the VLR checks whether TMSI reallocation is required.

- if so, the VLR sends a MAP_SET_CIPHERING_MODE request containing:
  - Ciphering Mode (version 1 GSM); and
  - Kc, the cipher key to be used.

- if IMEI checking is required by the operator, the VLR will invoke the CHECK_IMEI_VLR macro (see subclause 21.6) to initiate both requesting IMEI from the MS and checking of this IMEI towards the EIR. As result either the service is granted, with process continuation as given below, or the service is rejected, in which case the VLR marks the subscriber as detached and returns an Illegal Equipment error in the MAP_UPDATE_LOCATION_AREA response before the process terminates.

- the VLR then sends a MAP_FORWARD_NEW_TMSI request containing the new TMSI, and the MAP_UPDATE_LOCATION_AREA response containing no parameters. The process will thereafter wait for the MAP_FORWARD_NEW_TMSI confirm. If this indicates a negative outcome, or if a MAP_P_ABORT or a MAP_U_ABORT primitive is received, the old TMSI is frozen. Subsequent accesses of the MS shall be accepted with both old or new TMSI.

- if TMSI reallocation is not required, the VLR invokes the CHECK_IMEI_VLR macro (see subclause 21.6) to initiate both requesting IMEI from the MS and checking of this IMEI towards the EIR, if IMEI Checking is required by the operator. As a result, either the service is granted, in which case the MAP_UPDATE_LOCATION_AREA response is sent without any parameters, or the service is rejected, in which case an Illegal Equipment error is returned in the MAP_UPDATE_LOCATION_AREA response, before the process terminates.

In all cases where the VLR sends a MAP_UPDATE_LOCATION_AREA response to the MSC, the dialogue towards the MSC is terminated by a MAP_CLOSE request with parameter Release Method indicating Normal Release.

**The macro VLR Update HLR**

This macro is invoked by the VLR process for location updating or by some other process handling the first subscriber access to the network after a register failure in order to perform HLR updating. If the VLR does not know the subscribers HLR (e.g. no IMSI translation exists as there are not yet any SS7 links to the subscribers HPLMN), the error Roaming Not Allowed with cause PLMN Roaming Not Allowed is returned.

If the subscribers HLR can be reached, the VLR opens a dialogue towards the HLR (see figure 16.1.1/8) by sending a MAP_OPEN request without any user specific parameters, together with a MAP_UPDATE_LOCATION request containing the parameters

- IMSI, identifying the subscriber;
- Location Info, containing the MSC number;
- VLR Number, the E.164 address of the VLR, to be used by the HLR when addressing the VLR henceforth (e.g. when requesting an MSRN);
- the LMSI as an VLR operator option; this is a subscriber identification local to the VLR, used for fast data base access.

In case the HLR rejects dialogue opening (see subclause 21.1), the VLR will terminate the procedure indicating procedure error. If the HLR indicates version Vr protocol to be used, the VLR will revert to the version Vr procedure concerning the dialogue with the HLR, with outcomes as for the current MAP version procedure.

If the HLR accepts the dialogue, the HLR will respond with:

- a MAP_INSERT_SUBSCRIBER_DATA indication, handled by the macro Insert_Subs_Data_VLR defined in subclause 21.7;

NOTE: The HLR may repeat this service several times depending on the amount of data to be transferred to the VLR and to replace subscription data in case they are not supported by the VLR.
- a MAP_ACTIVATE_TRACE_MODE indication, handled by the macro Activate_Tracing_VLR defined in subclause 21.9;

- a MAP_FORWARD_CHECK_SS_INDICATION_ind. This indication will be relayed to the MSC without any change of the current state.

- the MAP_UPDATE_LOCATION confirmation:
  - if this confirmation contains the HLR Number, this indicates that the HLR has passed all information and that updating has been successfully completed. The VLR is updated using the parameters provided in the service and needed by the VLR. If certain parameters are not needed in the VLR, e.g. because some service is not supported, the corresponding data may be discarded. The VLR sets the “Confirmed by HLR” and “Location information confirmed in HLR” indicators to “Confirmed” to indicate successful subscriber data updating;

  - if the confirmation contains an User error cause (Unknown Subscriber, Roaming Not Allowed or some other), the process calling the macro continues accordingly. In the last case, the subscriber data are marked as incomplete by setting the indicators “Confirmed by HLR” and “Location information confirmed in HLR” to “Not Confirmed”. The same holds if there is a Provider error or a Data error in the confirmation;

- a MAP_P_ABORT, MAP_U_ABORT, or MAP_CLOSE indication. In these cases, the subscriber data are marked to be incomplete and the process continues as in the case of an error reported by the HLR;

- a MAP_NOTICE indication. Then, the dialogue towards the HLR is terminated, the subscriber data are marked to be incomplete and the process continues as in the case of an error reported by the HLR;

- if during HLR updating the VLR receives a MAP_P_ABORT, MAP_U_ABORT or a MAP_CLOSE indication concerning the MSC dialogue, the process is terminated by sending a MAP_U_ABORT request towards the HLR, and subscriber data are marked to be incomplete;

- if during HLR updating the VLR receives a MAP_NOTICE indication concerning the MSC dialogue, the dialogue with the MSC is terminated by sending a MAP_CLOSE, the dialogue with the HLR is terminated by sending a MAP_U_ABORT, subscriber data are marked to be incomplete and the process is terminated.

Abort Handling

If the VLR receives a MAP_NOTICE indication from the MSC while waiting for a MAP service primitive, the VLR will terminate the MSC dialogue by sending a MAP_CLOSE and any pending HLR dialogue by sending a MAP_U_ABORT (Remote Operations Failure), and the process is terminated.
Figure 16.1.1/6 (sheet 1 of 4): Process Update_Location_Area_VLR
**Figure 16.1.1/6 (sheet 2 of 4): Process Update_Location_Area_VLR**
Figure 16.1.1/6 (sheet 3 of 4): Process Update_Location_Area_VLR
Figure 16.1.1/6 (sheet 4 of 4): Process Update_Location_Area_VLR
Figure 16.1.1/7: Location updating in VLR: closing sequence

LA_not_allowed_flag := not allowed
LA_not_allowed_flag := allowed

Figure 16.1.1/7: Macro Location_Update_Completion_VLR
Macro definition VLR_Update_HLR

Figure 16.1.1/8 (sheet 1 of 2): Macro VLR_Update_HLR
Figure 16.1.1/8 (sheet 2 of 2): Macro VLR_Update_HLR
16.1.1.4 Detailed procedure in the HLR

The following macros are used by this process:

- Receive_Open_Ind, defined in subclause 21.1;
- Check_indication, defined in subclause 21.2;
- Insert_Subs_Data_Framed_HLR, described in subclause 16.4.1;
- Control_Tracing_HLR, described in subclause 21.9;

and the processes Cancel_Location_HLR (see subclause 16.1.2) and Subscriber_Present_HLR (see subclause 16.1.1.7) are invoked.

The location updating process in the HLR is activated by receipt of a MAP_UPDATE_LOCATION indication (see figure 16.1.1/9):

- if there is a parameter problem in the indication, the error Unexpected Data Value is returned in the MAP_UPDATE_LOCATION response (see Check_indication macro defined in subclause 21.2); if the subscriber is not known in the HLR, the error Unknown Subscriber is returned in the response. In either case the process terminates;
- tracing shall be set to deactivate in the VLR.
- if the VLR address received in the MAP_UPDATE_LOCATION indication differs from the one actually stored against the subscriber, the Cancel_Location_HLR process is started to cancel the subscriber data in the stored VLR (see subclause 16.1.2).

The next action will be to check whether the subscriber is allowed to roam into the PLMN indicated by the VLR Number given in the MAP_UPDATE_LOCATION indication:

- if the subscriber is not allowed to roam into the PLMN, the error Roaming not Allowed with cause PLMN Roaming Not Allowed is returned in the MAP_UPDATE_LOCATION response, and the routing information stored (VLR number, MSC Number, LMSI) is deleted (deregistration);
- otherwise the HLR database will be updated with information received in the indication. The HLR sets the “MS purged” flag to False and checks whether tracing is required for that subscriber. This is handled by the macro Control_Tracing_HLR described in subclause 21.9.

Thereafter, the macro Insert_Subs_Data_Framed_HLR described in subclause 16.4.1 is invoked. The outcome of this macro may be:

- aborted, in which case the process terminates;
- error, in which case the error System Failure is returned in the MAP_UPDATE_LOCATION response and the process terminates;
- OK, indicating successful outcome of downloading the subscriber data to the VLR.

The SUBSCRIBER_PRESENT_HLR process is then started to alert the Short Message Service Centre, if required (see subclause 16.1.7). Additionally, the MAP_FORWARD_CHECK_SS_INDICATION request is sent to inform the subscriber about an uncertain state of his SS-Data if this is needed due to previous HLR restoration (use of this service may be omitted as an HLR operator option).

Finally the HLR number is returned in the MAP_UPDATE_LOCATION response.

In all cases where the HLR sends a MAP_UPDATE_LOCATION response to the VLR, the dialogue towards the VLR is terminated by a MAP_CLOSE request with parameter Release Method indicating Normal Release.
Figure 16.1.1/9 (sheet 1 of 2): Process Update_Location_HLR
Figure 16.1.1/9 (sheet 2 of 2): Process Update_Location_HLR
16.1.1.5  Send Identification

16.1.1.5.1  General

This service is invoked by a VLR when it receives a MAP_UPDATE_LOCATION_AREA indication containing a LAI indicating that the subscriber was registered in a different VLR (henceforth called the Previous VLR, PVLR). If the identity of the PVLR is derivable for the VLR (usually if both are within the same network), the IMSI and authentication sets are requested from the PVLR (see subclause 16.1.1.3), using the service described in subclause 6.1.4.

NOTE: The service shown in dotted lines indicates the trigger provided by other MAP signalling.

Figure 16.1.1/10: Interface and services for Send Identification

16.1.1.5.2  Detailed procedure in the VLR

The VLR procedure is part of the location area updating process described in subclause 16.1.1.3, see also figure 16.1.1/6 sheet 3.

16.1.1.5.3  Detailed procedure in the PVLR

On receipt of a dialogue request for the Send Identification procedure, (see Receive_Open_Ind macro in subclause 21.1), the PVLR will:

- terminate the procedure in case of parameter problems;
- revert to the MAP version Vr procedure in case the VLR indicated version Vr protocol; or
- continue as below, if the dialogue is accepted.

If the PVLR process receives a MAP_NOTICE indication, it terminates the dialogue by sending a MAP_CLOSE request.

If the PVLR process receives a MAP_SEND_IDENTIFICATION indication from the VLR (see figure 16.1.1/11), it checks whether the subscriber identity provided is known:

- if so, the IMSI and - if available - authentication parameters for the subscriber are returned in the MAP_SEND_IDENTIFICATION response;
- if not, the error Unidentified Subscriber is returned in the MAP_SEND_IDENTIFICATION response.

In all cases where the PVLR sends a MAP_SEND_IDENTIFICATION response to the VLR, the dialogue towards the VLR is terminated by a MAP_CLOSE request with parameter Release Method indicating Normal Release.
Figure 16.1.1/11: Process Send_Identification_PVLR
16.1.1.6 The Process Update Location VLR

This process is started by some other MAP user process in case the HLR need to be updated due to previous network failure. It is invoked when the subscriber accesses the network, e.g. for mobile originated call set-up, response to paging or supplementary services handling. Here, location updating consists only of invoking the macro VLR_Update_HLR described above (see subclause 16.1.1.3), which performs HLR updating and downloading of subscriber data.

If updating is successful (OK) the HLR Number is received in the MAP_UPDATE_LOCATION confirm primitive and the process terminates.

If one of the errors Roaming not Allowed or Unknown Subscriber is received instead, all subscriber data are deleted from the VLR before the process terminates.

In case some other error occurs during HLR updating, the process simply terminates. Note, in all error cases the initiating restoration flags in VLR remain false, therefore a new HLR updating attempt will be started later on.

NOTE: This process will be performed independent from the calling process, no coordination is required.
Figure 16.1.1/12: Process UL_VLR
16.1.1.7 The Process Subscriber Present HLR

The process Subscriber Present HLR is started by the location updating process in HLR to perform actions required for short message alerting. The process checks the Message Waiting Data flag, and if this is set, the macro Alert_Service_Centre_HLR defined in subclause 21.10 is invoked. This macro will alert all service centres from which there are short messages waiting for this subscriber.
Figure 16.1.1/13: Process Subscriber_Present_HLR
16.1.2 Location Cancellation

16.1.2.1 General

The purpose of this process is to delete a subscriber's record from a previous visitor location register after she has registered with a new visitor location register. The procedure may also be used if the subscriber's record is to be deleted for other operator determined purposes, e.g. withdrawal of subscription, imposition of roaming restrictions or modifications to the subscription which result in roaming restrictions. Location cancellation can be used to enforce location updating including updating of subscriber data in the VLR at the next subscriber access.

In all cases, the process is performed independently of the invoking process (e.g. Location Updating).

The service as described in subclause 6.1.3 is invoked when an HLR receives a MAP_UPDATE_LOCATION indication from a VLR other than that stored in its table for this subscriber. Additionally the service may be invoked by operator intervention. The MAP_CANCEL_LOCATION service is in any case invoked towards the VLR whose identity is contained in the HLR table.

NOTE: The service shown in dotted lines indicates the trigger provided by other MAP signalling.

Figure 16.1.2/1: Interface and services for Location Cancellation

16.1.2.2 Detailed procedure in the HLR

The location cancellation process is started by an external process as stated above. The HLR opens a dialogue with the VLR whose identity is contained in the HLR table (MAP_OPEN request without any user specific parameters), sending the MAP_CANCEL_LOCATION request primitive (see figure 16.1.2/2), containing the parameters:

- IMSI, to identify the subscriber to be deleted from that VLR;
- LMSI, which is included if available in the HLR.

The HLR then waits for the MAP_OPEN confirmation (see macro Receive_Open_Cnf, subclause 21.1), indicating either:

- reject of the dialogue (process terminates);
- reversion to version Vr (process will be performed according to MAP version Vr); or
- dialogue acceptance.

When the VLR accepts the dialogue, it will return a MAP_CANCEL_LOCATION confirmation, containing:

- no parameter, indicating successful outcome of the procedure;
- a user error, provider error or a data error indicating unsuccessful outcome of the procedure.

In case of unsuccessful outcome or if a MAP_P_ABORT indication has been received, the HLR may repeat the MAP_CANCEL_LOCATION request later, where the number of repeat attempts and time in between are HLR operator options, depending on the error returned by the VLR.
16.1.2.3 Detailed procedure in the VLR

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1, with outcomes:

- reversion to version Vr procedure;
- procedure termination; or
- dialogue acceptance, with processing as below.

If the VLR process receives a MAP_NOTICE indication, it terminates the dialogue by sending a MAP_CLOSE request.

If the VLR process receives a MAP_CANCEL_LOCATION indication from the HLR (see figure 16.1.2/3), the parameters are checked first (macro Check_Indication, see subclause 21.2). In case of parameter problems the appropriate error is sent in the MAPCANCELLOCATION response.

If the MAPCANCELLOCATION indication contains both the IMSI and the LMSI, the VLR checks whether the stored IMSI matches the received IMSI. If it does not, the VLR attempts to process the request using the IMSI received from the HLR to define the subscriber record to be deleted.

Thereafter the VLR checks whether the subscriber identity provided is known in the VLR:

- if so, the data of the subscriber are deleted from VLR table and a MAP_CANCELLOCATION response is returned without any parameters;
- if not, location cancellation is regarded as being successful, too, and the MAPCANCELLOCATION response is returned without any parameters.

In either case, after sending the MAPCANCELLOCATION response the VLR process releases any TMSI which may be associated with the IMSI of the subscriber, terminates the dialogue (MAP_CLOSE with Release Method Normal Release) and returns to the idle state.
Figure 16.1.2/2: Location Cancellation in the HLR

Process Cancel_Location_HLR

- MAP_OPEN_Req, MAP_CANCEL_LOCATION_Req, MAP_DELIMITER_Req
- Receive_Open_cnf
- MAP.Cancel_Location_Cnf, MAP.Abort_Ind, MAP.Notice_Ind
- MAP_CLOSE_Req
- Check Confirmation

- OK
- Error
- WAIT_FOR_ACK
- Perform MAP Vr Dialogue
- Repeat location cancellation
- Wait for repetition timer expiry
- Provider Error, User Error, Data Error
- no
- yes
- O.K.
Figure 16.1.2/3: Location Cancellation in the VLR

Process Cancel_Location_VLR

1. NULL
2. Receive_Open_Ind
   - OK
   - Wait_For_Service_Primitive
      - MAP_CANCELLOCATION_Ind
3. MAP_NOTICE_Ind
   - NULL
   - Check_Indication
     - OK
     - Subscriber registered
       - no
       - Delete subscriber from register
       - Release TMSI
         - MAP_CANCELLOCATION_Cmd
           - MAP_CLOSE_Cmd
4. NULL

Figure 16.1.2/3: Process Cancel_Location_VLR
16.1.3 Detach IMSI

16.1.3.1 General

On receipt of an A_LU_REQUEST (DETACH IMSI) indication from the radio interface this procedure invokes the MAP_DETACH_IMSI service described in subclause 6.1.5 in order to inform the visitor location register that a subscriber is no longer reachable (see figure 16.1.3/1), e.g. due to switched off station. This information is used by the VLR to reject mobile terminating calls or short messages without sending page messages on the radio path. The service is unconfirmed as it is likely that the MS is switched off before receiving a confirmation.

The detach IMSI feature is optional for the network operator. The MS is informed by the network whether detach IMSI is to be used or not.

NOTE: The service shown in dotted lines indicates the trigger provided by the radio interface (see GSM 09.10).

Figure 16.1.3/1: Interface and services for MAP_DETACH_IMSI

16.1.3.2 Detailed procedure in the MSC

The MAP_DETACH_IMSI service is invoked by the MSC when receiving an A_LU_Request (DETACH IMSI) for a subscriber (see figure 16.1.3/2).

The MSC will open the dialogue to the VLR with a MAP_OPEN request containing no user specific parameters. The MAP_DETACH_IMSI request will contain the following parameter received from the radio side (for the mapping see GSM 09.10):

- Subscriber Id, being either a TMSI or an IMSI.

The MSC then waits for the MAP_OPEN confirmation (see macro Receive_Open_Cnf, subclause 21.1), indicating either:

- reject of dialogue (process terminates);
- reversion to version Vr(process terminates); or
- dialogue acceptance.

Thereafter, the dialogue is terminated locally by the MSC (MAP_CLOSE request with Release Method Prearranged End).

16.1.3.3 Detailed procedure in the VLR

When the VLR receives a MAP_DETACH_IMSI indication (see figure 16.1.3/3), it first checks the indication data (macro Check_Indication, see subclause 21.2). Thereafter it is checked whether the subscriber is known:

- if the subscriber is unknown the VLR ignores the indication;
- if the subscriber is known in the VLR, the IMSI detached flag is set.

The VLR process will terminate the dialogue locally (MAP_CLOSE request with Release Method Prearranged End).
Process Detach.IMSI.MSC

Figure 16.1.3/2: Process Detach.IMSI.MSC
Figure 16.1.3/3: Process Detach.IMSI.VLR
16.1.4 Purge MS

16.1.4.1 General

When the VLR receives an indication on the O&M interface that the MS record is to be purged (either because of administrative action or because the MS has been inactive for an extended period), this procedure invokes the MAP_PURGE_MS service described in subclause 6.1.6 to request the HLR to set the "MS purged" flag for the MS so that any request for routing information for a mobile terminated call or a mobile terminated short message will be treated as if the MS is not reachable. The message flow is shown in figure 16.1.4/1.

It is optional for the network operator to delete MS records from the VLR, but if the option is used the VLR shall notify the HLR when a record has been deleted.

The O&M process in the VLR must ensure that during the MS purging procedure any other attempt to access the MS record is blocked, to maintain consistency of data.

When the VLR receives an indication from O&M that an MS record is to be purged, it invokes the MAP_PURGE_MS service (see figure 16.1.4/2).

The VLR opens the dialogue to the HLR with a MAP_OPEN request containing no user specific parameters. The MAP_PURGE_MS request contains the IMSI of the MS which is to be purged and the VLR number.

The VLR then waits for the MAP_OPEN confirmation (see macro Receive_Open_Cnf, subclause 21.1), indicating one of:
- rejection of the dialogue (process terminates);
- reversion to version one (process terminates);
- dialogue acceptance.

If the HLR accepts the dialogue it returns a MAP_PURGE_MS confirmation, containing no parameter, indicating successful outcome of the procedure.

If a MAP_PURGE_MS confirmation containing a provider error, data error or user error, or a MAP_P_ABORT, MAP_NOTICE or premature MAP_CLOSE indication, has been received, the failure is reported to the O&M interface. Successful outcome of the procedure leads to deletion of the subscriber data and freezing of the TMSI, and is reported to the O&M interface.

16.1.4.3 Detailed procedure in the HLR

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1. The possible outcomes are:
- termination of the procedure if the AC indicates a version 1 dialogue, as this procedure is defined only for version 2;
- termination of the procedure if there is an error;
If the HLR receives a MAP_NOTICE indication, it terminates the dialogue by sending a MAP_CLOSE request.

If the HLR receives a MAP_PURGE_MS indication (see figure 16.1.4/3), it first checks the indication data (macro Check_Indication, see subclause 21.2). If there is a parameter error the HLR terminates the dialogue by sending a MAP_CLOSE request (local termination). If there is no parameter error the HLR then checks whether the subscriber is known.

- if the subscriber is unknown, the HLR reports an error to the O&M interface, and terminates the dialogue by sending a MAP_CLOSE request (local termination);

- if the subscriber is known, the HLR checks whether the purging notification came from the VLR where the MS was last registered:
  - if the received VLR number and the stored VLR number match, the HLR sets the "MS purged" flag for the subscriber and sends a MAP_PURGE_MS response containing an empty result to indicate successful outcome;
  - if the received VLR number and the stored VLR number do not match, the HLR sends a MAP_PURGE_MS response containing an empty result to indicate successful outcome. Since the MS is known by the HLR to be in a different VLR area, it is not appropriate to block mobile terminated calls or short messages to the MS, but the VLR which initiated the purging procedure can safely purge its record for the MS.

In either cases of successful termination the HLR terminates the dialogue by sending a MAP_CLOSE request.
Figure 16.1.4/2: Process Purge_MS_VLR
Figure 16.1.4/3: Process Purge_MS_HLR
16.2 Handover procedure

16.2.1 General

The handover between different MSCs is called Inter-MSC handover. The interfaces involved for Inter-MSC handover are shown in figure 16.2/1. Following two Inter-MSC handover procedures apply:

1) Basic Inter-MSC handover:

The call is handed over from the controlling MSC, called MSC-A to another MSC, called MSC-B (figure 16.2/1a).

Figure 16.2/2 shows a successful handover between MSC-A and MSC-B including a request for handover number allocation by MSC-B to VLR-B.

2) Subsequent Inter-MSC handover:

After the call has been handed over from MSC-A to MSC-B, a handover to either MSC-A (figure 16.2/1a) or to a third MSC (MSC-B') (figure 16.2/1b) is necessary in order to continue the connection.

Figure 16.2/3 shows a successful subsequent handover.

a) Basic handover procedure MSC-A to MSC-B and subsequent handover procedure MSC-B to MSC-A.

b) Subsequent handover procedure MSC-B to MSC-B'.

Figure 16.2/1: Interface structure for handover

The MAP handover procedures achieve the functionality required to set up an MSC-MSC dialogue, to optionally allocate a handover number and to transport BSSAP messages.

The transported BSSAP messages are controlled and handled by the Handover Control Application in the MSCs. This information will be transparent to the MAP protocol. If the MSC receives via the MAP protocol BSSAP messages, this information will be forwarded to the Handover Control Application (shown in the handover SDL diagrams with the internal HO_CA signalling, it is an internal process in the MSC) and vice versa if the Handover Control Application requires the sending of BSSAP messages via the MAP protocol.
For detailed interworking between the A-interface and MAP procedures, see GSM 03.09 and GSM 09.10.

NOTE: This can be sent at any time after the connection between MSC-A and MSC-B is established.

**Figure 16.2/2: Example of a successful basic handover procedure to MSC-B**
NOTE: This can be sent at any time after the connection between MSC-A and MSC-B is established

Figure 16.2/3a: Example of a handover towards a third MSC
16.2.2 Handover procedure in MSC-A

This subclause describes the handover procedure in MSC-A, including the request for a basic handover to another MSC (MSC-B), subsequent handover to a third MSC (MSC-B') or back to the controlling MSC (MSC-A).

16.2.2.1 Basic handover

When MSC-A has decided that a call has to be handed over to MSC-B, the Handover Control Application in MSC-A requests the MAP application to initiate the MAP_PREPARE_HANDOVER request to MSC-B.

MSC-A opens the dialogue to MSC-B with a MAP_OPEN request containing no user specific parameters and sends a MAP_PREPARE_HANDOVER request. This request may optionally contain an indication that a handover number allocation is not required, targetCellId, for compatibility reasons, and all information required by MSC-B to allocate the necessary radio resources.

If MSC-B accepts the dialogue, it returns a MAP_PREPARE_HANDOVER confirmation containing a handover number, unless the request has included the HO-NumberNotRequired parameter, and BSSAP information which is forwarded to and handled by the Handover Control Application in MSC-A.

Optionally MSC-A can receive, after a MAP_PREPARE_HANDOVER confirmation, a MAP_FORWARD_ACCESS_SIGNALLING indication containing BSSAP information.

When the connection has been established between the MS and MSC-B, MSC-A will be informed by a MAP_SEND_END_SIGNAL indication.

When MSC-A wants to clear the connection with BSS-B, an indication from the Handover Control Application is received in the Map Application to send the MAP_SEND_END-SIGNAL response to MSC-B to close the MAP dialogue.

MSC-A may abort the handover procedure at any time (e.g. if the call is cleared).

16.2.2.2 Handling of access signalling

If required, the Handover Control Application in MSC-A requests the MAP application to invoke the MAP_FORWARD_ACCESS_SIGNALLING request containing the information to be transferred to the A-interface of MSC-B (e.g. call control information).

MAP_FORWARD_ACCESS_SIGNALLING is a non-confirmed service.

MSC-B will then forward the required information to the Handover Control Application. The MAP_FORWARD_ACCESS_SIGNALLING is composed in such a way that the information can be passed transparently to the A-interface for call control and mobility management information. Any response received in MSC-
B from the A-interface that should be brought to MSC-A will require a new independent request from the Handover Control Application in MSC-B to MSC-A by invoking a MAP_PROCESS_ACCESS_SIGNALLING request.

16.2.2.3 Other procedures in stable handover situation

During a call and after handover, a number of procedures between MSC-A and BSS-B controlled by or reported to MSC-A may be initiated in both directions by invoking a MAP_FORWARD_ACCESS_SIGNALLING request and reception of a MAP_PROCESS_ACCESS_SIGNALLING indication.

16.2.2.4 Subsequent handover

When MSC-A receives a MAP_PREPARE_SUBSEQUENT_HANDOVER request, it will start the procedure of handing the call over to a third MSC (MSC-B'), or back to the controlling MSC (MSC-A). If the new handover procedure towards MSC-B' or MSC-A is successful, the handover control application in MSC-A will request the release of the dialogue towards MSC-B by sending the MAP_SEND_END_SIGNAL confirmation.

16.2.2.5 SDL Diagrams

The SDL diagrams on the following pages describe the user processes in MSC-A for the procedures described in this subclause.

The services used are defined in subclause 6.4.

NOTE: The message primitives HO_CA_MESSAGE used in the SDL-Diagrams are used to show the internal co-ordination between the MAP application and the Handover Control Application. For a detailed description of the co-ordination between the applications for the handover procedure, see GSM 03.09.

Note that in case of reception of errors from the MSCs (see the Handover error handling macro), the MAP user reports them to the Handover Control Application and does not take any action except in cases explicitly mentioned in the SDL diagrams.
Figure 16.2.2/1: HO in MSC-A

- **Figure 16.2.2/1: Process MSC_A_HO**

  - **NULL**
  - **HO_CA_MESSAGE_req**, see NOTE 1 (Handover request)
  - **MAP_OPEN_req**, **MAP_PREPARE_HANDOVER_req**, **MAP_DELIMITER_req**
  - **Receive_Open_CnH**
    - **See section 21.1**
  - **Ok**
    - **Wait_for_PH_result**
  - **MAP_PREPARE_HANDOVER_conf**
  - **Check_Confirmation**
    - **Ok**
      - **MAP_U_ABORT_req**, see NOTE 1
    - **Error**
      - **Error** Ok
  - **Provider error**
  - **User error**
  - **Data error**
  - **Wait_for_HO_completion on MSC-B**

**See section 21.2**

**See section 16.2.4**

**See section 21.1**

**See section 16.2.4**

**See section 16.2.4**
Figure 16.2.2/1 (sheet 2 of 12): Process MSC_A_HO
Call on MSC-B

HO_CA_MESSAGE_ind
see NOTE 1:
[Message transfer]

MAP_PROCESS_ACCESS_SIGNALLING_ind
See section 16.2.4

Receive_error
from HO_CA
or MSC

Null

Call on MSC-B

Check_indication
See section 21.2

Error
Ok

HO_CA_MESSAGE_req
see NOTE 1:
[Message transfer]

MAP_FORWARD_ACCESS_SIGNALLING_req
MAP_DELIMITER_req

MAP_PROCESS_ACCESS_SIGNALLING_ind

Call on MSC-B

Figure 16.2.2/1 (sheet 3 of 12): Process MSC_A_HO
Figure 16.2.2/1 (sheet 4 of 12): Process MSC_A_HO
Figure 16.2.2/1 (sheet 5 of 12): Process MSC_A_HO
Figure 16.2.2/1 (sheet 6 of 12): Process MSC_A_HO
Figure 16.2.2/1 (sheet 7 of 12): Process MSC_A_HO
Wait for SPH_result

HO_CA_MESSAGE_ind, see NOTE 1, [Message Transfer]

MAP_FORWARD_ACCESS_SIGNALLING_req, MAP_DELIMITER_req

to MSC-B

Check_indication

Error

Ok

HO_CA_MESSAGE_req, see NOTE 1, [Message Transfer]

Wait for SPH_result

Figure 16.2.2/1 (sheet 8 of 12): Process MSC_A_HO
Wait_for SPH_result

MAP_PREPARE_HANNOVER_rej

from new MSC-B'

MAP_U_ABORT_req

Ok

Provider error
User error
Data error

To old MSC-B

UserError=
SubsequentHandover
Failure

MAP_PREPARE_SUBSEQUENT_HANNOVER_REQ

MAP_DELIMITER_req

Null.
Error

Figure 16.2.2/1 (sheet 9 of 12): Process MSC_A_HO
Wait for HO_request for MSC-B

HO_CA_MESSAGE_ind, see NOTE 1
[HO preparation result]

Receive error from HO_CA or MSC
See section 16.2.4

User Error=
SubsequentHandover Failure

Null

Null

Call on MSC-B

MAP_PREPARE_SUBSEQUENT_HANDOVER_rsp
MAP_DELIMITER_req

To old MSC-B

MAP_PREPARE_SUBSEQUENT_HANDOVER_rsp
MAP_DELIMITER_req

Figure 16.2.2/1 (sheet 10 of 12): Process MSC_A_HO
Figure 16.2.2/1 (sheet 11 of 12): Process MSC_A_HO
Figure 16.2.2/1 (sheet 12 of 12): Process MSC_A_HO
16.2.3 Handover procedure in MSC-B

This subclause describes the handover procedure in MSC-B, including the request for a handover from another MSC (MSC-A), subsequent handover to a third MSC (MSC-B') or back to the controlling MSC (MSC-A).

16.2.3.1 Basic handover

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1.

When MSC-B process receives a MAP_PREPARE_HANDOVER indication from MSC-A, MSC-B requests its associated VLR to provide a handover number, unless the parameter HO-NumberNotRequired is received in the indication.

When the connection between the MS and MSC-B is established on MSC-B, the Handover Control Application will request the MAP application to indicate this event to MSC-A by invoking the MAP_SEND_END_SIGNAL request. When a call is released, MSC-A will inform MSC-B by MAP_SEND_END_SIGNAL response and the MAP dialogue between MSC-A and MSC-B is closed.

16.2.3.2 Allocation of handover number

When a handover number is required, a MAP_ALLOCATE_HANDOVER_NUMBER request will be sent to the VLR. The handover number is received in the MAP_SEND_HANDOVER_REPORT request, and will be included in the MAP_PREPARE_HANDOVER response to MSC-A.

As soon as the call from MSC-A using the handover number arrives in MSC-B, MSC-B shall release the handover number in the VLR using the MAP_SEND_HANDOVER_REPORT response.

16.2.3.3 Handling of access signalling

If required by the Handover Control Application, MSC-B invokes the MAP_PROCESS_ACCESS_SIGNALLING request containing the information received on the A-interface that should be transferred to MSC-A (e.g. call control information).

MAP_PROCESS_ACCESS_SIGNALLING is a non-confirmed service and any response from MSC-A will require a MAP_FORWARD_ACCESS_SIGNALLING request.

16.2.3.4 Other procedures in stable handover situation

During a call and after handover, a number of procedures between MSC-A and BSS-B controlled by or reported to MSC-A may be initiated by involving access signalling transfer in both directions.

16.2.3.5 Subsequent handover

The procedure is used when the Handover Control Application in MSC-B has decided that a call is to be handed over to another MSC (either back to the controlling MSC (MSC-A) or to a third MSC (MSC-B')).

After the MAP_Prepare_SUBSEQUENT_HANDOVER response is received from MSC-A, MSC-B will await the disconnection of the call. Once the disconnect is complete, MSC-B will inform its VLR by invoking the MAP_SEND_HANDOVER_REPORT confirmation. VLR-B will then release the allocated handover number.

The subsequent handover procedure is shown in figure 16.2/3.
16.2.3.6 SDL Diagrams

The SDL diagrams on the following pages describe the user process in MSC-B for the procedures described in this subclause.

The services used are defined in subclause 6.4.

NOTE 1: The message primitives HO_CA_MESSAGE in the SDL-diagrams are used to show the internal co-ordination between the MAP application and the Handover Control Application. For a detailed description of the co-ordination between the applications for the handover procedure, see GSM 03.09.

NOTE 2: The order in the SDL diagrams to allocate first the handover number and then the radio resources is not binding.
Figure 16.2.3/1 (sheet 1 of 11): Process MSC_B_HO
Figure 16.2.3/1 (sheet 2 of 11): Process MSC_B_HO
Figure 16.2.3/1 (sheet 3 of 11): Process MSC_B_HO
Figure 16.2.3/1 (sheet 4 of 11): Process MSC_B_HO
Figure 16.2.3/1 (sheet 5 of 11): Process MSC_B_HO
Figure 16.2.3/1 (sheet 6 of 11): Process MSC_B_HO
Figure 16.2.3/1 (sheet 7 of 11): Process MSC_B_HO
Figure 16.2.3/1 (sheet 8 of 11): Process MSC_B_HO
Figure 16.2.3/1 (sheet 9 of 11): Process MSC_B_HO
Figure 16.2.3/1 (sheet 10 of 11): Process MSC_B_HO
Figure 16.2.3/1 (sheet 11 of 11): Process MSC_B_HO
16.2.4 Handover error handling macro

This macro is used for the handover procedures to receive errors from the MSCs and from the Handover Control Application at any state of a handover process.

If a MAP_NOTICE indication is received, the Handover Control Application is informed and the actual situation is kept and the Handover Control Application decides how the handover process should continue. In all other cases the MSC is returned to a "NULL" state.
Figure 16.2.4/1: Macro Receive_error_from_HO_CA_or_MSC
16.2.5 Handover procedure in VLR

16.2.5.1 Allocation of handover number

When receiving the MAPALLOCATE_HANDOVER_NUMBER indication, the VLR will determine whether a handover number is available. If no handover number is available, this will be indicated by a MAPALLOCATE_HANDOVER_NUMBER response with the appropriate error.

The handover number allocated will otherwise be returned to MSC-B in the MAP_SEND_HANDOVER_REPORT request.

The handover number will be reserved until a MAP_SEND_HANDOVER_REPORT confirmation is received from MSC-B.

16.2.5.2 SDL Diagrams

The SDL diagrams on the following pages describe the user processes in VLR for the procedures described in this subclause.

The services used are defined in subclause 6.4.
Figure 16.2.5/1 (sheet 1 of 2): Process VLR_B_HO
Figure 16.2.5/1 (sheet 2 of 2): Process VLR_B_HO
16.3 Fault recovery procedures

After a fault of a location register, the fault recovery procedures ensure that the subscriber data in the VLR become consistent with the subscriber data that are stored in the HLR for the MS concerned and that the location information in HLR and VLR reflect accurately the current location of the MS.

The detailed specification of fault recovery procedures of location registers is given in GSM 03.07.

16.3.1 VLR fault recovery procedures

The following processes are involved with the restoration of one IMSI record in the VLR:

- In case of a location registration request from the MS:
  Update_Location_Area_VLR subclause 16.1.1.3;
  Update_Location_HLR subclause 16.1.1.4.

- In case of a mobile terminated call:
  PRN_VLR subclause 18.2.4;
  RESTORE_DATA_VLR subclause 18.2.4;
  RESTORE_DATA_HLR subclause 16.3.3;
  ICS_VLR subclause 18.3.3.

After a restart, the VLR shall erase all IMSI records affected by the failure and shall cause all affected TMSIs and all affected LMSIs to become invalid. There will be no subscriber data or location information stored for an affected MS until after the VLR has received either a MAP_PROVIDE_ROAMING_NUMBER indication or a MAP_UPDATE_LOCATION_AREA indication for that MS. Restoration of subscriber data in the VLR is triggered individually for each IMSI record by receipt of either of these indications.

Reception of either a MAP_UPDATE_LOCATION_AREA indication or a MAP_PROVIDE_ROAMING_NUMBER indication with an IMSI that is unknown in the VLR causes creation of a skeleton IMSI record that is marked as:

- not confirmed by radio contact by the indicator "Confirmed by Radio Contact" (The function of this indicator is described in GSM 03.07), and
- not confirmed by HLR by the indicator "Confirmed by HLR" (The function of this indicator is described in GSM 03.07).

A third indicator "Location Information Confirmed in HLR" is allocated to each IMSI record in the VLR (The function of this indicator is described in GSM 03.07).

The indicator "Location Information Confirmed in HLR" shall be checked whenever authenticated radio contact with an MS has been established. The status "Not Confirmed" of this indicator shall force the VLR to invoke the MAP_UPDATE_LOCATION service but it shall never cause rejection of a mobile originated request. The status is changed from "Not Confirmed" to "Confirmed" only after successful completion of a MAP_UPDATE_LOCATION procedure for the MS concerned.

If the VLR serves only one MSC, the indicator "Location Information Confirmed in HLR" is only relevant to the HLR restoration procedure and an initial value must be assigned when an IMSI record is created in the VLR:

- if the IMSI record was created due to a roaming number request, the initial value must be set to "Confirmed";
- if reception of a MAP_UPDATE_LOCATION_AREA indication causes creation of the IMSI record, the initial value must be "Not Confirmed".

If the VLR serves more than one MSC, the indicator "Location Information Confirmed in HLR" is used in the VLR restoration procedure as well as in the HLR restoration procedure. When an IMSI record is created in the VLR, the indicator must be set to "Not Confirmed".
VLR restoration triggered by a location registration request

Upon receipt of a MAP_UPDATE_LOCATION_AREA indication, the VLR retrieves authentication data from the HLR by using the MAP_SEND_AUTHENTICATION_INFO service if authentication is required and if no authentication data are available in the VLR for the IMSI concerned (see figure 16.1.1/6).

Receipt of a MAP_UPDATE_LOCATION_AREA indication for an MS whose IMSI is unknown in the VLR or whose data stored in the VLR are marked as "Not Confirmed" by the indicator "Confirmed by HLR" and/or by the indicator "Location Information Confirmed in HLR" forces the VLR to invoke the MAP_UPDATE_LOCATION service after successful authentication, if required. The location updating procedure is performed as described in subclause 16.1.

Any other mobile originated request from an MS whose IMSI is unknown in the VLR or whose subscriber data stored in the VLR are marked as "Not Confirmed" by the indicator "Confirmed by HLR" shall be rejected with error cause "Unidentified Subscriber". This causes the MS to trigger the location registration procedure.

After successful completion of the MAP_UPDATE_LOCATION procedure, the indicators "Confirmed by HLR" and "Location Information Confirmed in HLR" are set to "Confirmed".

The indicator "Confirmed by Radio Contact" is set to "Confirmed" when the radio contact with the MS is authenticated.

VLR restoration triggered by a roaming number request

Figure 16.3/1 illustrates the signalling sequence for restoration of an IMSI record in the VLR triggered by a mobile terminating call set-up.

Upon receipt of a MAP_PROVIDE_ROAMING_NUMBER indication for an IMSI that is unknown in the VLR and for which authentication is required, the VLR retrieves authentication data from the HLR by using the MAP_SEND_AUTHENTICATION_INFO service after an MSRN has been sent to the HLR in the MAP_PROVIDE_ROAMING_NUMBER response.

Receipt of a MAP_PROVIDE_ROAMING_NUMBER indication for an MS whose IMSI is unknown in the VLR or whose data record in the VLR is marked as "Not Confirmed" by the indicator "Confirmed by HLR" forces the VLR to request subscriber data from the HLR by sending a MAP_RESTORE_DATA request which triggers one or more INSERT_SUBSCRIBER_DATA operations from the HLR. The MAP_RESTORE_DATA request may also be used to send the LMSI to the HLR.

The MAP_RESTORE_DATA process in the VLR is described in subclause 18.2.4.

The MAP_RESTORE_DATA process in the HLR is described in subclause 16.3.3.

After successful completion of the MAP_RESTORE_DATA procedure, the indicator "Confirmed by HLR" is set to "Confirmed".

If restoration of an IMSI record was triggered by a MAP_PROVIDE_ROAMING_NUMBER indication (i.e. by a mobile terminating call), the VLR has no valid Location Area Identity information for the MS concerned before successful establishment of the first authenticated radio contact. Upon receipt of a MAP_SEND_INFO_FOR_INCOMING_CALL indication from the MSC (see 5 in figure 16.3/1) for an MS whose subscriber data are marked as "Confirmed" by the indicator "Confirmed by HLR" but not confirmed by radio contact, the VLR shall invoke a "MAP_SEARCH_FOR_MS" instead of a "MAP_PAGE".

A MAP_SEARCH_FOR_MS shall also be performed if the VLR receives a MAP_SEND_INFO_FOR_MT_SMS indication from the MSC for an MS whose IMSI record is marked as "Confirmed" by the indicator "Confirmed by HLR" but not confirmed by radio contact.

The indicator "Confirmed by Radio Contact" is set to "Confirmed" when authenticated radio contact caused by a mobile originated or a mobile terminated activity is established.
16.3.2 HLR fault recovery procedures

The following processes are involved with the restart of the HLR:

- HLR_RESTART subclause 16.3.2;
- REC_RESET_IN_VLR subclause 16.3.2.

In the case of a location registration request from the MS, the following processes are involved with the HLR restoration procedure:

- Update_Location_Area_VLR subclause 16.1.1.3;
- Update_Location_HLR subclause 16.1.1.4.

In the case of a mobile originated service request, the

- Macro Process_Access_Request_VLR subclause 21.4.2; and the
- Process Update_Location_HLR subclause 16.1.1.4,

are involved with the HLR restoration procedure.
For the HLR, periodic back-up of data to non-volatile memory is mandatory.

Data that have been changed in the period of time after the last back-up storage and before the restart of the HLR cannot be recovered by reload from the non-volatile memory. Therefore, a restoration procedure is triggered individually for each IMSI record that has been affected by the HLR fault at the first authenticated radio contact that is established with the MS concerned.

The HLR restoration procedure forces updating of MSC number, VLR number and, if provided by the VLR, LMSI in the HLR. Consistency of subscriber data that are stored in the VLR for an MS that has been affected by a HLR fault with the subscriber data stored in the HLR for this MS will be achieved.

As an implementation option, a notification can be forwarded to the MS to alert the subscriber to check the parameters for supplementary services that allow subscriber controlled input (MAP_FORWARD_CHECK_SS_INDICATION service). If the VLR receives this notification from the HLR it shall forward the notification to the MS.

Figure 16.3/2 illustrates the signalling sequence for HLR restoration.

After a restart, the home location register performs the following actions for the subscriber data records that have been affected by the HLR fault (see figure 16.3/3):

- reload all data from the non-volatile back-up;
- if the MAP_FORWARD_CHECK_SS_INDICATION service is implemented, mark each subscriber record "SS Check Required" by setting the "Check SS" indicator;
- set subscriber tracing deactive in the VLR for each of its Mss;
- reset the "MS Purged" flag for each of its MSs;
- send a MAP_RESET request to the VLRs where its MSs are located (see figure 16.3/4).

The MAP_RESET request contains the HLR number and optionally the HLR Identity List.

When receiving a MAP_RESET indication, the VLR will derive all involved MSs of that HLR either from the HLR Identity List (if present), or from the HLR number. The VLR will then mark these MSs with the indicator "Location Information Confirmed in HLR" set to "Not Confirmed" and will deactivate all subscriber tracings for these Mss (see figure 16.3/5).

The status "Not Confirmed" of the indicator "Location Information Confirmed in HLR" forces the VLR to invoke the MAP_UPDATE_LOCATION service after establishment of authenticated radio contact with the MS concerned.

The MAP_UPDATE_LOCATION procedure is performed as described in subclause 16.1.

After receipt of the MAP_UPDATE_LOCATION acknowledge containing the HLR number, the status of the indicator "Location Information Confirmed in HLR" is changed to "Confirmed".

If the MAP_UPDATE_LOCATION procedure is unsuccessful for any reason, the status of the indicator "Location Information Confirmed in HLR" remains unchanged except for the case that the IMSI record in the VLR is deleted because either of the errors "Unknown Subscriber" or "Roaming Not Allowed" has been received from the HLR in response to a MAP_UPDATE_LOCATION request.
Figure 16.3/2: Procedures related to restoration of HLR
Figure 16.3/3: Process HLR_RESTART
Figure 21.1/2: Perform MAP_V1_Dialogue

Figure 16.3/4: Process SEND_RESET_TO_VLR
Figure 16.3/5: Process REC_RESET_IN_VLR

Process REC_RESET_IN_VLR

163_5(1)

Figure 16.3/5: Restoration of the HLR - Application process in the VLR for reception of the RESET message from HLR

Process REC_RESET_IN_VLR

NULL

Receive_OPEN_Ind

'OK'

WAIT

MAP_RESET_Ind

MAP_CLOSE_Req

'OK'

NULL

NULL

Perform_MAP_Vr_Dialogue

NULL

NULL

Identify_IMSI_Records

Location_Info_Conf_in_HLR := Not Confirmed

For each identified IMSI record

Deactivate Subscriber Tracking

For each identified IMSI record

The IMSIs are derived from the HLR_Number or from the HLR_Id_List.

Release method 'Prearranged End'

The HLR_Number or HLR_Id_List contains HLR_Number or HLR_Id_List and HLR_Id_List.
16.3.3 VLR restoration: the restore data procedure in the HLR

The MAP_RESTORE_DATA procedure in the HLR (Process RESTORE_DATA_HLR) is described in this subclause; the corresponding procedure in the VLR (RESTORE_DATA_VLR) is described in subclause 18.2.4.

The process RESTORE_DATA_HLR makes use of the following macros:

- Receive_Open_Ind subclause 21.1.1;
- Check_Indication subclause 21.2.1;
- Insert_Subs_Data_Framed_HLR subclause 16.4.1.

The MAP_RESTORE_DATA service is invoked by the VLR after provision of a roaming number in response to a MAP_PROVIDE_ROAMING_NUMBER indication for an unidentified MS (i.e. IMSI unknown in VLR), or for a known MS whose IMSI record is marked as "Not Confirmed" by the indicator "Confirmed by HLR" (see 4 in figure 16.3/1). The process RESTORE_DATA_VLR is shown in figure 18.2/6.

The restore data process in the HLR is activated by receipt of a MAP_RESTORE_DATA indication from the VLR (see figure 16.3/6). If there is a parameter problem in the indication, either of the errors "Unexpected Data Value" or "Data Missing" is returned in the MAP_RESTORE_DATA response; if the subscriber is not known in the HLR, the error "Unknown Subscriber" is returned in the MAP_RESTORE_DATA response. In all of these cases the process in the HLR terminates.

If the MAP_RESTORE_DATA indication is accepted and if the LMSI is received, the HLR updates the LMSI for the IMSI received in the MAP_RESTORE_DATA indication. For this IMSI the HLR sets "subscriber-tracing-not-active-in-VLR" and checks whether tracing is required. This check is handled by the macro "Control_Tracing_HLR" that is described in subclause 21.9. Thereafter, the macro "Insert_Subs_Data_Framed_HLR" that is described in subclause 16.4.1 is invoked. The outcome of the macro Insert_Subs_Data_Framed_HLR is one of:

- abort, in which case the process terminates;
- error, in which case the HLR returns the error "System Failure" in the MAP_RESTORE_DATA response, and the process terminates;
- OK, indicating successful outcome of downloading the subscriber data to the VLR.

After successful completion of the framed MAP_INSERT_SUBSCRIBER_DATA procedure, the HLR Number and, if applicable, the "MS Not Reachable Flag" which is used for SMS, are provided in the MAP_RESTORE_DATA response.

Upon receipt of the MAP_RESTORE_DATA confirmation, the VLR behaves as described in subclause 18.2.4, figure 18.2/6.
Section 16.4.1

Insert_Subscriber_Data_Framed_HLR

MAP_RESTORE_DATA_Rsp

Section 16.4.1

MAP_CLOSE_Req

Figure 21.2/1

Check_Indication

Figure 21.3/4

Control_Tracing_HLR

'Aborted'

'Set_Error := System_Failure'

MAP_RESTORE_DATA_Ind

MAP_RESTORE_DATA_Rsp

MAP_CLOSE_Req

Figure 21.9/4

'Aborted'

'MSI Known'

'OK'

'Error'

'No'

Yes

'OK'

'No'

'Yes'

'OK'

'Error'

'OK'

'No'

'Yes'

'Error'

'OK'

'Set_Error := Unknown_Subscriber'

Update_LMSI (if received)

'Set_Parameter_HLR_NUMBER'

Set_Parameter_MS_NOT_REACHABLE'

Set_Error := Unknown_Subscriber'

MAP_RESTORE_DATA_Ind

MAP_RESTORE_DATA_Rsp

MAP_CLOSE_Req

Figure 16.3/6: Process RESTORE_DATA_HLR
16.4 Macro Insert_Subs_Data_Framed_HLR

This macro is used by any procedure invoked in HLR which requires the transfer of subscriber data by means of the InsertSubscriberData operation (e.g. Update Location or Restore Data).

The invocation of the operation is done in a dialogue already opened by the framing procedure. Therefore the latter is the one that handles the reception of the open indication and sends the dialogue close request.

The macro calls the process "Send_Insert_Subs_Data" (see subclause 21.7.4) as many times as it is needed for transferring all subscriber data. This process call is meant to describe two possible behaviours of HLR to handle service requests and confirmations:

- either the HLR handles requests and confirmations in parallel; or
- the HLR sends the next request only after receiving the confirmation to the previous one.

Another call is done to the macro "Wait_for_Insert_Subscriber_Data" (see subclause 21.7.3). There the reception and handling of the service confirmations is described.

If certain services required for a subscriber are not supported by the VLR (e.g. Advice of Charge Charging Level), this may result in one of the following outcomes:

- The HLR stores and sends "Roaming Restriction Due To Unsupported Feature" in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. If "Roaming Restriction Due To Unsupported Feature" is stored in the HLR, the "MSC Area Restricted Flag" shall be set to "restricted". This will prevent MT calls, MT SM and MT USSD from being forwarded to the MSC/VLR;

- The HLR stores and sends other induced subscriber data (e.g. a specific barring program) in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. This will cause rejection of mobile originated service requests, except emergency calls.

When the VLR receives regional subscription data (Zone Code List) it may respond with "MSC Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response. In this case the "MSC Area Restricted Flag" shall be set to "restricted" in the HLR. This will prevent MT calls, MT SM and MT USSD from being forwarded to the MSC/VLR.

If the HLR neither stores "Roaming Restriction Due To Unsupported Feature" nor receives "MSC Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response, the "MSC Area Restricted Flag" in the HLR shall be set to "not restricted".

The SDL diagram is shown in figure 16.4/1.
MAP_INSERT_SUBSCRIBER_DATA_Req
MAP_DELIMITER_Req

Count:=1

Yes

All data inserted

Figure 21.7/3

Wait_for_Insert_Subc_Data_Cnf

Figure 21.7/4

Count:= Count+1

Count:= Count-1

Count > 0

 MSC_area_restricted_flag := not restricted

 MSC_area_restricted_flag := restricted

Replace Service

Abort

Replace Service

Abort

MSC_area_restricted_flag received?

Roaming restriction due to unsupported feature or MSC_area restricted received?

Yes

No

Figure 16.4/1: Macro Insert_Subc_Data_Framed_HLR
17 Operation and maintenance procedures

17.1 General

The Operation and Maintenance procedures are needed for operating and maintaining the GSM PLMN network.

The following procedures exist for operation and maintenance purposes:

i) Tracing procedures;

ii) Subscriber Data Management procedures;

iii) Subscriber Identity procedures.

The following application contexts refer to complex MAP Users consisting of several processes:

- subscriberDataManagementContext;
- tracingContext.

These two application contexts need a co-ordinating process in the VLR as described in the following subclauses.

17.1.1 Tracing Co-ordinator for the VLR

The MAP_OPEN indication opens the dialogue for the stand-alone tracing procedure when the application context tracingContext is received. If that service is successful, the Co-ordinator can receive the first service primitive from the MAP_PM. Depending on the received primitive, the user process is created as follows:

- if the MAP_ACTIVATE_TRACE_MODE indication is received, the process ATM_VLR_Standalone is created;
- if the MAP_DEACTIVATE_TRACE_MODE indication is received, the process DTM_VLR_Standalone is created.

After creation of the user process the Co-ordinator relays the messages between the MAP_PM and the invoked process until a request or an indication for dialogue termination is received.

The Tracing Co-ordinator is shown in the figure 17.1/1.
Figure 17.1/1: Co-ordinating process for the tracing procedures in the VLR.

Figure 17.2/9: ATM_VLR_Standalone

Figure 17.2/10: DTM_VLR_Standalone

Workflow SERVICE PRIMITIVE

- MAP_TIMEOUT
- MAP_P_ABORT
- MAP_P_ABORT_req
- MAP_P_ABORT_ind
- MAP_U_ABORT
- MAP_U_ABORT_req
- MAP_U_ABORT_ind
- MAP_CLOSE
- MAP_CLOSE_req
- MAP_CLOSE_ind
- MAP_DEACTIVATE_TRACE_MODE
- MAP_ACTIVATE_TRACE_MODE
- MAP_NOTICE
- MAP_ACTIVATE_TRACE_MODE_req
- MAP_ACTIVATE_TRACE_MODE_ind
- MAP_DEACTIVATE_TRACE_MODE_req
- MAP_DEACTIVATE_TRACE_MODE_ind
- MAP_CLOSE_req
- MAP_CLOSE_ind
- MAP_NOTICE_req
- MAP_NOTICE_ind
- MAP_P_ABORT_req
- MAP_P_ABORT_ind
- MAP_U_ABORT_req
- MAP_U_ABORT_ind
- MAP_CLOSE_req
- MAP_CLOSE_ind
- MAP_NOTICE_req
- MAP_NOTICE_ind
- MAP_P_ABORT
- MAP_P_ABORT_req
- MAP_P_ABORT_ind
- MAP_U_ABORT
- MAP_U_ABORT_req
- MAP_U_ABORT_ind
- MAP_CLOSE
- MAP_CLOSE_req
- MAP_CLOSE_ind
- MAP_DEACTIVATE_TRACE_MODE
- MAP_ACTIVATE_TRACE_MODE
- MAP_NOTICE
- MAP_P_ABORT
- MAP_P_ABORT_req
- MAP_P_ABORT_ind
- MAP_U_ABORT
- MAP_U_ABORT_req
- MAP_U_ABORT_ind
- MAP_CLOSE
- MAP_CLOSE_req
- MAP_CLOSE_ind
- MAP_DEACTIVATE_TRACE_MODE
- MAP_ACTIVATE_TRACE_MODE
- MAP_NOTICE
- MAP_P_ABORT
- MAP_P_ABORT_req
- MAP_P_ABORT_ind
- MAP_U_ABORT
- MAP_U_ABORT_req
- MAP_U_ABORT_ind
- MAP_CLOSE
- MAP_CLOSE_req
- MAP_CLOSE_ind
- MAP_DEACTIVATE_TRACE_MODE
- MAP_ACTIVATE_TRACE_MODE
- MAP_NOTICE
- MAP_P_ABORT
- MAP_P_ABORT_req
- MAP_P_ABORT_ind
- MAP_U_ABORT
- MAP_U_ABORT_req
- MAP_U_ABORT_ind
- MAP_CLOSE
- MAP_CLOSE_req
- MAP_CLOSE_ind
- MAP_DEACTIVATE_TRACE_MODE
- MAP_ACTIVATE_TRACE_MODE
- MAP_NOTICE
- MAP_P_ABORT
- MAP_P_ABORT_req
- MAP_P_ABORT_ind
- MAP_U_ABORT
- MAP_U_ABORT_req
- MAP_U_ABORT_ind
- MAP_CLOSE
- MAP_CLOSE_req
- MAP_CLOSE_ind
- MAP_DEACTIVATE_TRACE_MODE
- MAP_ACTIVATE_TRACE_MODE
- MAP_NOTICE
17.1.2 Subscriber Data Management Co-ordinator for the VLR

The MAP_OPEN indication opens the dialogue for the stand-alone subscriber data management procedure when the application context subscriberDataManagementContext is received. If that service is successful, the Co-ordinator can receive the first service primitive from the MAP_PM. Depending on the received primitive, the user process is created as follows:

- if the MAP_INSERT_SUBSCRIBER_DATA indication is received, the process INS_SUBS_DATA_VLR is created;
- if the MAP_DELETE_SUBSCRIBER_DATA indication is received, the process Delete_Subscriber_Data_VLR is created.

After creation of the user process the Co-ordinator relays the messages between the MAP_PM and the invoked process until a request or an indication for dialogue termination is received.

The Subscriber_Data_Management Co-ordinator is shown in the figure 17.1/2.
Figure 17.1/2: Co-Data_Management_VLR
17.2 Tracing procedures

Three type of tracing procedures exist:

i) Subscriber tracing management procedures;

ii) Subscriber tracing procedures;

iii) Event tracing procedures.

The subscriber tracing management procedures are used for management of the status and the type of the tracing. The subscriber tracing activation procedure is used at location updating or data restoration when the trace mode of a subscriber is set active in the HLR or, as a stand alone procedure, when the subscriber is already registered and the trace mode becomes active in the HLR. The procedures for providing a trace request to the VLR are shown in figures 17.2/1 and 17.2/2.

1) Subscriber Tracing Activation
2) MAP_ACTIVATE_TRACE_MODE
3) MAP_ACTIVATE_TRACE_MODE_ACK
4) Subscriber Tracing Activation Accepted

Figure 17.2/1: Stand alone subscriber tracing activation procedure

![Diagram](image1)

1) MAP_UPDATE_LOCATION or MAP_RESTORE_DATA
2) MAP_ACTIVATE_TRACE_MODE
3) MAP_ACTIVATE_TRACE_MODE_ACK
4) MAP_UPDATE_LOCATION_ACK or MAP_RESTORE_DATA_ACK

Figure 17.2/2: Subscriber tracing activation procedure at location updating or data restoration

The HLR sends the trace request (IMSI, trace reference, trace type and identity of the OMC) to the VLR in a MAP_ACTIVATE_TRACE_MODE request. The receipt of this primitive is acknowledged. The acknowledge primitive will indicate that the trace request is accepted by the VLR. If the request is not accepted, the reason will be reported to the HLR.

The subscriber tracing deactivation procedure is used when the trace request of a subscriber is to be cancelled in the VLR. The procedure is shown in figure 17.2/3.
1) Subscriber Tracing Deactivation
2) MAP_DEACTIVATE_TRACE_MODE
3) MAP_DEACTIVATE_TRACE_MODE_ACK
4) Subscriber Tracing Deactivation Accepted

**Figure 17.2/3: Subscriber tracing deactivation procedure**

The HLR sends a MAP_DEACTIVATE_TRACE_MODE request to the VLR. The VLR will acknowledge the deactivation. The acknowledge primitive will indicate that the trace request has been deleted by the VLR. If the deactivation is not accepted, the reason will be reported to the HLR.

The subscriber tracing procedures are used when the VLR detects any subscriber related activity for which the trace mode is activated, e.g. receives the MAP_PROCESS_ACCESS_REQUEST indication. The procedure is shown in figure 17.2/4.

**Figure 17.2/4: Subscriber tracing procedure in the servicing MSC**

The VLR will generate the MAP_TRACE_SUBSCRIBER_ACTIVITY indication. The receiving MSC will send the trace record to the OMC.

[Figure numbers 17.2/5 and 17.2/6 are spare.]

### 17.2.1 Procedures in the HLR

#### 17.2.1.1 Subscriber tracing activation procedure

When receiving the subscriber tracing mode activation command for a subscriber from the OMC, the HLR will activate tracing, if the subscriber is known and registered in the HLR and the subscriber is roaming in the home PLMN area. The MAP_ACTIVATE_TRACE_MODE request is sent to the VLR where the subscriber is registered.

If the MAP_ACTIVATE_TRACE_MODE confirmation is received indicating an error situation, the errors are mapped to the OMC interface. The activation request may also be repeated; the number of repeat attempts and the time in between are HLR operator options, depending on the error returned by the VLR.

If the subscriber is known in the HLR, but is deregistered or roaming outside the home PLMN area, the subscriber tracing status is activated in the HLR, but the VLR is not updated.

When receiving a request for location updating or data restoration while the subscriber trace mode is active, the macro Control_Tracing_HLR (see figure 21.9/4) shall be initiated by the location updating process in the HLR.

The subscriber tracing activation process in the HLR is shown in figure 17.2/7.
Figure 17.2/7: The subscriber tracing activation process in the HLR

OM_Subscriber_Tracing_Activation_cnf

MAP_CLOSE_ind, MAP_U_ABORT_ind, MAP_P_ABORT_ind

Error

yes

no

Set Subscriber Tracing = Active in VLR

OM_Subscriber_Tracing_Activation_cnf

NULL

MAP_CLOSE_ind, MAP_U_ABORT_ind, MAP_P_ABORT_ind

1 Page 1

Figure 17.2/7 (sheet 2 of 2): Process ATM_HLR
17.2.1.2 Subscriber tracing deactivation procedure

When receiving the subscriber trace mode deactivation command for a subscriber from the OMC, the HLR will send the MAP_DEACTIVATE_TRACE_MODE request to the VLR where the subscriber is registered, if the trace mode activation has been carried out. The subscriber tracing in HLR is set to a deactive state.

If the operation is successful, the HLR will set the subscriber tracing in VLR to a deactive state.

If the MAP_DEACTIVATE_TRACE_MODE confirmation is received indicating an error situation, the errors are mapped to the OMC interface. The deactivation request may be also repeated; the number of repeat attempts and the time in between are HLR operator options, depending on the error returned by the VLR.

The subscriber tracing deactivation procedure is shown in figure 17.2/8.
Figure 17.2/8 (sheet 1 of 2): Process DTM_HLR
Figure 17.2/8: The subscriber tracing deactivation process in the HLR

Figure 17.2/8 (sheet 2 of 2): Process DTM_HLR
17.2.2 Procedures in the VLR

The VLR is involved in the following tracing procedures:

i) Subscriber tracing activation procedure;

ii) Subscriber tracing deactivation procedure;

iii) Subscriber tracing procedure.

17.2.2.1 Subscriber tracing activation procedure

When receiving a MAP_ACTIVATE_TRACE_MODE indication, the VLR will check the parameters and data in the primitive. Data errors are reported as an unexpected data value error or as a data missing error depending on the nature of the error.

If the subscriber is known, the tracing facility is supported and the tracing capacity is not exceeded, the successful report is sent in the MAP_ACTIVATE_TRACE_MODE response primitive.

The MAP_ACTIVATE_TRACE_MODE indication primitive may be received during a location updating or data restoration procedure, so the location updating or restore data process shall use the macro Activate_Tracing_VLR (see figure 21.9/3).

The subscriber tracing activation process in the VLR is shown in figure 17.2/9.
FIGURE 17.2/9 The subscriber tracing activation process for standalone operation in the VLR

MAP_ACTIVATE_TRACE_MODE_ind

Check Indication

Subscriber known

yes

no

Tracing supported

yes

no

Tracing buffer full

yes

no

Set Subscriber Tracing = Active

SET UE = UNIDENTIFIED SUBSCRIBER

SET UE = FACILITY NOT SUPPORTED

Set Subscriber Tracing = Active

MAP_ACTIVATE_TRACE_MODE_rsp

MAP_CLOSE

Figure 17.2/9: Process ATM_VLR_Standalone
17.2.2.2 Subscriber tracing deactivation procedure

When receiving a MAP_DEACTIVATE_TRACE_MODE indication, the VLR will check the parameters and data in the primitive. Data errors are reported as an unexpected data value error or as a data missing error depending on the nature of the error.

If the subscriber is known and the tracing facility is supported, the successful report is sent in the MAP_DEACTIVATE_TRACE_MODE response primitive.

The subscriber tracing deactivation procedure in the VLR is shown in figure 17.2/10.
Figure 17.2/10: The subscriber tracing deactivation process in the VLR

Figure 17.2/10: Process DTM_VLR_Standalone
17.2.2.3 Subscriber tracing procedure

When the VLR receives a MAP_PROCESS_ACCESS_REQUEST or MAP_UPDATE_LOCATION_AREA indication related to any subscriber activity from the MSC, the subscriber tracing procedure may be carried out. The macro Trace_Subscriber_Activity_VLR is shown in figure 21.9/2.

17.2.3 Procedures in the MSC

The MSC is involved in the following tracing procedure:

i) Subscriber tracing procedure.

17.2.3.1 Subscriber tracing procedure

When receiving the MAP_TRACE_SUBSCRIBER_ACTIVITY indication from the VLR, the MSC stores trace reference, trace type and the identity of the OMC in charge of the trace, and the MSC starts to collect the trace information. The MSC will send the trace record to the OMC.

The macro Trace_Subscriber_Activity_MSC is shown in figure 21.9/1.

17.3 Subscriber data management procedures

Two types of subscriber data management procedures exist in the Mobile Application Part

i) Subscriber Deletion;

ii) Subscriber Data Modification.

No requirements have been identified for the Subscriber creation and subscriber data interrogation procedures.

The subscriber deletion and subscriber data modification procedures are initiated by the OMC (see figures 17.3/1 and 17.3/2).

1) Delete Subscriber
2) MAP_CANCEL_LOCATION
3) MAP_CANCEL_LOCATION_ACK
4) Subscriber Deleted

Figure 17.3/1: Subscriber deletion procedure
In the subscriber deletion procedure the subscriber data should be removed from the VLR and from the HLR. The HLR uses the MAP_CANCEL_LOCATION service.

1) Modify Subscriber Data
2) MAP_CANCEL_LOCATION, MAP_INSERT_SUBSCRIBER_DATA or MAP_DELETE_SUBSCRIBER_DATA
3) MAP_CANCEL_LOCATION_ACK, MAP_INSERT_SUBSCRIBER_DATA_ACK or MAP_DELETE_SUBSCRIBER_DATA_ACK
4) Subscriber Data Modified

Figure 17.3/2: Subscriber data modification procedure

In the subscriber data modification procedure the subscriber data is modified in the HLR and when necessary also in the VLR. The HLR initiates either the MAP_INSERT_SUBSCRIBER_DATA, MAP_DELETE_SUBSCRIBER_DATA or MAP_CANCEL_LOCATION service depending on the modified data.

17.3.1 Procedures in the HLR

17.3.1.1 Subscriber deletion procedure

When the subscriber deletion request is received from the OMC, the HLR shall delete the subscriber data from the HLR and initiate the MAP_CANCEL_LOCATION request to the VLR where the subscriber is registered.

The subscriber deletion procedure in the HLR is shown in the figure 17.3/3.
Figure 17.3/3: The subscriber deletion process in the HLR

Figure 17.3/3: Process Delete_Subscriber_HLR
17.3.1.2 Subscriber data modification procedure

The OMC can modify the subscriber data in several different ways. The modifications can be categorized in five groups:

a) HLR internal modification, no effect in the VLR;

b) data shall be modified both in the HLR and VLR;

c) withdrawal of a basic service or a supplementary service;

d) modification affects on the roaming of the subscriber and the subscriber shall be removed from the VLR data base;

e) authentication algorithm or authentication key of the subscriber is modified.

In case "b" the MAP_INSERT_SUBSCRIBER_DATA service is initiated in the HLR.

In case "c" the MAP_DELETE_SUBSCRIBER_DATA service is initiated in the HLR.

In cases "d" and "e" the MAP_CANCEL_LOCATION service is initiated in the HLR.

If the result of a primitive received from the VLR is unsuccessful, the HLR may initiate re-attempts; the number of repeat attempts and the time in between are HLR operator options, depending on the error returned by the VLR.

The subscriber data modification procedure in the HLR is shown in the figures 17.3/4, 17.3/5 and 21.7/2.
Figure 17.3/4: The subscriber data modification process in the HLR

Figure 17.3/4: Process Modify_Data_HLR
Macrodefinition Delete_Subscriber_Data_HLR

Figure 17.3/5: The delete subscriber data macro in the HLR

MAP_DELETE_SUBSCRIBER_DATA_req

MAP_CLOSE_ind

OK

Error

Error

MSC_Area.Restricted_Flag = not_restricted

MSC_Area.Restricted_Flag = restricted

MAP_CLOSE_req

MAP_NOTICE_ind

MAP_U_ABORT_ind,

MAP_P_ABORT_ind,

MAP/Delete_Subscriber_Data_HLR

MAP_DELIMITER

Receive_Open_Cnf

Wait_For_Confirm

MSC_Area.Restricted_Received or "Roaming_Restricted_Due_ToUnsupported_Feature"

Figure 17.3/5: Macro Delete_Subscriber_Data_HLR
17.3.2 Procedures in the VLR

17.3.2.1 Subscriber deletion procedure

The subscriber deletion procedure in the VLR is described in the subclause 16.1.

17.3.2.2 Subscriber data modification procedure

When receiving either the MAP_INSERT_SUBSCRIBER_DATA indication or the MAP_DELETE_SUBSCRIBER_DATA indication, the VLR check the parameters and data in the primitive. Data errors are reported as an unexpected data value error or a data missing error depending on the nature of the error.

After receiving the first MAP_INSERT_SUBSCRIBER_DATA indication, the VLR will check the IMSI that is included in the primitive. If the IMSI is unknown, the error "Unidentified subscriber" is returned.

If the VLR does not support received basic or supplementary services or the network feature Operator Determined Barring, or there is a problem with Regional Subscription Data then it reports it to the HLR.

If the entire MSC area is restricted due to regional subscription, this is reported to the HLR.

If the updating of the subscriber data is not possible, the VLR will initiate the MAP_U_ABORT request primitive. If the updating is successful, the MAP_CLOSE indication is received from the HLR.

The subscriber data modification procedure in the VLR is shown in the figures 17.3/6, 17.3/7 and 21.7/1.
Figure 17.3/6: The insert subscriber data process in the VLR

Process INS_SUBS_DATA_VLR

Figure 17.3/6: Process INS_SUBS_DATA_VLR
Figure 17.3/7: Process Delete_Subscriber_Data_VLR
17.4 Subscriber Identity procedure

In the subscriber identity procedure the IMSI of the subscriber is retrieved from the HLR. The procedure is shown in figure 17.4/1.

1) Identity request
2) MAP_SEND_IMSI
3) MAP_SEND_IMSI_ACK
4) Identity confirm

Figure 17.4/1: The subscriber identity procedure

17.4.1 Subscriber identity procedure in the HLR

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1, with outcomes:

- procedure termination; or
- dialogue acceptance, with proceeding as below.

When receiving the MAP_SEND_IMSI indication, the HLR will check the parameters and data in the primitive. Data errors are reported as an unexpected data value error or a data missing error depending on the nature of the error.

If the subscriber is known in the HLR, the IMSI is fetched from the database and sent to the VLR. If the MSISDN cannot be identified, unknown subscriber indication is passed to the VLR.

The subscriber identity procedure in the HLR is shown in figure 17.4/2.
Figure 17.4/2: Process Send_IMSI_HLR

- **Receive_Open_Ind**
- **V1**
- **MAP_SEND_IMSI_Ind**
- **MAP_CLOSE_req**
- **MAP_CLOSE_ind**
- **MAP_NOTICE_ind**
- **Check_Indication**
  - **OK**
  - **no**
    - **yes**
      - **Fetch_Subscriber_Data**
      - **Set UE = Unknown Subscriber**
    - **null**
17.4.2 Subscriber identity procedure in the VLR

When the IMSI request is received from the OMC, the VLR will send the MAP_SEND_IMSI request to the HLR. The contents of the response is sent to the OMC.

The subscriber identity procedure in the VLR is shown in figure 17.4/3.
Figure 17.4/3: Process Send_IMSI_VLR
18 Call handling procedures

18.1 General

The MAP call handling procedures are used to retrieve routing information to handle a mobile terminating call and to transfer control of a call back to the GMSC if the call is to be forwarded.

The procedures to handle a mobile originating call and a mobile terminating call after the call has arrived at the destination MSC do not require any signalling over a MAP interface. These procedures are specified in GSM 03.18 [97].

The stage 2 specification for the retrieval of routing information to handle a mobile terminating call is in GSM 03.18 [97]; modifications to this procedure for CAMEL are specified in GSM 03.78 [98], for optimal routing of a basic mobile-to-mobile call in GSM 03.79 [99]. The interworking between the MAP signalling procedures and the call handling procedures for each entity (GMSC, HLR and VLR) is shown by the transfer of signals between these procedures.

The stage 2 specification for the transfer of control of a call back to the GMSC if the call is to be forwarded is in GSM 03.79 [99]. The interworking between the MAP signalling procedures and the call handling procedures for each entity (VMSC and GMSC) is shown by the transfer of signals between these procedures.

The interworking between the call handling procedures and signalling protocols other than MAP is shown in GSM 03.18,GSM 03.78 and GSM 03.79.

18.2 Retrieval of routing information

18.2.1 General

The message flows for successful retrieval of routing information for a mobile terminating call are shown in figure 18.2/1 (mobile terminating call which has not been optimally routed) and 18.2/2 (mobile-to-mobile call which has been optimally routed).
Notes:

xxx = Optional Procedure

NOTE 1: This service may also be used by an ISDN exchange for obtaining routing information from the HLR.

NOTE 2: TUP or ISUP may be used in signalling between MSCs, depending on the network type between the
MSCs. For further details on the TUP and ISUP procedures refer to the following ITU-T Recommendations
and ETSI specification:

Q.721-725 - Telephone User Part (TUP);
ETS 300 356-1 - Integrated Services Digital Network (ISDN); Signalling System No.7; ISDN User Part (ISUP) version
2 for the international interface; Part 1: Basic services.

NOTE 3: As a network operator option, the HLR sends
MAP_PROVIDE_SUBSCRIBER_INFORMATION to the VLR. For further details on the CAMEL
procedures refer to GSM TS 03.78;

Figure 18.2/1: Message flow for retrieval of routeing information (non-optimally routed call)
Notes:

- **xxx = Optional Procedure**
- For Optimal Routeing phase 1, only one of the information flows for Provide Subscriber Info and Provide Roaming Number is used. For later phases of Optimal Routeing, the HLR may return a MAP_SEND_ROUTEING_INFORMATION ack after the Provide Subscriber Info information flow, and the GMSC may send a second MAP_SEND_ROUTEING_INFORMATION, which will trigger the Provide Roaming Number information flow.

TUP or ISUP may be used in signalling between MSCs, depending on the network type between the MSCs. For further details on the TUP and ISUP procedures refer to the following CCITT Recommendations & ETSI specification:

- Q.721-725 - Telephone User Part (TUP);
- ETS 300 356-1 - Integrated Services Digital Network (ISDN); Signalling System No.7; ISDN User Part (ISUP) version 2 for the international interface; Part 1: Basic services.

**Figure 18.2/2: Message flow for retrieval of routing information (optimally routed call)**

The following MAP services are used to retrieve routing information:

- **MAP_SEND_ROUTING_INFORMATION** see subclause 8.1;
- **MAP_PROVIDE_ROAMING_NUMBER** see subclause 8.2;
- **MAP_PROVIDE_SUBSCRIBER_INFO** see subclause 6.11.2;
- **MAP_RESTORE_DATA** see subclause 6.10.3.

### 18.2.2 Process in the GMSC

The MAP process in the GMSC to retrieve routing information for a mobile terminating call is shown in figure 18.2/3. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

- **Receive_Open_Cnf** see subclause 21.1.2;
- **Check_Confirmation** see subclause 21.2.2.

**Successful Outcome**

When the MAP process receives a Send Routeing Info request from the call handling process in the GMSC, it requests a dialogue with the HLR whose identity is contained in the Send Routeing Info request by sending a MAP_OPEN service request, requests routing information using a MAP_SEND_ROUTING_INFORMATION service request and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the HLR.
If the MAP process receives a MAP_SEND_ROUTING_INFORMATION service confirm from the HLR, the MAP process invokes the macro Check_Confirmation to check the content of the confirm. If the MAP_SEND_ROUTING_INFORMATION confirm from the HLR cannot be carried in a single TC-Result component, it is carried in one or more TC-Result-NL components (each sent in a TC-CONTINUE), followed by a TC-Result-L component in a TC-END message.

If the macro Check_Confirmation takes the OK exit, the MAP process sends a Send Routing Info ack containing the routing information received from the HLR to the call handling process in the GMSC and returns to the idle state.

**Earlier version MAP dialogue with the HLR**

If the macro Receive_Open_Cnf takes the Vr exit, the MAP process checks whether this is an OR interrogation (indicated by the inclusion of the OR interrogation parameter in the MAP_SEND_ROUTING_INFORMATION service request).

If this is not an OR interrogation, the GMSC performs the earlier version MAP dialogue as specified in [51] or [96] and the process returns to the idle state.

If this is an OR interrogation, the MAP process sends a Send Routeing Info negative response indicating OR not allowed to the call handling process in the GMSC and returns to the idle state.

**Dialogue opening failure**

If the macro Receive_Open_Cnf indicates that the dialogue with the HLR could not be opened, the MAP process sends an Abort to to the call handling process in the GMSC and returns to the idle state.

**Error in MAP_SEND_ROUTING_INFORMATION confirm**

If the MAP_SEND_ROUTING_INFORMATION service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a Send Routeing Info negative response to the call handling process in the GMSC and returns to the idle state.

**Call release**

If the call handling process in the GMSC indicates that the call has been aborted (i.e. prematurely released by the calling subscriber), the MAP process returns to the idle state. Any response from the HLR will be discarded.

**Abort of HLR dialogue**

After the dialogue with the HLR has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication, or the HLR may send a MAP_U_ABORT indication or a MAP_CLOSE indication. In any of these cases, the MAP process sends a Send Routeing Info negative response to the call handling process in the GMSC and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the HLR, sends a Send Routeing Info negative response indicating system failure to the call handling process in the GMSC and returns to the idle state.
Process SRI_GMSC

Figure 18.2/3: Process in the GMSC for retrieval of routing information

Signals to/from the left are to/from the GMSC call handling process; signals to/from the right are to/from the HLR.

1. **Idle**
2. **Send Routing Info**
3. **Receive Open_Cnf**

Section 21.1

**MAP_OPEN_req, MAP_SEND_ROUTING_INFORMATION_req, MAP_DELIMITER_req**

Section 21.2

**MAP_SEND_ROUTING_INFORMATION_cnf**

**Check Confirmation**

Section 21.2

**Send Routing Info ack**

**Send Routing Info negative response**

**Send Routing Info negative response**

**Perform MAP Vr Dialogue**

**Refer to the relevant version of GSM 09.02**

**OK**

**Abort**

**Idle**

**Wait_For_HLR_Response**

**OR interrogation?**

**Yes**

**No**

**Set negative response OR not allowed**

**Vr**

**OK**

**Provider Error, User Error, Data Error**

**Error**

**Vr**

**Yes**

**No**

Figure 18.2/3 (sheet 1 of 2): Process SRI_GMSC
Figure 18.2/3: Process in the GMSC for retrieval of routing information

Process SRI_GMSC

Figure 18.2/3 (sheet 2 of 2): Process SRI_GMSC
18.2.3 Procedures in the HLR

The MAP process in the HLR to retrieve routeing information for a mobile terminating call is shown in figure 18.2/4. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

<table>
<thead>
<tr>
<th>Macro Name</th>
<th>Subclause Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receive_Open_Ind</td>
<td>21.1.1</td>
</tr>
<tr>
<td>Receive_Open_Cnf</td>
<td>21.1.2</td>
</tr>
<tr>
<td>Check_Confirmation</td>
<td>21.2.2</td>
</tr>
</tbody>
</table>

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context locInfoRetrieval, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_SEND_ROUTING_INFORMATION service indication is received, the MAP process sends a Send Routeing Info request to the call handling process in the HLR, and waits for a response. The Send Routeing Info request contains the parameters received in the MAP_SEND_ROUTING_INFORMATION service indication.

If the call handling process in the HLR returns a Send Routeing Info ack, the MAP process constructs a MAP_SEND_ROUTING_INFORMATION service response containing the routeing information contained in the Send Routeing Info ack, constructs a MAP_CLOSE service request, sends them to the GMSC and returns to the idle state. If the MAP_SEND_ROUTING_INFORMATION response cannot be carried in a single TC-Result component, it is carried in one or more TC-Result-NL components (each sent in a TC-CONTINUE), followed by a TC-Result-L component in a TC-END message.

If the call handling process in the HLR returns a Provide Subscriber Info request, the MAP process requests a dialogue with the VLR whose identity is contained in the Provide Subscriber Info request by sending a MAP_OPEN service request, requests the subscriber status using a MAP_PROVIDE_SUBSCRIBER_INFO service request, and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request.

If the macro takes the OK exit, the MAP process waits for the response from the VLR.

If the MAP process receives a MAP_PROVIDE_SUBSCRIBER_INFO service confirm, it invokes the macro Check_Confirmation to check the content of the confirm.

If the Check_Confirmation macro takes the OK exit, the MAP process sends a Provide Subscriber Info ack containing the information received in the MAP_PROVIDE_SUBSCRIBER_INFO service confirm to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP_PROVIDE_SUBSCRIBER_INFO service confirm contains a provider error or a data error, the MAP process sends a Provide Subscriber Info negative response indicating the type of error to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

**NOTE:** The ‘User Error’ exit from the macro Check_Confirmation is shown for formal completeness; the MAP_PROVIDE_SUBSCRIBER_INFO_cnf primitive cannot contain a user error.

If the call handling process in the HLR returns a Provide Roaming Number request, the MAP process requests a dialogue with the VLR whose identity is contained in the Provide Roaming Number request by sending a MAP_OPEN service request, requests a roaming number using a MAP_PROVIDE_ROAMING_NUMBER service request, and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request.

If the macro takes the OK exit, the MAP process waits for the response from the VLR.

If the MAP process receives a MAP_PROVIDE_ROAMING_NUMBER service confirm, it invokes the macro Check_Confirmation to check the content of the confirm.

If the Check_Confirmation macro takes the OK exit, the MAP process sends a Provide Roaming Number ack containing the MSRN received in the MAP_PROVIDE_ROAMING_NUMBER service confirm to the call handling
process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP_PROVIDE_ROAMING_NUMBER service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a Provide Roaming Number negative response indicating the type of error to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

**Negative response from HLR call handling process**

If the call handling process in the HLR returns a negative response, either before or after a dialogue with the VLR to obtain a roaming number, the MAP process constructs a MAP_SEND_ROUTING_INFORMATION service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the GMSC and returns to the idle state.

**Earlier version MAP Provide Roaming Number dialogue with the VLR**

If the macro Receive_Open_Cnf takes the Vr exit after the MAP process has requested opening of a Provide Roaming Number dialogue with the VLR, the MAP process checks whether this is an OR interrogation (indicated by the inclusion of the OR interrogation parameter in the MAP_PROVIDE_ROAMING_NUMBER service request).

If this is not an OR interrogation, the HLR performs the earlier version MAP dialogue as specified in [51] or [96], relays the result of the dialogue to the HLR call handling process, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If this is an OR interrogation, the MAP process sends a Provide Roaming Number negative response indicating OR not allowed to the call handling process in the HLR and waits for a response. The handling of the response from the call handling process in the HLR is described above.

**Failure of Provide Subscriber Info dialogue with the VLR**

If the Receive_Open_Cnf macro takes the Vr exit or the Error exit after the MAP process has requested opening of a Provide Subscriber Info dialogue with the VLR, the MAP process sends a Provide Subscriber Info negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

**Failure of Provide Roaming Number dialogue with the VLR**

If the Receive_Open_Cnf macro takes the Error exit after the MAP process has requested opening of a Provide Roaming Number dialogue with the VLR, the MAP process sends a Provide Roaming Number negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP process receives a MAP_U_ABORT, a MAP_P_ABORT or a premature MAP_CLOSE from the MAP provider, it sends a Provide Roaming Number negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP process receives a MAP_NOTICE from the MAP provider, it returns a MAP_CLOSE request to the MAP provider, sends a Provide Roaming Number negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

**Earlier version MAP dialogue with the GMSC**

If the macro Receive_Open_Ind takes the Vr exit, the the HLR performs the earlier version MAP dialogue as specified in [51] or [96] and the process returns to the idle state.

**Failure of dialogue opening with the GMSC**

If the macro Receive_Open_Ind takes the Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.
If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.
Figure 18.2/4 (sheet 2 of 3): Process SRI_HLR
Figure 18.2/4 (sheet 3 of 3): Process SRI_HLR
18.2.4  Process in the VLR to provide a roaming number

The MAP process in the VLR to provide a roaming number for a mobile terminating call is shown in figure 18.2/5. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

\[
\text{Receive\_Open\_Ind} \quad \text{see subclause 21.1.1;}
\]

**Successful outcome**

When the MAP process receives a MAP_OPEN indication with the application context roamingNbEnquiry, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_PROVIDE_ROAMING_NUMBER service indication is received, the MAP process sends a Provide Roaming Number request to the call handling process in the VLR, and waits for a response. The Provide Roaming Number request contains the parameters received in the MAP_PROVIDE_ROAMING_NUMBER service indication.

If the call handling process in the VLR returns a Provide Roaming Number ack, the MAP process constructs a MAP_PROVIDE_ROAMING_NUMBER service response containing the roaming number contained in the Send Routeing Info ack, constructs a MAP_CLOSE service request, sends them to the HLR and returns to the idle state.

**Earlier version MAP dialogue with the HLR**

If the macro Receive_Open_Ind takes the Vr exit, the VLR performs the earlier version MAP dialogue as specified in [51] or [96] and the process returns to the idle state.

**Failure of dialogue opening with the HLR**

If the macro Receive_Open_Ind takes the Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.

**Negative response from VLR call handling process**

If the call handling process in the HLR returns a negative response, the MAP process constructs a MAP_PROVIDE_ROAMING_NUMBER service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the HLR and returns to the idle state.
Signals to/from the left are to/from the HLR; signals to/from the right are to/from the VLR call handling process.

Process PRN_VLR

Figure 18.2/6: Process in the VLR to handle a request for a roaming number

Figure 18.2/5: Process PRN_VLR
18.2.5 Process in the VLR to restore subscriber data

The MAP process in the HLR to restore subscriber data is shown in figure 18.2/6. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

- Receive_Open_Cnf see subclause 21.1.2;
- Check_Confirmation see subclause 21.2.2;
- Insert_Subs_Data_VLR see subclause 21.7.1;
- Activate_Tracing_VLR see subclause 21.9.3.

**Successful outcome**

When the MAP process receives a Restore Data request from the data restoration process in the VLR, it requests a dialogue with the HLR whose identity is contained in the Restore Data request by sending a MAP_OPEN service request, requests data restoration using a MAP_RESTORE_DATA service request and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the HLR.

The VLR may receive a MAP_INSERT_SUBSCRIBER_DATA service indication from the HLR; this is handled by the macro Insert_Subs_Data_VLR as described in subclause 21.7.1, and the MAP process waits for a further response from the HLR.

The VLR may receive a MAP_ACTIVATE_TRACE_MODE service indication from the HLR; this is handled by the macro Activate_Tracing_VLR as described in subclause 21.9.3, and the MAP process waits for a further response from the HLR.

If the MAP process receives a MAP_RESTORE_DATA service confirm, it invokes the macro Check_Confirmation to check the content of the confirm.

If the Check_Confirmation macro takes the OK exit, the MAP process sends a Restore Data ack containing the information received from the HLR to the data restoration process in the VLR and returns to the idle state.

**Error in MAP_RESTORE_DATA confirm**

If the MAP_RESTORE_DATA service confirm contains a user error or a provider error, or the macro Check_Confirmation indicates that there is a data error, the MAP process sends a Restore Data negative response indicating the type of error to the call handling process in the HLR, and returns to the idle state.

**Earlier version MAP dialogue with the HLR**

If the macro Receive_Open_Cnf takes the Vr exit, the VLR performs the earlier MAP version dialogue as specified in [51] or [96] and the process terminates.

**Dialogue opening failure**

If the macro Receive_Open_Cnf indicates that the dialogue with the HLR could not be opened, the MAP process sends a negative response indicating system failure to the data restoration process in the GMSC and returns to the idle state.
Figure 18.2/6: Process Restore_Data_VLR

Process Restore_Data_VLR

1. **Idle**

2. **Restore Data**

3. **Receive Open_Cnf**

   - **MAP_OPEN_req**
   - **MAP_RESTORE_DATA_req**
   - **MAP_DELIMITER_req**

4. **Wait For HLR_Response**

5. **Insert_Sub_Data_VLR**

6. **Activate_Tracing_VLR**

7. **Wait For HLR_Response**

8. **MAP_RESTORE_DATA_cnf**

9. **Check Confirmation**

   - **MAP_Vr_Dialogue**
   - **Idle**

10. **Restore Data ack**

11. **Idle**

12. **OK**

13. **Error**

   - **Vr Error**
   - **Provider Error, User Error, Data Error**

14. **Restore Data negative response**

Signals to/from the left are to/from the VLR data restoration process; signals to/from the right are to/from the HLR.
18.2.6 Process in the VLR to provide subscriber information

The MAP process in the VLR to provide subscriber information for a mobile terminating call subject to CAMEL invocation is shown in figure 18.2/6. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind see subclause 21.1.1;

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context subscriberInfoEnquiry, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_PROVIDE_SUBSCRIBER_INFO service indication is received, the MAP process sends a Provide Subscriber Info request to the subscriber information request process in the VLR, and waits for a response. The Provide Subscriber Info request contains the parameters received in the MAP_PROVIDE_SUBSCRIBER_INFO service indication.

If the subscriber information request process in the VLR returns a Provide Subscriber Info ack, the MAP process constructs a MAP_PROVIDE_SUBSCRIBER_INFO service response containing the information contained in the Provide Subscriber Info ack, constructs a MAP_CLOSE service request, sends them to the HLR and returns to the idle state.

Failure of dialogue opening with the HLR

If the macro Receive_Open_Ind takes the Vr exit or the Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.
Signals to/from the left are to/from the HLR; signals to/from the right are to/from the VLR subscriber information request process.

Figure 18.2/7: Process PSI_VLR

Figure 18.2/7: Process PSI_VLR
18.2.7 Process in the HLR for Any Time Interrogation

The message flows for successful retrieval of subscriber information related to an any time interrogation from the CAMEL server are shown in figure 18.2/8.

![Message flow for any time interrogation](image)

**Figure 18.2/8: Message flow for any time interrogation**

The following MAP services are used to retrieve routing information:

- `MAP_ANY_TIME_INTERROGATION` see subclause 6.11.1;
- `MAP_PROVIDE_SUBSCRIBER_INFO` see subclause 6.11.2;

18.2.7.1 Process in the gsmSCF

Out of the scope of the MAP specification.

18.2.8 Process in the HLR

The MAP process in the HLR to provide subscriber information in response to an interrogation from the CAMEL server is shown in figure 18.2/8. The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

- `Receive_Open_Ind` see subclause 21.1.1;
- `Receive_Open_Cnf` see subclause 21.1.2;
- `Check_Confirmation` see subclause 21.2.2.

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context `anyTimeInterrogationEnquiry`, it checks it by invoking the macro `Receive_Open_Ind`.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a `MAP_ANY_TIME_INTERROGATION` service indication is received, the MAP process sends an Any Time Interrogation request to the call handling process in the HLR (described in GSM 03.78), and waits for a response. The Any Time Interrogation request contains the parameters received in the `MAP_ANY_TIME_INTERROGATION` service indication.

If the call handling process in the HLR returns an Any Time Interrogation response, the MAP process constructs a `MAP_ANY_TIME_INTERROGATION` service response containing the subscriber information contained in the Any Time Interrogation response, constructs a `MAP_CLOSE` service request, sends them to the CAMEL server and returns to the idle state.

If the call handling process in the HLR returns a Provide Subscriber Info request, the MAP process requests a dialogue with the VLR whose identity is contained in the Provide Subscriber Info request by sending a MAP_OPEN service
request, requests the subscriber status using a MAP_PROVIDE_SUBSCRIBER_INFO service request, and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request.

If the macro takes the OK exit, the MAP process waits for the response from the VLR.

If the MAP process receives a MAP_PROVIDE_SUBSCRIBER_INFO service confirm, it invokes the macro Check_Confirmation to check the content of the confirm.

If the Check_Confirmation macro takes the OK exit, the MAP process sends a Provide Subscriber Info ack containing the information received in the MAP_PROVIDE_SUBSCRIBER_INFO service confirm to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

If the MAP_PROVIDE_SUBSCRIBER_INFO service confirm contains a provider error or a data error, the MAP process sends a Provide Subscriber Info negative response indicating the type of error to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

**NOTE:** The ‘User Error’ exit from the macro Check_Confirmation is shown for formal completeness; the MAP_PROVIDE_SUBSCRIBER_INFO_cnf primitive cannot contain a user error.

**Negative response from HLR call handling process**

If the call handling process in the HLR returns a negative response, either before or after a dialogue with the VLR to obtain subscriber information, the MAP process constructs a MAP_ANY_TIME_INTERROGATION service response containing the appropriate error, constructs a MAP_CLOSE service request, sends them to the CAMEL server and returns to the idle state.

**Failure of Provide Subscriber Info dialogue with the VLR**

If the Receive_Open_Cnf macro takes the Vr exit or the Error exit after the MAP process has requested opening of a Provide Subscriber Info dialogue with the VLR, the MAP process sends a Provide Subscriber Info negative response indicating system failure to the call handling process in the HLR, and waits for a response. The handling of the response from the call handling process in the HLR is described above.

**Failure of dialogue opening with the CAMEL server**

If the macro Receive_Open_Ind takes the Vr or Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.
Figure 18.2/9 (sheet 1 of 2): Process ATI_HLR (New)
Figure 18.2/9 (sheet 2 of 2): Process ATI_HLR (New)
18.3 Transfer of call handling

18.3.1 General

The message flow for successful transfer of call handling to forward a call is shown in figure 18.3/1.

NOTES:

xxx = Optional Procedure

TUP or ISUP may be used in signalling between MSCs, depending on the network type between the
MSCs. For further details on the TUP and ISUP procedures refer to the following CCITT
Recommendations & ETSI specification:

Q.721-725 - Telephone User Part (TUP);

ETS 300 356-1 - Integrated Services Digital Network (ISDN); Signalling System No.7; ISDN User Part (ISUP) version
2 for the international interface; Part 1: Basic services.

Figure 18.3/1: Message flow for transfer of call handling

If the HLR indicated in the response to the original request for routeing information that forwarding interrogation is
required, the GMSC executes the Send Routeing Information procedure with the HLR to obtain forwarding
information; otherwise the GMSC uses the forwarding data which were sent in the
MAP_RESUME_CALL_HANDLING req/ind.

18.3.2 Process in the VMSC

The MAP process in the VMSC to retrieve routeing information for a mobile terminating call is shown in figure 18.3/2.
The MAP process invokes macros not defined in this subclause; the definitions of these macros can be found as follows:

Receive_Open_Cnf see subclause 21.1.2;

Check_Confirmation see subclause 21.2.2.
Successful Outcome

When the MAP process receives a Resume Call Handling request from the call handling process in the VMSC, it requests a dialogue with the GMSC whose identity is contained in the Resume Call Handling request by sending a MAP_OPEN service request, requests routeing information using a MAP_RESUME_CALL_HANDLING service request and invokes the macro Receive_Open_Cnf to wait for the response to the dialogue opening request. If the dialogue opening is successful, the MAP process waits for a response from the GMSC.

If the MAP process receives a MAP_RESUME_CALL_HANDLING service confirm from the GMSC, the MAP process invokes the macro Check_Confirmation to check the content of the confirm.

If the macro Check_Confirmation takes the OK exit, the MAP process sends a Resume Call Handling ack to the call handling process in the VMSC and returns to the idle state.

Dialogue opening failure

If the macro Receive_Open_Cnf indicates that the dialogue with the GMSC could not be opened or that the dialogue can be opened only at an earlier version, the MAP process sends an Resume Call Handling negative response indicating system failure to the call handling process in the VMSC and returns to the idle state.

Error in MAP_RESUME_CALL_HANDLING confirm

If the MAP_RESUME_CALL_HANDLING service confirm contains a user error or a provider error, the MAP process sends a Resume Call Handling negative response to the call handling process in the VMSC and returns to the idle state.

NOTE: the ‘Data Error’ exit from the macro Check_Confirmation is shown for formal completeness; the result is empty, so the MAP_PROVIDE_SUBSCRIBER_INFO cnf primitive cannot contain a data error.

Abort of GMSC dialogue

After the dialogue with the GMSC has been established, the MAP service provider may abort the dialogue by issuing a MAP_P_ABORT indication, or the GMSC may send a MAP_CLOSE indication. In either of these cases, the MAP process sends a Resume Call Handling negative response to the call handling process in the GMSC and returns to the idle state.

If the MAP provider indicates a protocol problem by sending a MAP_NOTICE indication, the MAP process closes the dialogue with the GMSC, sends a Resume Call Handling negative response indicating system failure to the call handling process in the VMSC and returns to the idle state.
Figure 18.3/2: Process RCH_VMSC

Signals to/from the left are to/from the GMSC; signals to/from the right are to/from the call handling process in the VMSC.
18.3.3 Process in the GMSC

The MAP process in the GMSC to handle a request for the GMSC to resume call handling is shown in figure 18.3/3. The MAP process invokes a macro not defined in this subclause; the definition of this macro can be found as follows:

Receive_Open_Ind see subclause 21.1.1;

Successful outcome

When the MAP process receives a MAP_OPEN indication with the application context callControlTransfer, it checks it by invoking the macro Receive_Open_Ind.

If the macro takes the OK exit, the MAP process waits for a service indication.

If a MAP_RESUME_CALL_HANDLING service indication is received, the MAP process sends a Resume Call Handling request to the call handling process in the GMSC, and waits for a response. The Resume Call Handling request contains the parameters received in the MAP_RESUME_CALL_HANDLING service indication.

If the call handling process in the GMSC returns a Resume Call Handling ack, the MAP process constructs a MAP_RESUME_CALL_HANDLING service response, constructs a MAP_CLOSE service request, sends them to the HLR and returns to the idle state.

Failure of dialogue opening with the VMSC

If the macro Receive_Open_Ind takes the Vr exit or the Error exit, the MAP process returns to the idle state.

If the MAP provider sends a MAP_P_ABORT while the MAP process is waiting for a service indication, the MAP process returns to the idle state.

If the MAP provider sends a MAP_NOTICE while the MAP process is waiting for a service indication, the MAP process sends a MAP_CLOSE request to terminate the dialogue and returns to the idle state.
Figure 18.3/3: Process in the GMSC to handle a request to resume call handling

### Process RCH_GMSC

1. **Idle**
2. **Receive_Open_Ind**
3. **MAP_P_ABORT_ind**
   - **Idle**
4. **MAP_RESUME_CALL_HANDLING_ind**
5. **MAP_NOTICE_ind**
6. **MAP_CLOSE_req**
   - **Idle**
7. **Resume_Call_Handling**
   - **Wait_For_Call_Handling_Response**
   - **Resume_Call_Handling_ack**
   - **Resume_Call_Handling_negative_response**
   - **Set result**
   - **Set error**
8. **MAP_RESUME_CALL_HANDLING_rsp, MAP_CLOSE_rsp**
   - **Idle**

Signals to/from the left are to/from the GMSC call handling process; signals to/from the right are to/from the VMSC.
19 Supplementary services procedures

The following application contexts exist for handling of supplementary services:

- accessUnstructuredSsContext;
- accessFunctionalSsContext.

The accessUnstructuredSsContext refers to a simple MAP users, for which the corresponding MAP process can be identified by the MAP-Provider directly.

However, the accessFunctionalSsContext refers to a complex MAP-User consisting of several processes. For this user, a process co-ordinator is defined for each network entity, in order to identify the correct process to invoke. These processes open and validate the dialogue, then invoke the necessary operation-specific process. These processes are described below.

19.0 Functional supplementary service processes

19.0.1 Functional supplementary service process co-ordinator for MSC

Upon receipt of a CM-Service request with CM-service type = SS, the MSC initiates the process access request procedure towards the VLR as described in clause 21 of this ETS.

Once a CM connection is established, the MSC can handle supplementary service indications from the MS. Table 19.0/1 shows the co-ordinating process' reaction on receipt of specific SS service indications on the air interface. After the relevant process is invoked, the received air interface service indication is sent to that process. The creation of service requests on the basis of air interface messages is described in GSM 09.11.

Table 19.0/1: Relationship between received service indication and invoked process in the MSC

<table>
<thead>
<tr>
<th>Service indication received</th>
<th>Process invoked</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_REGISTER_SS_ind</td>
<td>REGISTER_SS_MSC</td>
</tr>
<tr>
<td>A_ERASE_SS_ind</td>
<td>ERASE_SS_MSC</td>
</tr>
<tr>
<td>A_ACTIVATE_SS_ind</td>
<td>ACTIVATE_SS_MSC</td>
</tr>
<tr>
<td>A_DEACTIVATE_SS_ind</td>
<td>DEACTIVATE_SS_MSC</td>
</tr>
<tr>
<td>A_INTERROGATE_SS_ind</td>
<td>INTERROGATE_SS_MSC</td>
</tr>
<tr>
<td>AREGISTER_PASSWORD</td>
<td>REGISTER_PASSWORD_MSC</td>
</tr>
</tbody>
</table>

Figure 19.0/1 shows the co-ordinating process in the MSC.
Figure 19.0/1: Supplementary Service Coordination process in the MSC, to identify which functional supplementary service process shall be invoked.

A_CM_SERV_REQ

Process_Access_Request_MSC

Wait_for_service_ind

null

A_REGISTER_SS_ind

Register_SS_MSC

A_ERASE_SS_ind

Erase_SS_MSC

A_ACTIVATE_SS_ind

Activate_SS_MSC

A_DEACTIVATE_SS_ind

Deactivate_SS_MSC

A_INTERROGATE_SS_ind

Interrogate_SS_MSC

A_REGISTER_PASSWORD_ind

Register_PASSWORD_MSC

A_REL_FROM_A_INTERFACE

A_REL_FROM_OFFSPRING

A_REL_A_INTERFACE

A_REL_A_OFFSPRING

A_DISC_TO_A_INTERFACE

A_DISC_TO_A_OFFSPRING

Relay_Info

* FROM A_INTERFACE

* TO OFFSPRING

* FROM A_INTERFACE

* TO OFFSPRING

* FROM A_INTERFACE

* TO OFFSPRING

null

null

null

null

Receive_errors_MSC

Null
19.0.2 Functional supplementary service process co-ordinator for VLR

Any functional SS process in the VLR starts by the VLR receiving the MAP_PROCESS_ACCESS_REQUEST indication. The VLR then acts as described in clause 21 of this ETS.

If the Process Access Request was successful, the VLR can handle supplementary service indications from the MSC. Table 19.0/2 shows the co-ordinating process’ reaction on receipt of specific SS service indications from the MSC. After the relevant process is invoked, the received service indication is sent to that process, and the co-ordinating process terminates.

Table 19.0/2: Relationship between received service indication and invoked process in the VLR

<table>
<thead>
<tr>
<th>Service indication received</th>
<th>Process invoked</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP_REGISTER_SS_ind</td>
<td>REGISTER_SS_VLR</td>
</tr>
<tr>
<td>MAP_ERASE_SS_ind</td>
<td>ERASE_SS_VLR</td>
</tr>
<tr>
<td>MAP_ACTIVATE_SS_ind</td>
<td>ACTIVATE_SS_VLR</td>
</tr>
<tr>
<td>MAP_DEACTIVATE_SS_ind</td>
<td>DEACTIVATE_SS_VLR</td>
</tr>
<tr>
<td>MAP_INTERROGATE_SS_ind</td>
<td>INTERROGATE_SS_VLR</td>
</tr>
<tr>
<td>MAP_REGISTER_PASSWORD</td>
<td>REGISTER_PASSWORD_VLR</td>
</tr>
</tbody>
</table>

Figure 19.0/2 shows the co-ordinating process in the VLR.
Figure 19.0/2: Supplementary Service Coordination process in the VLR, to open and process the access request from the MSC, and then identify which functional supplementary service process shall be invoked.

Figure 19.0/2 (sheet 1 of 2)
Figure 19.0/2 (sheet 2 of 2)
19.0.3 Functional supplementary service process co-ordinator for HLR

Any functional SS process in the HLR starts by the HLR receiving a MAP-OPEN service indication. If that service is successful, the HLR can handle supplementary service indications from the VLR. Table 19.0/3 shows the co-ordinating process' reaction on receipt of specific SS service indications from the VLR. After the relevant process is invoked, the received service indication is sent to that process, and the co-ordinating process terminates.

Table 19.0/3: Relationship between received service indication and invoked process in the HLR

<table>
<thead>
<tr>
<th>Service indication received</th>
<th>Process invoked</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP_REGISTER_SS_ind</td>
<td>REGISTER_SS_HLR</td>
</tr>
<tr>
<td>MAP_ERASE_SS_ind</td>
<td>ERASE_SS_HLR</td>
</tr>
<tr>
<td>MAP_ACTIVATE_SS_ind</td>
<td>ACTIVATE_SS_HLR</td>
</tr>
<tr>
<td>MAP_DEACTIVATE_SS_ind</td>
<td>DEACTIVATE_SS_HLR</td>
</tr>
<tr>
<td>MAP_INTERROGATE_SS_ind</td>
<td>INTERROGATE_SS_HLR</td>
</tr>
<tr>
<td>MAP_REGISTER_PASSWORD</td>
<td>REGISTER_PASSWORD_HLR</td>
</tr>
</tbody>
</table>

Figure 19.0/3 shows the co-ordinating process in the HLR.
Figure 19.0/3 (sheet 1 of 2)
Figure 19.0/3 (sheet 2 of 2)
19.1 Registration procedure

19.1.1 General

The registration procedure is used to register data related to a supplementary service in the HLR. The registration procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described in the subclauses below.

The registration procedure is shown in figure 19.1.1/1.

The following services may be used:

MAP_PROCESS_ACCESS_REQUEST (defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY (defined in clauses 7 and 21);
MAP_PROVIDE_IMSI (defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI (defined in clauses 6 and 21);
MAP_AUTHENTICATE (defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE (defined in clauses 6 and 21);
MAP_CHECK_IMEI (defined in clauses 6 and 21);
MAP_READY_FOR_SM (defined in clauses 10 and 21);
MAP_INSERT_SUBSCRIBER_DATA (defined in clauses 6 and 21);
MAP_REGISTER_SS (defined in clause 9).

NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: Services printed in italics are optional.

Figure 19.1.1/1: Interfaces and services for supplementary service registration
19.1.2 Procedures in the MSC

Supplementary service registration

The A_REGISTER_SS service indication received by the MAP user in the MSC contains the SS-Code and any parameters that are related to the supplementary service.

The MAP user transfers the received information to the VLR in the MAP_REGISTER_SS request without checking the contents of the service indication. Rules for the mapping are described in GSM 09.11.

The MSC then awaits the receipt of the MAP_REGISTER_SS confirm from the VLR. The outcome of the procedure is reported to the MS in the A_REGISTER_SS response message as described in GSM 04.8x, 04.9x and 09.11. Finally the SS-connection is released.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

Error handling

If at any time during the supplementary service part of this procedure a MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE or unexpected MAP_CLOSE indication is received from the VLR concerning the process, a CM_RELEASE_COMPLETE indication is sent to the MS (as specified in GSM 09.11). Upon receipt of a MAP_NOTICE indication from the VLR, the MSC must close the VLR dialogue by sending a MAP_CLOSE request. The process is then terminated.

If an A_CM_RELEASE indication is received from the MS, all open transactions shall be released using the MAP_U_ABORT request indicating application procedure cancellation, and the process is terminated.

The registration procedure in the MSC is shown in figure 19.1.2/1.
Figure 19.1.2/1
19.1.3 Procedures in the VLR

Supplementary service registration

When receiving the MAP_REGISTER_SS indication, the MAP user in the VLR transfers the information to the HLR in the MAP_REGISTER_SS request without checking the contents of the service indication.

The VLR then awaits the receipt of the MAP_REGISTER_SS confirm from the HLR. The MAP user in the VLR shall transfer the information contained in this primitive to the MSC in the MAP_REGISTER_SS response without checking its contents.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

Error handling

If at any time during this procedure a MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE or unexpected MAP_CLOSE indication is received from the MSC concerning the process, a MAP_U_ABORT request indicating application procedure cancellation is sent to the HLR (if a connection exists). If a MAP_NOTICE indication was received from the MSC, that dialogue must be closed by sending a MAP_CLOSE request towards the MSC. The process is terminated.

If a MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indication is received from the HLR, a MAP_U_ABORT request shall be sent to the MSC terminating the process. If a MAP_NOTICE indication was received from the HLR, that dialogue must be closed by sending a MAP_CLOSE request towards the HLR. The process terminates.

The registration procedure in the VLR is shown in figure 19.1.3/1.
Receive error from MSC

MAP REGISTER SS_req

MAP OPEN req

MAP REGISTER SS_req

MAP DISCONNECT req

Receive Open Confirm

Perform MAP V1 Dialogue

MAP U ABORT req

To HLR, including
- Destination reference = subscriber's IMSI
- Originating reference = VLR number

To HLR

From MSC

Figure 19.1.3/1 (sheet 1 of 2)
Figure 19.1.3/1 (sheet 2 of 2)
19.1.4 Procedures in the HLR

The procedure in the HLR is initiated when it receives a MAP_REGISTER_SS indication.

The HLR acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the Call Barred error should be returned to the VLR. The parameter "operatorBarring" shall be included with the error.

The supplementary service request shall then be processed according to GSM 03.11 and the 03.8x and 03.9x-series of technical specifications. This handling may lead to either a successful result, a partially successful result, or an error being returned.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11):

- if the VLR is to be updated after the supplementary service registration, the MAP_INSERT_SUBS_DATA_HLR process shall be initiated;

- if at any time during this procedure a MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indication concerning the process is received from the VLR, the process is terminated. If a MAP_NOTICE indication is received, a MAP_CLOSE request indicating sent towards the VLR.

The registration procedure in the HLR is shown in figure 19.1.4/1.
Figure 19.1.4/1 (sheet 1 of 2)
Figure 19.1.4/1 (sheet 2 of 2)
19.2 Erasure procedure

19.2.1 General

The erasure procedure is used to erase data related to a supplementary service in the HLR. The erasure procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described in the subclauses below.

The erasure procedure is shown in figure 19.2.1/1.

The following services may be used:

- MAP_PROCESS_ACCESS_REQUEST (defined in subclauses 6 and 21);
- MAP_TRACE_SUBSCRIBER_ACTIVITY (defined in clauses 7 and 21);
- MAP_PROVIDE_IMSI (defined in clauses 6 and 21);
- MAP_FORWARD_NEW_TMSI (defined in clauses 6 and 21);
- MAP_AUTHENTICATE (defined in clauses 6 and 21);
- MAP_SET_CIPHERING_MODE (defined in clauses 6 and 21);
- MAP_CHECK_IMEI (defined in clauses 6 and 21);
- MAP_READY_FOR_SM (defined in clauses 10 and 21);
- MAP_INSERT_SUBSCRIBER_DATA (defined in clauses 6 and 21);
- MAP_ERASE_SS (defined in clause 9).

NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: Services printed in *italics* are optional.

Figure 19.2.1/1: Interfaces and services for supplementary service erasure

19.2.2 Procedures in the MSC

The MSC procedures for erasure are identical to those specified for registration in subclause 19.1.2. The text and diagrams in subclause 19.1.2 apply with all references to registration changed to erasure.
### 19.2.3 Procedures in the VLR

The VLR procedures for erasure are identical to those specified for registration in subclause 19.1.3. The text and diagrams in subclause 19.1.3 apply with all references to registration changed to erasure.

### 19.2.4 Procedures in the HLR

The HLR procedure for erasure is identical to those specified for registration in subclause 19.1.4. The text and diagrams in subclause 19.1.4 apply with all references to registration changed to erasure.

### 19.3 Activation procedure

#### 19.3.1 General

The activation procedure is used to activate a supplementary service in the HLR. The activation procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described in the subclauses below.

The activation procedure is shown in figure 19.3.1/1.

The following services may be used:

- **MAP_PROCESS_ACCESS_REQUEST** (defined in clauses 6 and 21);
- **MAP_TRACE_SUBSCRIBER_ACTIVITY** (defined in clauses 7 and 21);
- **MAP_PROVIDE_IMSI** (defined in clauses 6 and 21);
- **MAP_FORWARD_NEW_TMSI** (defined in clauses 6 and 21);
- **MAP_AUTHENTICATE** (defined in clauses 6 and 21);
- **MAP_SET_CIPHERING_MODE** (defined in clauses 6 and 21);
- **MAP_CHECK_IMEI** (defined in clauses 6 and 21);
- **MAP_READY_FOR_SM** (defined in clauses 10 and 21);
- **MAP_GET_PASSWORD** (defined in clause 9);
- **MAP_INSERT_SUBSCRIBER_DATA** (defined in clauses 6 and 21);
- **MAP_ACTIVATE_SS** (defined in clause 9).
NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: Services printed in italics are optional.

**Figure 19.3.1/1: Interfaces and services for supplementary service activation**
19.3.2 Procedures in the MSC

The A_ACTIVATE_SS service indication received by the MAP user in the MSC contains the SS-Code and any parameters related to the supplementary service.

The MSC transfers the received information to the VLR in the MAP_ACTIVATE_SS request without checking the contents of the service indication. Rules for the mapping are described in GSM 09.11.

The MAP user may subsequently receive the MAP_GET_PASSWORD indication from the VLR. Upon receipt of this indication, the MSC sends the A_GET_PASSWORD message towards the MS and then awaits the response from the MS. When an A_GET_PASSWORD confirm message is received from the MS, the MSC initiates the MAP_GET_PASSWORD response towards the VLR without checking further the contents of the indication. Also see GSM 09.11.

The MSC will receive a MAP_ACTIVATE_SS confirm from the VLR. The outcome of the procedure is reported to the MS in the A_ACTIVATE_SS response message, see GSM 04.8x, 04.9x and 09.11. Finally the SS connection is released.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

The handling of MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE and unexpected MAP_CLOSE or A_CM_RELEASE in this procedure is identical to the handling in the Registration procedure in the MSC, see subclause 19.1.2 of this ETS.

The activation procedure in the MSC is shown in figure 19.3.2/1.
Figure 19.3.2/1
19.3.3 Procedures in the VLR

Supplementary service activation

When receiving the MAP_ACTIVATE_SS indication, the MAP user in the VLR transfers the information to the HLR in the MAP_ACTIVATE_SS request without checking the contents of the service indication.

The VLR may then receive the MAP_GET_PASSWORD indication. This information is transferred to the MSC in the MAP_GET_PASSWORD request. If a MAP_GET_PASSWORD confirm primitive is received from the MSC, the VLR initiates the MAP_GET_PASSWORD response towards the HLR.

The VLR will receive the MAP_ACTIVATE_SS confirm from the HLR. The MAP user in the VLR shall transfer the information contained in this primitive to the MSC in the MAP_ACTIVATE_SS response without checking its contents.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

Error handling

The handling of MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE and unexpected MAP_CLOSE in this procedure is identical to the handling in the Registration procedure in the VLR, see subclause 19.1.3 of this ETS.

The activation procedure in the VLR is shown in figure 19.3.3/1.
MAP_ACTIVATE_S6_ind

MAP_OPEN_req

MAP_ACTIVATE_S6_req

MAP_DELIMITER_req

Receive_Open_Confirm

From MSC

OK

Wait_for_ss_on

Performs_MAP_V1_Dialogue

MAP_U_ABORT_req

Err

To MSC

Err

Section 21.1

Figure 19.10.2/2

Figure 19.3.3/1 (sheet 1 of 2)
Figure 19.3.3/1 (sheet 2 of 2)
19.3.4 Procedures in the HLR

The procedure in the HLR is initiated when it receives a MAP_ACTIVATE_SS indication.

The HLR acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the Call Barred error should be returned to the VLR. The parameter "operatorBarring" shall be included with the error.

The supplementary service request shall then be processed according to GSM 03.11 and the 03.8x and 03.9x-series of technical specifications. This handling may lead to either a successful result, a partially successful result, or an error being returned.

During the handling of activation, the get password procedure may be initiated (as specified in GSM 03.11). This will involve the sending of a MAP_GET_PASSWORD request to the VLR.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11):

- if the VLR is to be updated after the supplementary service activation, the MAP_INSERT_SUBS_DATA_HLR process is initiated;

- handling of receipt of MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indications from the VLR is identical to their handling in the registration procedure, see subclause 19.1.4 above.

The activation procedure in the HLR is shown in figure 19.3.4/1.
Figure 19.3.4/1 (sheet 1 of 2)
Figure 19.3.4/1 (sheet 2 of 2)
19.4 Deactivation procedure

19.4.1 General

The deactivation procedure is used to deactivate a supplementary service in the HLR. The deactivation procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described in the subclauses below.

The deactivation procedure is shown in figure 19.4.1/1.

The following services may be used:

- MAP_PROCESS_ACCESS_REQUEST (defined in clauses 6 and 21);
- MAP_TRACE_SUBSCRIBER_ACTIVITY (defined in clauses 7 and 21);
- MAP_PROVIDE_IMSI (defined in clauses 6 and 21);
- MAP_FORWARD_NEW_TMSI (defined in clauses 6 and 21);
- MAP_AUTHENTICATE (defined in clauses 6 and 21);
- MAP_SET_CIPHERING_MODE (defined in clauses 6 and 21);
- MAP_CHECK_IMEI (defined in clauses 6 and 21);
- MAP_READY_FOR_SM (defined in clauses 10 and 21);
- MAP_GET_PASSWORD (defined in clause 9);
- MAP_INSERT_SUBSCRIBER_DATA (defined in clauses 6 and 21);
- MAP_DEACTIVATE_SS (defined in clause 9).

NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: Services printed in *italics* are optional.

Figure 19.4.1/1: Interfaces and services for supplementary service deactivation.
19.4.2 Procedures in the MSC

The MSC procedures for deactivation are identical to those specified for activation in subclause 19.3.2. The text and diagrams in subclause 19.3.2 apply with all references to activation changed to deactivation.

19.4.3 Procedures in the VLR

The VLR procedures for deactivation are identical to those specified for activation in subclause 19.3.3. The text and diagrams in subclause 19.3.3 apply with all references to activation changed to deactivation.

19.4.4 Procedures in the HLR

The HLR procedures for deactivation are identical to those specified for activation in subclause 19.3.4. The text and diagrams in subclause 19.3.4 apply with all references to activation changed to deactivation.

19.5 Interrogation procedure

19.5.1 General

The interrogation procedure is used to retrieve information related to a supplementary service from the VLR or the HLR. It is the VLR which decides whether an interrogation request should be forwarded to the HLR or not. Some non-supplementary service related services may be invoked as a result of the procedure, as described in the subclauses below.

The interrogation procedure is shown in figure 19.5.1/1.

The following services may be used:

- MAP_PROCESS_ACCESS_REQUEST (defined in clauses 6 and 21);
- MAP_TRACE_SUBSCRIBER_ACTIVITY (defined in clauses 7 and 21);
- MAP_PROVIDE_IMSI (defined in clauses 6 and 21);
- MAP_FORWARD_NEW_TMSI (defined in clauses 6 and 21);
- MAP_AUTHENTICATE (defined in clauses 6 and 21);
- MAP_SET_CIPHERING_MODE (defined in clauses 6 and 21);
- MAP_CHECK_IMEI (defined in clauses 6 and 21);
- MAP_READY_FOR_SM (defined in clauses 10 and 21);
- MAP_INTERROGATE_SS (defined in clause 9).
NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: Services printed in italics are optional.

Figure 19.5.1/1: Interfaces and services for supplementary service interrogation

19.5.2 Procedures in the MSC

The MSC procedures for interrogation are identical to those specified for registration in subclause 19.1.2. The text and diagrams in subclause 19.1.2 apply with all references to registration changed to interrogation.

19.5.3 Procedures in the VLR

Supplementary service interrogation

When receiving the MAP_INTERROGATE_SS indication, the MAP user acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the error Call Barred is returned to the MSC. The parameter "operatorBarring" shall be included with the error.

The interrogation is either answered by the VLR or by the HLR, depending on the service interrogated.

a) Interrogation to be handled by the VLR

The supplementary service request shall then be processed according to GSM 03.11 and the 03.8x and 03.9x-series of technical specifications. This handling may lead to either a successful result, a partially successful result, or an error being returned.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

b) Interrogation to be handled by HLR

If the interrogation is to be handled by the HLR, on receiving the MAP_INTERROGATE_SS indication, the MAP user in the VLR transfers the information to the HLR in the MAP_INTERROGATE_SS request without further checking the contents of the service indication.

The VLR will receive the MAP_INTERROGATE_SS confirm from the HLR. The MAP user in the VLR shall transfer the information contained in this primitive to the MSC in the MAP_INTERROGATE_SS response without checking its contents.

For call independent SS operations, each message shall only contain a single component. Messages which contain more than one component will be stopped at the air interface (as specified in GSM 09.11).

Error handling

Handling of MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE and unexpected MAP_CLOSE in this procedure is identical to the handling in the Registration procedure in the VLR, subclause 19.1.3. The Interrogation procedure is described in figure 19.5.3/1.
Figure 19.5.3/1 (sheet 1 of 3)
MAP_OPEN_req
To HLR, including
Destination reference = subscriber's IMSI
Originating reference = VLR number

MAP_INTERROGATE_SS_req
To HLR

MAP_DELIMITER_req
To HLR

Receive_Open_Confirm
Section 21.1

Wait_for_sa_off

Perform_MAP_V1_Dialogue

MAP_U_ABORT_req
To MSC

Figure 19.5.3/1 (sheet 2 of 3)
Figure 19.5.3/1 (sheet 3 of 3)
19.5.4 Procedures in the HLR

When receiving the MAP_INTERROGATE_SS indication, the MAP user acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the error Call Barred is returned to the MSC. The parameter "operatorBarring" shall be included with the error;

- if the supplementary service is not supported in HLR the error Unexpected Data Value is returned to the VLR.

The interrogation is either answered by the VLR or by the HLR, depending on the service interrogated.

a) Interrogation to be handled by the VLR

If the interrogation procedure should have been answered by the VLR, then the HLR assumes that the VLR does not support the interrogated supplementary service, and returns the SS Not Available error to the VLR.

b) Interrogation to be handled by HLR

The supplementary service request shall be processed according to GSM 03.11 and the 03.8x and 03.9x-series of technical specifications. This handling may lead to either a successful result or an error being returned.

For call independent SS operations, each message shall only contain a single component.

Error handling

Handling of MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE and unexpected MAP_CLOSE in this procedure is identical to the handling in the Registration procedure in the VLR, subclause 19.1.3. The Interrogation procedure is described in figure 19.5.4/1.
Figure 19.5.4/1
19.6 Invocation procedure

19.6.1 General

The invocation procedure is used to check subscription data in the VLR for certain supplementary services which are invoked after the call set-up phase is finished. For invocation of supplementary services which are invoked during the call set-up phase, please refer to the Call Handling procedure descriptions.

The invocation procedure is shown in figure 19.6.1/1. Note that some optional services may be invoked in connection with this procedure, as described in the subclause below.

The following services are used:

MAP_PROCESS_ACCESS_REQUEST (defined in clauses 6 and 21);
MAP_TRACE_SUBSCRIBER_ACTIVITY (defined in clauses 7 and 21);
MAP_PROVIDE_IMSI (defined in clauses 6 and 21);
MAP_FORWARD_NEW_TMSI (defined in clauses 6 and 21);
MAP_AUTHENTICATE (defined in clauses 6 and 21);
MAP_SET_CIPHERING_MODE (defined in clauses 6 and 21);
MAP_CHECK_IMEI (defined in clauses 6 and 21);
MAP_READY_FOR_SM (defined in clauses 10 and 21);
MAP_INVOKE_SS (defined in clause 9).

NOTE 1: For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines indicate the trigger provided by the signalling on the radio path, and the signalling triggered on the radio path.

NOTE 2: For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

NOTE 3: A_INVOKESS is a generic message to illustrate any supplementary service invocation request message on the air interface, e.g. BuildMPTY, see GSM 04.80.

Figure 19.6.1/1: Interfaces and services for supplementary service invocation
19.6.2 Procedures in the MSC

Process access request

Before the Call Hold or Multi-Party supplementary services can be invoked, a CC connection must be established between the MS and the MSC as described in GSM 04.08 and the Call Handling procedure descriptions within this ETS.

When an A_INVOKE_SS request message arrives at the MSC during a call (as described in GSM 04.10, 04.8x and 04.9x-series of technical specifications), then if control of subscription to the invoked supplementary service is required, the MSC initiates the process access request procedure towards the VLR as described in clause 21 of this ETS.

Supplementary service invocation

If the Process Access Request procedure towards the VLR is successful, the MSC shall forward a MAP_INVOKE_SS service request towards the VLR. This request shall contain the SS-Code of the supplementary service to be invoked, and possibly the Basic service code. Mapping from the A_INVOKE_SS to this service request is described in GSM 09.11.

The MSC will receive a MAP_INVOKE_SS confirm from the VLR. If the outcome of the service is successful (i.e. the service confirm is empty), the MSC will invoke the requested supplementary service as described in GSM 02.8x-series, 03.8x and 03.9x-series of technical specifications. If the outcome of the service is unsuccessful, the MSC shall send an appropriate A_INVOKE_SS response towards the MS. The structure of this message is described in GSM 09.11 and 04.8x and 04.9x-series of technical specifications.

Error handling

If at any time during this procedure a MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE or MAP_CLOSE indication concerning the process is received from the VLR, the process is terminated. If a MAP_NOTICE indication was received from the VLR, the VLR dialogue must also be aborted by sending a MAP_U_ABORT request indicating Procedure error towards the VLR. Possible signalling to the MS is described in GSM 04.10.

If an A_CM_RELEASE indication is received from the MS, all open transactions are released using the MAP_U_ABORT request indicating application procedure cancellation; the process terminates.

The invocation procedure in the MSC is shown in figure 19.6.2/1.
Figure 19.6.2/1 (sheet 1 of 2)

Note 1: Fictitious signal to indicate receipt/sending of SS invocation invoke component on the air interface (e.g., BuildMPTY). Described in GSM 04.8X and 09.11.
Note 1: Fictitious signal to indicate receipt/sending of SS invocation invoke component on the air interface, (eg. BuildMPTY). Described in GSM 04.8X and 09.11.

Figure 19.6.2/1 (sheet 2 of 2)
19.6.3 Procedures in the VLR

Process Access Request

When receiving the MAP_PROCESS_ACCESS_REQUEST indication, the VLR acts as described in clause 21 of this ETS.

Supplementary service invocation

When receiving the MAP_INVOKE_SS indication, the MAP user acts as follows:

- if the operator has barred the subscriber from access to supplementary services, the error "Call Barred" is returned to the MSC. The parameter "operatorBarring" shall be included with the error;

- if any irrelevant information elements (according to the service description) or invalid information element values are present in the service request, then the unexpected data value error is returned to the MSC in the MAP_INVOKE_SS response;

- if the VLR does not support the invoked supplementary service then the VLR shall respond with the SS Not Available error;

- if the requested supplementary service cannot be invoked by subscriber actions, then the VLR shall respond with the Illegal SS Operation error;

- if the subscriber is not provided with (i.e. subscribed to) the requested supplementary service, then the SS error status error (possibly including the SS-Status as parameter) is returned to the MSC in the MAP_INVOKE_SS response.

If all checks are passed the VLR returns an empty MAP_INVOKE_SS response to the MSC, thus indicating that the invocation request was accepted.

If at any time during this procedure a MAP_P_ABORT, MAP_U_ABORT, MAP_NOTICE or unexpected MAP_CLOSE indication concerning the process is received from the MSC, the process terminates. If a MAP_NOTICE indication was received from the MSC, that dialogue must be aborted by sending a MAP_U_ABORT request indicating Procedure error towards the MSC. The process terminates.

The invocation procedure in the VLR is shown in figure 19.6.3/1.
Figure 19.6.3/1
19.7 Password registration procedure

19.7.1 General

The password registration procedure is used to register a password in the HLR. The password registration procedure is a fully transparent communication between the MS and the HLR, except that some services may be invoked as a result of the procedure, as described below.

The password registration procedure is shown in figure 19.7.1/1.

The following services may be used:

- **MAP_PROCESS_ACCESS_REQUEST** (defined in clauses 6 and 21);
- **MAP_TRACE_SUBSCRIBER_ACTIVITY** (defined in clauses 7 and 21);
- **MAP_PROVIDE_IMSI** (defined in clauses 6 and 21);
- **MAP_FORWARD_NEW_TMSI** (defined in clauses 6 and 21);
- **MAP_AUTHENTICATE** (defined in clauses 6 and 21);
- **MAP_SET_CIPHERING_MODE** (defined in clauses 6 and 21);
- **MAP_CHECK_IMEI** (defined in clauses 6 and 21);
- **MAP_READY_FOR_SM** (defined in clauses 10 and 21);
- **MAP_GET_PASSWORD** (defined in clause 9).

### Figure 19.7.1/1: Interfaces and services for supplementary service password registration

---

<table>
<thead>
<tr>
<th>MS</th>
<th>MSC</th>
<th>B</th>
<th>VLR</th>
<th>D</th>
<th>HLR</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>A CM SERV REQ</th>
<th>MAP PROCESS ACCESS REQ</th>
<th>MAP REGISTER PW</th>
<th>MAP REGISTER PW</th>
<th>MAP REGISTER PW</th>
</tr>
</thead>
<tbody>
<tr>
<td>(note 1)</td>
<td>(note 2)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A REGISTER PW</th>
<th>MAP REGISTER PW</th>
<th>MAP REGISTER PW</th>
</tr>
</thead>
<tbody>
<tr>
<td>A GET PW</td>
<td>MAP GET PW</td>
<td>MAP GET PW</td>
</tr>
<tr>
<td>A GET PW ack</td>
<td>MAP GET PW ack</td>
<td>MAP GET PW ack</td>
</tr>
<tr>
<td>A REGISTER PW</td>
<td>MAP REGISTER PW</td>
<td>MAP REGISTER PW</td>
</tr>
</tbody>
</table>

### Notes:

**NOTE 1:** For details of the procedure on the radio path, see GSM 04.08, 04.10, 04.8x and 04.9x. Services shown in dotted lines are triggers/triggered signalling on the radio path.

**NOTE 2:** For details on the Process Access Request procedure, please refer to clause 21 in this ETS.

**NOTE 3:** Use of each of the three MAP_GET_PASSWORD operations is described in subclause 19.7.4.
19.7.2 Procedures in the MSC

The password registration procedure in the MSC is identical to that for activation specified in subclause 19.3.2. All the text and diagrams in subclause 19.3.2 apply with all references to activation changed to password registration.

19.7.3 Procedures in the VLR

The password registration procedure in the VLR is identical to that for activation specified in subclause 19.3.3. All the text and diagrams in subclause 19.3.3 apply with all references to activation changed to password registration.

19.7.4 Procedures in the HLR

The procedure in the HLR is initiated when it receives a MAP_REGISTER_PASSWORD indication.

The HLR acts as follows:

- if the operator has barred the subscriber for access to supplementary services, the Call Barred error is returned to the VLR. The parameter "operatorBarring" shall be included with the error;

- if any irrelevant information elements (according to the service description) or invalid information element values are present, then the unexpected data value error is returned to the VLR in the response. This error should thus be returned if the SS-Code provided by the mobile subscriber is not allocated.

The HLR shall then process the MAP_REGISTER_PASSWORD indication as specified in GSM 03.11. During the handling of password registration, the password procedure will be initiated (as specified in GSM 03.11) This will involve the sending of MAP_GET_PASSWORD requests to the VLR.

- Handling of receipt of MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indications from the VLR is identical to their handling in the registration procedure, see subclause 19.1.4 above.

The password registration procedure in the HLR is shown in figure 19.7.4/1.
Operator determined barring of SS Management

Map\_REGISTER\_PASSWORD\_ind

Check\_Indication

SS\_Register\_Password

Wait\_for\_PW

---

Figure 19.7.4/1 (sheet 1 of 2)
All these indications are from process PW1, PW2, PW3 or PW4.

Figure 19.7.4/1 (sheet 2 of 2)
19.8 Mobile Initiated USSD procedure

19.8.1 General

The procedure supports supplementary service signalling procedures which can allow PLMN specific services to be introduced.

The message flow for the procedure can be found in GSM 03.90.

The following services may be used:

- MAP_PROCESS_ACCESS_REQUEST (defined in clauses 6 and 21);
- MAP_TRACE_SUBSCRIBER_ACTIVITY (defined in clauses 7 and 21);
- MAP_PROVIDE_IMSI (defined in clauses 6 and 21);
- MAP_FORWARD_NEW_TMSI (defined in clauses 6 and 21);
- MAP_AUTHENTICATE (defined in clauses 6 and 21);
- MAP_SET_CIPHERING_MODE (defined in clauses 6 and 21);
- MAP_CHECK_IMEI (defined in clauses 6 and 21);
- MAP_READY_FOR_SM (defined in clauses 10 and 21);
- MAP_UNSTRUCTURED_SS_REQUEST (defined in clause 9);
- MAP_UNSTRUCTURED_SS_NOTIFY (defined in clause 9).

The following service is certainly used:

- MAP_PROCESS_UNSTRUCTURED_SS_REQUEST (defined in clause 9).

19.8.2 Procedures in the MSC

Before the Process Unstructured SS Request service can be invoked, a call independent CM connection must be created between the MS and the MSC.

Once a CM-connection is established, the MSC may handle the A_PROCESS_UNSTRUCTURED_SS_REQUEST from the MS. This message contains information input by the user, the message may be fed to an application contained locally in the MSC or to the VLR. The rules for determining this are specified in GSM 03.90.

1) Message Destined for VLR

If the message is destined for the VLR then the MSC shall transfer the message to the VLR using the mapping specified in detail in GSM 09.11.

The MSC may subsequently receive one or more MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications from the VLR. These shall be sent transparently to the MS. When a confirmation is received from the MS this shall be returned to the VLR.

When the MSC receives a MAP_PROCESS_UNSTRUCTURED_SS_REQUEST confirmation from the VLR then it shall pass this to the MS and initiate release of the CM connection.
2) Message Destined for Local Application

If the message is destined for the local USSD application then the MSC shall transfer the message to the application.

The MSC may subsequently receive one or more requests from the application which correspond to the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications. These shall be sent transparently to the MS. When a confirmation is received from the MS this shall be returned to the application.

When the MSC receives the result of the original operation from the application then it shall pass this to the MS and initiate release of the CM connection.

Error Handling

Both the MS and the VLR or USSD Application may initiate release of the CM-connection at any time. This is handled as shown in the diagrams.

The procedure in the MSC is shown in figure 19.8.2/1.
Null

A_CM_SERV_REQ

GSM 04.08, 04.10

Process_Access_Request_MSC

Section 21.4

OK

Wait_for_service_ind

Error

Wait_for_service_ind

NULL

NULL

A_PROCESS_UNSTD_SS_REQUEST_ind

Handle at MSC? (See GSM 03.90)

no

yes

MAP_PROCESS_UNSTD_SS_REQUEST_req

MAP_DELIMITER_req

1

Wait_for_pussd_cnf

Wait_for_service_ind

Wait_for_service_ind

NULL

NULL

Receive_Errors_MSC

NULL

MAP_ERROR

Yes

no

A_PROCESS_UNSTD_SS_REQUEST_rep

A_CM_REL_COMP

NULL

NULL

Section 19.10

Receive_Errors_MSC

NULL

MAP_U_ABORT_req

A_PROCESS_UNSTD_SS_NOTIFY_rep

A_CM_REL_COMP

NULL

NULL

Section 19.10

Receive_Errors_MSC

NULL

MAP_U_ABORT_req

A_PROCESS_UNSTD_SS_NOTIFY_rep

A_CM_REL_COMP

NULL

NULL

Section 19.10

Figure 19.8.2/1 (sheet 1 of 2)
Figure 19.8.2/1 (sheet 2 of 2)
19.8.3 Procedures in the VLR

The initiation of the process is shown in subclause 19.0.2.

Once a MAP dialogue is established, the VLR may handle the MAP_PROCESS_UNSTRUCTURED_SS_REQUEST from the MSC. This message contains information input by the user, the message may be fed to an application contained locally in the VLR or to the HLR. The rules for determining this are specified in GSM 03.90.

Message Destined for HLR

If the message is destined for the HLR then the VLR shall transfer the message transparently to the HLR.

The VLR may subsequently receive one or more MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications from the HLR. These shall be sent transparently to the MSC. When a confirmation is received from the MSC this shall be returned to the HLR.

When the VLR receives a MAP_PROCESS_UNSTRUCTURED_SS_REQUEST confirmation from the HLR then it shall pass this to the MS and close the MAP provider service.

Message Destined for Local Application

If the message is destined for the local USSD application then the VLR shall transfer the message to the application.

The VLR may subsequently receive one or more requests from the application which correspond to the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications. These shall be sent transparently to the MSC. When a confirmation is received from the MSC this shall be returned to the application.

When the VLR receives the result of the original operation from the application then it shall pass this to the MSC and initiate release of the CM connection.

Error Handling

Both the MSC and the HLR or USSD Application may initiate release of the MAP service at any time. This is handled as shown in the diagrams.

The procedure in the VLR is shown in figures 19.8.3/1 and 19.8.3/2.
Figure 19.8.3/1 (sheet 1 of 3)
Figure 19.8.3/1 (sheet 2 of 3)
Arrows to left are to MSC, arrows to right are to USSD application unless otherwise stated.

Figure 19.8.3/1 (sheet 3 of 3)
Figure 19.8.3/2
19.8.4 Procedures in the HLR

The initiation of the process is shown in subclause 19.0.3.

Once a MAP dialogue is established, the HLR may handle the MAP_PROCESS_UNSTRUCTURED_SS_REQUEST from the VLR. This message contains information input by the user. If the alphabet used for the message is understood then the message shall be fed to an application contained locally in the HLR. If the alphabet is not understood then the error "UnknownAlphabet" shall be returned.

The HLR may subsequently receive one or more requests from the application which correspond to the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY indications. These shall be sent transparently to the VLR. When a confirmation is received from the VLR this shall be returned to the application.

When the HLR receives the result of the original operation from the application then it shall pass this to the VLR and initiate release of the CM connection.

Error Handling

Both the VLR and the USSD Application may initiate release of the MAP service at any time. This is handled as shown in the diagrams.

The procedure in the HLR is shown in figure 19.8.4/1.
Figure 19.8.4/1 (sheet 1 of 2)
Figure 19.8.4/1 (sheet 2 of 2)
19.9  Network initiated USSD procedure

19.9.1  General

The procedure supports supplementary service signalling procedures which can allow PLMN specific services to be introduced.

The message flow for the procedure can be found in GSM 03.90.

The following services may be used:

- MAP_PAGE (defined in clauses 6 and 21);
- MAP_SEARCH_FOR_MOBILE_SUBSCRIBER (defined in clauses 6 and 21);
- MAP_PROCESS_ACCESS_REQUEST (defined in clauses 6 and 21);
- MAP_AUTHENTICATE (defined in clauses 6 and 21);
- MAP_SET_CIPHERING_MODE (defined in clauses 6 and 21);
- MAP_FORWARD_NEW_TMSI (defined in clauses 6 and 21);
- MAPREADY_FOR_SM (defined in clauses 10 and 21).

At least one of the following services will certainly be used, and both may be used:

- MAP_UNSTRUCTURED_SS_REQUEST (defined in clause 9);
- MAP_UNSTRUCTURED_SS_NOTIFY (defined in clause 9).

19.9.2  Procedure in the MSC

The procedure may be invoked either by the VLR or by a USSD application local to the MSC. They may start by using either the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY service. If the request is initiated by a local USSD application then the MSC will open a dialogue with the HLR.

In both cases the MSC will initiate a CM connection to the MS (using the page or search macros defined in subclause 21.3). Once the connection is successfully established the message received from the VLR or USSD application will be sent to the MS using the mapping specified in GSM 09.11.

Following transfer of the message the MSC will wait for a confirmation from the MS. This will be sent to the VLR or USSD application as appropriate.

Following this, the MSC may receive further uses of the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY services, or may receive an indication to release the connection to the MS.

In the event of an error, the connection to the MS shall be released, and the MAP process with the VLR shall be aborted as shown in the diagram.

The procedure in the MSC is shown in figure 19.9.2/1.
Figure 19.9.2/1 (sheet 1 of 4)
Figure 19.9.2/1 (sheet 2 of 4)
Arrows to left are to MS, arrows to right are to USSD application unless otherwise stated.

Figure 19.9.2/1 (sheet 4 of 4)
19.9.3 Procedure in the VLR

The procedure may be invoked either by the HLR or by a USSD application local to the VLR. They may start by using either the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS NOTIFY service.

In both cases the VLR will first initiate a MAP dialogue with the MSC. When the indication for the unstructured SS request or notify is received then the macro Start_USSD_VLR will be used to page the MS and open a CM connection. Once the CM connection is successfully established the indication received from the HLR or USSD application will be sent to the MSC.

Following transfer of the message the VLR will wait for a confirmation from the MSC. This will be sent to the HLR or USSD application as appropriate.

Following this, the VLR may receive further uses of the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY services, or may receive a MAP_CLOSE_ind.

In the event of an error, the MAP process with the MSC shall be released, and if necessary the MAP process with the HLR shall be aborted as shown in the diagram.

The procedure in the VLR is shown in figure 19.9.3/1.

MSC Initiated USSD

If a USSD application in the MSC wishes to use the network initiated USSD procedure, and a connection to the MS does not exist then it shall open a dialogue to the VLR. This dialogue will automatically lead to the VLR performing page and search using the macro Start_USSD_VLR.

Macro Start_USSD_VLR

This macro is used to initiate a CM connection with the MS for transfer of network initiated unstructured SS data.

It first checks for correct data in the VLR. If a problem is found then "Err" is returned.

A page or search procedure (as appropriate) will then be used to contact the MS. Following successful page or search the macro Process_Access_Request_VLR specified in subclause 21.4 will be used to handle the CM connection establishment.

The macro is shown in figure 19.9.3/2.
Figure 19.9.3/1 (sheet 1 of 4)
Arrows to left are to MSC, arrows to right are to HLR unless otherwise stated.

Figure 19.10.2/2

Receive_error_from_HLR

MAP_CLOSE_ind

MAP_CLOSE_req

NULL

Wait_for_ussd_or_close

MAP_UNSTD_SS_NOTIFY_ind

MAP_UNSTD_SS_NOTIFY_req

MAP_DELIMITER_req

Wait_for_ussd_or_close

Receive_error_from_MSC

MAP_U_ABORT_req

NULL

Section 19.8.3

MAP_UNSTD_SS_NOTIFY_cnf

MAP_UNSTD_SS_NOTIFY_rsp

MAP_DELIMITER_req

Wait_for_ussd_or_close

MAP_UNSTD_SS_NOTIFY_cnf

MAP_UNSTD_SS_NOTIFY_rsp

MAP_DELIMITER_req

Wait_for_ussd_or_close

Receive_Error_at_VLR

NULL

Section 19.8.3

MAP_UNSTD_SS_NOTIFY_cnf

MAP_UNSTD_SS_NOTIFY_rsp

MAP_DELIMITER_req

Wait_for_ussd_or_close

Receive_Error_at_VLR

NULL

Section 19.8.3

Figure 19.10.2/3

Figure 19.10.2/2

Figuur 19.9.3/1 (sheet 2 of 4)
Figure 19.9.3/1 (sheet 3 of 4)
Figure 19.9.3/1: Handling of network initiated USSD at VLR

**Process NW_INIT_USSD_VLR**

1. **MAP_OPEN_req**
   - **MAP_DELIMITER_req**
   - **Wait for USSD_ind**
     - Arrows to left are to MSC.
     - Arrows to right are to HLR unless otherwise stated.

2. **Receive_Open_Confirm**
   - **Section 21.1**
   - **MAP_U_ABORT_req**
     - **NULL**
     - **OK**

3. **From HLR**
   - **MAP_NOTICE_ind**
   - **MAP_CLOSE_req**
     - **NULL**

4. **From HLR**
   - **MAP_UNST'D_SS_NOTIFY_ind**
   - **Start_USSD_VLR**
     - **OK**
     - **Local Err**

5. **MAP_UNST'D_SS_NOTIFY_req**
   - **MAP_DELIMITER_req**
   - **Wait for ussdh_cnf**
     - **Sheet 2**
     - **NULL**

6. **MAP_CLOSE_req**
   - **Set Error**

7. **MAP_U_ABORT_ind**
   - **MAP_P_ABORT_ind**
   - **MAP_CLOSE_ind**

---

**Figure 19.9.3/1 (sheet 4 of 4)**
Figure 19.9.3/2 (sheet 1 of 2)
Figure 19.9.3/2 (sheet 2 of 2)
19.9.4 Procedure in the HLR

The procedure may be invoked by the USSD application local to the HLR. It may start by using either the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY service.

In both cases the HLR will first check whether the MS is reachable (i.e. there is a VLR identity stored in the subscriber record, the MS record is not marked as purged and the MS record is not marked "MSC Area Restricted").

If the MS is reachable, the HLR will initiate a MAP dialogue with the VLR. Once the dialogue is successfully established the message received from the USSD application will be sent to the MSC.

Following transfer of the message the HLR will wait for a confirmation from the MSC. This will be sent to the USSD application.

Following this, the HLR may receive further uses of the MAP_UNSTRUCTURED_SS_REQUEST or MAP_UNSTRUCTURED_SS_NOTIFY services, or may receive a MAP_CLOSE_ind.

In the event of an error, the MAP process with the VLR shall be released as shown in the diagram.

The procedure in the HLR is shown in figure 19.9.4/1.
Figure 19.9.4/1 (sheet 1 of 2)
Figure 19.9.4/1 (sheet 2 of 2)
19.10 Common macros for clause 19

The following macros are used for the description of more than one of the supplementary service processes described in clause 19:

19.10.1 SS Password handling macros

Macro Get_Password_MSC

This macro is used by the MSC to relay a request for password from the VLR to the MS, and to relay a response from the MS back to the VLR. The macro is described in figure 19.10.1/1.

Macro Get_Password_VLR

This macro is used by the VLR to relay a request for password from the HLR to the MSC, and to relay a response from the MSC back to the HLR. The macro is described in figure 19.10.1/2.
Figure 19.10.1/1
MAP_GET_PASSWORD_end
MAP_GET_PASSWORD_req
MAP_DELIMITER_req
Wait_for_password

MAP_GET_PASSWORD_cnf
From MSC
MAP_GET_PASSWORD_rsp
To MSC

Receive_error_from_MSC
OK

Receive_error_from_HLR
Figure 19.10.2/2
Receive_error_from_MSC
Figure 19.10.2/3
NULL

Figure 19.10.1/2
19.10.2 SS Error handling macros

**Macro Receive_errors_MSC**

This macro is used by the MSC to receive signals which should lead to failure if received in any state of a supplementary service process. If the air interface connection is released by the MS, the communication towards the VLR is aborted, and the MSC should return to a stable "NULL" state. If a MAP_NOTICE indication is received from the VLR, or the VLR aborts or unexpectedly closes the connection, then the air interface connection shall be released. The macro is described in figure 19.10.2/1.

**Macro Receive_error_from_MSC**

This macro is used by the VLR to receive signals from the MSC which should lead to failure if received in any state of a supplementary service process. If a MAP_NOTICE indication is received from the MSC, that connection is closed before the only outcome of the macro, "err" is reported back to the calling process. The macro is described in figure 19.10.2/2.

**Macro Receive_error_from_HLR**

This macro is used by the VLR to receive signals from the HLR which should lead to failure if received in any state of a supplementary service process. If a MAP_NOTICE indication is received from the MSC, that connection is closed. The macro is described in figure 19.10.2/3.
Figure 19.10.2/1
Figure 19.10.2/2
Figure 19.10.2/3
20 Short message service procedures

20.1 General

The short message service procedures are used to control both mobile originated and mobile terminated short message transfer.

Four procedures exist for short message services:
- mobile originated short message service transfer;
- mobile terminated short message service transfer;
- short message alert procedure;
- short message waiting data set procedure.

The following application context refers to a complex MAP user consisting of several processes:
- shortMessageGatewayContext.

This application context needs a co-ordinating process in the HLR. Additionally a Co-ordinator has to be defined for the mobile originated situation in the MSC, because the A_CM_SERV_REQ message does not distinguish between mobile originated short message transfer and the short message alert procedures.

20.1.1 Mobile originated short message service Co-ordinator for the MSC

The A_CM_SERV_REQ message (GSM 04.08) is received from the A-interface containing the CM service type. This parameter indicates mobile originated short message service. The service MAP_PROCESS_ACCESS_REQUEST is started.

If the MAP_PROCESS_ACCESS_REQUEST service ends successfully, the MS initiates mobile originated short message transfer or alerting indication. Depending on the situation, the appropriate process is initiated as follows:
- if the A_RP_MO_DATA indication is received, the process MOSM_MSC is initiated (see subclause 20.2.1);
- if the A_RP_SM_MEMORY_AVAILABLE indication is received, the process SC_Alert_MSC is initiated (see subclause 20.4.1).

After creation of the user process the Co-ordinator relays the messages between the A-interface and the invoked process until a request or an indication for dialogue termination is received.

The SMS process Co-ordinator is shown in the figure 20.1/1.
Figure 21.4/1

Figure 20.4/4

Figure 20.2/2

Figure 20.1/1
20.1.2 Short message Gateway Co-ordinator for the HLR

The MAP_OPEN indication opens a dialogue for the short message procedure between the gateway MSC and the HLR when the application context shortMessageGatewayContext is received. If that service is successful, the Co-ordinator can receive the first service primitive from the MAP_PM. Depending on the received primitive, the user process is created as follows:

- if the MAP_SEND_ROUTING_INFO_FOR_SM indication is received, the process Mobile_Terminated_MS_HLR is created;
- if the MAP_REPORT_SM_DELIVERY_STATUS indication is received, the process Report_SM_delivery_stat_HLR is created.

After creation of the user process the Co-ordinator relays the messages between the MAP_PM and the invoked process until a request or an indication for dialogue termination is received.

The SM Gateway Co-ordinator is shown in the figure 20.1/2.
Refer to the relevant version of GSM 09.02.
20.2 The mobile originated short message transfer procedure

The mobile originated short message service procedure is used to forward short message from a mobile subscriber to a Service Centre. The mobile originated short message service procedure is shown in figure 20.2/1.

1) Short Message (BSSAP; GSM 04.11)
2) MAP_SEND_INFO_FOR_MO_SMS
3) MAP_SEND_INFO_FOR_MO_SMS_ACK
4) MAP_MO_FORWARD_SHORT_MESSAGE
5) Short message (TS GSM 03.40)
6) Short message Acknowledgement (TS GSM 03.40)
7) MAP_MO_FORWARD_SHORT_MESSAGE_ACK
8) Short Message Acknowledgment (BSSAP; GSM 04.11)

Figure 20.2/1: Mobile originated short message transfer

In addition the following MAP services are used:

- MAP_PROCESS_ACCESS_REQUEST (see subclause 6.3);
- MAP_AUTHENTICATE (see subclause 6.5);
- MAP_SET_CIPHERING_MODE (see subclause 6.6);
- MAP_PROVIDE_IMSI (see subclause 6.9);
- MAP_CHECK_IMEI (see subclause 6.7);
- MAP_FORWARD_NEW_TMSI (see subclause 6.9);
- MAP_TRACE_SUBSCRIBER_ACTIVITY (see subclause 7.1);
- MAP_READY_FOR_SM (see subclause 10.4).
20.2.1 Procedure in the servicing MSC

The activation of the MAP_PROCESS_ACCESS_REQUEST service is described in the subclause 21.4.1.

When receiving the short message from the A-interface, the MSC sends the MAP_SEND_INFO_FOR_MO_SMS request to the VLR. As a response the MSC will receive the MAP_SEND_INFO_FOR_MO_SMS confirmation from VLR indicating that:

- the service ends successfully. If the MSC is not itself the IWMSC, the short message transmission towards the IWMSC is initiated using the MAP_MO_FORWARD_SHORT_MESSAGE request;
- the service ends unsuccessfully. The error cause in the MAP_SEND_INFO_FOR_MO_SMS confirmation indicates the reason for the unsuccessful end. The mapping between MAP error causes and RP_ERROR causes is described in TS GSM 03.40.

If there are data errors in the MAP_SEND_INFO_FOR_MO_SMS confirmation, or there is an operation failure in MAP, the RP_ERROR cause network out of order is forwarded to the mobile station.

If the service MAP_MO_FORWARD_SHORT_MESSAGE is started, the MSC will check whether the grouping of MAP_OPEN request and MAP_MO_FORWARD_SHORT_MESSAGE request needs segmentation. If this is the case then the MAP_OPEN request primitive shall be sent first without any associated MAP service request primitive and the dialogue confirmation must be received before the MAP_MO_FORWARD_SHORT_MESSAGE request is sent. As a response to the procedure, the servicing MSC will receive the MAP_MO_FORWARD_SHORT_MESSAGE confirmation from the IWMSC indicating that:

- the short message has been successfully delivered to the Service Centre. The acknowledgement is sent to the mobile station;
- one of several error cases has occurred. The mapping between MAP error causes and RP_ERROR causes is described in TS GSM 03.40. The appropriate indication is provided to the mobile station.

If the procedure failed, a provider error or an abort indication is received. The RP_ERROR cause network out of order is provided to the mobile station.

If the MSC itself is the interworking MSC, the short message is forwarded to the Service Centre. In that case the service MAP_MO_FORWARD_SHORT_MESSAGE is not initiated. The acknowledge message from the Service Centre is forwarded to the mobile station (TS GSM 3.40, TS GSM 4.11).

The mobile originated short message service procedure is shown in figure 20.2/2.
Figure 20.2/2: The mobile originated short message service process in the MSC.
Figure 20.2/2 (sheet 2 of 3)
Figure 20.2/2: The mobile originated short message service process in the MSC.
20.2.2 Procedure in the VLR

The MAP_PROCESS_ACCESS_REQUEST indication starts the MAP_PROCESS_ACCESS_REQUEST service in the VLR. The application context in the MAP_OPEN indication is mobile originated short message transfer.

If the service MAP_PROCESS_ACCESS_REQUEST is successful, the VLR waits for the next message from the MSC. When receiving the MAP_SEND_INFO_FOR_MO_SMS indication, the VLR acts as follows:

- if there is incompatibility in the subscription check, the error teleservice not provisioned is returned to the MSC;
- if the short message transfer would contravene operator determined barring, the call barred error with cause operator barring is returned;
- if the short message transfer would contravene the supplementary service call barring conditions in the VLR, the call barred error with cause barring service active is returned.

When the mobile subscriber has passed all checks, the MAP_SEND_INFO_FOR_MO_SMS response is initiated and the procedure is terminated in the VLR. The mobile originated short message transfer procedure in the VLR is shown in figure 20.2/3.
Refer to the relevant version of GSM 09.02

Figure 20.2/3
20.2.3 Procedure in the interworking MSC

This procedure applies only when the IWMSC is not the servicing MSC.

When receiving a MAP_OPEN indication primitive that is not associated with any MAP service indication primitive and if the dialogue is accepted, the MAP service-user in the interworking MSC issues a MAP_DELIMITER request primitive in order to trigger the local MAP service-provider to confirm the dialogue. Then a MAP_MO_FORWARD_SHORT_MESSAGE indication shall be received.

When a MAP_MO_FORWARD_SHORT_MESSAGE indication is correctly received, the Interworking MSC invokes forwarding of the short message to the Service Centre. If invalid data content is detected, an unexpected data value error or a data missing error is returned to the servicing MSC.

The outcome of the procedure with the Service Centre is awaited before a MAP_MO_FORWARD_SHORT_MESSAGE response is given back to the servicing MSC:

- if a short message is accepted by the Service Centre, an acknowledgement is sent back to the servicing MSC;
- if the Service Centre is not identified, the SM Delivery Failure error is returned to the servicing MSC;
- if the Service Centre returns an error indication, the SM Delivery Failure error is returned to the servicing MSC with the error cause and any diagnostic information received from the Service Centre;
- if the short message cannot be forwarded to the Service Centre or the procedure towards the Service Centre fails for some reason, a system failure error is sent to the servicing MSC.

The mobile originated short message service transfer in the IWMSC is shown in figure 20.2/4.
Figure 20.2/4: The mobile originated short message service process in the IWMSC.
20.3 The mobile terminated short message transfer procedure

The mobile terminated short message transfer procedure is used for forwarding a short message or several short messages from a Service Centre to a mobile subscriber. The mobile terminated short message procedure for a single short message transfer is shown in figure 20.3/1.

1) Short Message (GSM 03.40)
2) MAP SEND ROUTING_INFO FOR_SM
3) MAP SEND ROUTING_INFO FOR_SM_ACK
4) MAP_MT_FORWARD_SHORT_MESSAGE
5) MAP_SEND_INFO_FOR_MT_SMS
6) MAP_PAGE/MAP_SEARCH_FOR_MOBILE_SUBSCRIBER
7) Page (BSSAP; TS GSM 04.08)
8) Page response (BSSAP; TS GSM 04.08)
9) MAP_PROCESS_ACCESS_REQUEST_ACK and MAP_SEARCH_FOR_MOBILE_SUBSCRIBER_ACK
10) MAP_SEND_INFO_FOR_MT_SMS_ACK
11) Short Message (BSSAP; TS GSM 04.11)
12) Short Message Acknowledgement (BSSAP; TS GSM 04.11)
13) MAP_MT_FORWARD_SHORT_MESSAGE_ACK
14) Short Message Acknowledgment (GSM 03.40)

Figure 20.3/1: Mobile terminated short message service procedures
The mobile terminated short message procedure for multiple short message transfer is shown in figure 20.3/2.

In the multiple short message transfer the service MAP_MT_FORWARD_SHORT_MESSAGE can be used several times. However, the short message transfer is always acknowledged to the Service Centre before the next short message is sent.
In addition the following MAP services are used:

- MAP_PROCESS_ACCESS_REQUEST (see subclause 6.3);
- MAP_PAGE (see subclause 6.2);
- MAP_SEARCH_FOR_MS (see subclause 6.2);
- MAP_AUTHENTICATE (see subclause 6.5);
- MAP_SET_CIPHERING_MODE (see subclause 6.6);
- MAP_CHECK_IMEI (see subclause 6.7);
- MAP_FORWARD_NEW_TMSI (see subclause 6.9);
- MAP_REPORT_SM_DELIVERY_STATUS (see subclause 10.3);
- MAP_INFORM_SERVICE_CENTRE (see subclause 10.6);
- MAP_TRACE_SUBSCRIBER_ACTIVITY (see subclause 7.1);
- MAP_READY_FOR_SM (see subclause 10.4).

20.3.1 Procedure in the Servicing MSC

When initiating the dialogue with the servicing MSC, the SMS Gateway MSC must provide the IMSI of the subscriber to whom the short message is directed.

The IMSI can be included either in the Destination Reference of the MAP_OPEN indication received from the SMS Gateway MSC or in the sm-RP-DA information field of the MAP_MT_FORWARD_SHORT_MESSAGE indication.

When receiving a MAP_OPEN indication primitive that is not associated with any MAP service indication primitive and if the dialogue is accepted, the MAP service-user in the servicing MSC issues a MAP_DELIMITER request primitive in order to trigger the local MAP service-provider to confirm the dialogue.

When receiving the first MAP_MT_FORWARD_SHORT_MESSAGE indication from the gateway MSC, the servicing MSC sends the MAP_SEND_INFO_FOR_MT_SMS request primitive to the VLR, if the MAP service primitive is accepted and if short message service is supported in the servicing MSC.

The MAP_MT_FORWARD_SHORT_MESSAGE indication primitive is checked by the macro "Check_Indication". If the received MAP service primitive contains errors, the service is aborted and an unexpected data value error or data missing error is returned to the GMSC.

If the MSC does not support the short message service, the service is aborted in the servicing MSC and the error "Facility Not Supported" is returned to the GMSC.

The subscriber identity information that may be included in the MAP_OPEN indication primitive and in the MAP service indication primitive is checked by the macro "Check_Subscr_Identity_For_MT_SMS" as follows.

If a Destination Reference has been received in the MAP_OPEN indication, an LMSI must be present in the sm-RP-DA information field of the MAP_MT_FORWARD_SHORT_MESSAGE indication. The LMSI shall be included in the sm-RP-DA information field of the MAP_SEND_INFO_FOR_MT_SMS request sent to the VLR; the associated MAP_OPEN request must contain a Destination Reference that carries an IMSI.

Otherwise, if the IMSI is included in the sm-RP-DA information field of the MAP_MT_FORWARD_SHORT_MESSAGE indication, it is mapped into the sm-RP-DA information field of the MAP_SEND_INFO_FOR_MT_SMS request that is sent to the VLR. In this case, the IMSI is not accompanied by an LMSI and neither the MAP_OPEN indication received from the gateway MSC nor the MAP_OPEN request sent to the VLR shall include a Destination Reference.

If a Destination Reference has been received in the servicing MSC and the sm-RP-DA information field of the MAP_MT_FORWARD_SHORT_MESSAGE indication does not include an LMSI or if no Destination Reference has been received and the sm-RP-DA information field does not cover an IMSI the service is aborted in the servicing MSC and the error "Unexpected Data Value" is returned to the SMS GMSC.
The following responses to the MAP_SEND_INFO_FOR_MT_SMS request may be received from the VLR:

- unidentified subscriber or system failure error. The error code is forwarded to the GMSC;
- absent subscriber error. The absent subscriber_SM error is forwarded to the GMSC with the absent subscriber diagnostic indication set to ‘IMSI Detached’;
- unknown subscriber error. The system failure indication is provided to the GMSC;
- data missing or unexpected data value error. The system failure indication is provided to the GMSC;
- a provider error or an abort indication. The system failure indication is provided to the GMSC;
- subscriber busy for MT SMS. The error code is forwarded to the GMSC;
- paging procedure invocation (see subclause 21.3) reporting the successful outcome of the procedure;
- search procedure invocation (see subclause 21.3) reporting the successful outcome of the procedure.

The result of the paging or the search procedure is processed as follows:

- if the procedure is completed successfully, the MSC will send the MAP_PROCESS_ACCESS_REQUEST request to the VLR (see subclause 21.4);
- if the procedure is completed successfully, but the MS has no mobile terminated short message transfer capability, the procedure is terminated and SM delivery failure indication with cause "equipment not SM equipped" is provided to the GMSC;
- if the procedure ends unsuccessfully, the termination of the procedure is awaited from the VLR. The absent subscriber_SM error is forwarded to the GMSC with the absent subscriber diagnostic indication set to 'No Paging Response’, but the other error causes are reported as a system failure indication.

If the short message transfer is aborted for any reason, the dialogue with the VLR is aborted. If the procedure with the VLR is aborted by the VLR or by the provider, a system failure indication is provided to the GMSC.

The unsuccessful outcome of the MAP_PROCESS_ACCESS_REQUEST service is reported by using the system failure error to the GMSC.

When the service MAP_PROCESS_ACCESS_REQUEST is carried out, the MSC will receive the MAP_SEND_INFO_FOR_MT_SMS confirmation indicating:

- the unsuccessful outcome of the procedure. The error indication received from the VLR is forwarded to the GMSC;
- the successful outcome of the procedure. The MSC initiates forwarding of the short message to the MS.

If the primitive itself is badly formatted or data is missing, the system failure error is sent to the GMSC.

If forwarding of the short message is initiated, the MSC awaits the result before one of the following responses is sent back to the GMSC:

- an acknowledge if the short message has been successfully delivered to the mobile subscriber;

- an SM delivery failure error containing a parameter indicating either of the following: there is a MS protocol error or the MS memory capacity is exceeded; detailed diagnostic information (see subclause 5.6.1.4) may also be carried;

- a system failure error if the delivery procedure is aborted.
If the More Messages To Send flag was FALSE or the service MAP_MT_FORWARD_SHORT_MESSAGE ends unsuccessfully, the transaction to the gateway MSC is terminated. Otherwise, the servicing MSC waits for the next short message from the Service Centre.

When receiving the next MAP_MT_FORWARD_SHORT_MESSAGE indication from the gateway MSC the servicing MSC will act as follows:

- if the received primitive contains errors, the unexpected data value error or data missing error is provided to the gateway MSC;

- if the More Messages To Send flag is FALSE, the servicing MSC will start the short message transfer procedure to the mobile subscriber. The successful or unsuccessful outcome of this procedure is reported to the gateway MSC and the transaction is terminated.

- if the More Messages To Send flag is TRUE, the servicing MSC will start the short message transfer to the mobile subscriber. If the outcome of this procedure is unsuccessful, the reason is reported to the gateway MSC and the procedure is terminated. If the procedure is successful, it is acknowledged to the gateway MSC and more short messages can be received.

The tracing procedure may be activated. It is described in detail in the clause 17.

The mobile terminated short message transfer procedure in the servicing MSC is shown in figures 20.3/3 and 20.3/4. The page and search procedures are shown in figure 21.3/1 and 21.3/2.
Figure 20.3/3 (sheet 1 of 3): Procedure MT_SM_Transfer_MSC
Figure 20.3/3 (sheet 2 of 3): Procedure MT_SM_Transfer_MSC
Figure 20.3/3 (sheet 3 of 3): Procedure MT_SM_Transfer_MSC
ETSI TS 100 974 V5.18.0 (2003-03)

Figure 20.3/4 (sheet 1 of 3): Macro MT_SM_MSC
Macrodefinition MT_SM_Transfer_MSC

Figure 20.3/4: The mobile terminated short message transfer macro in the MSC

Figure 20.3/4 (sheet 2 of 3): Macro MT_SM_MSC
Figure 20.3/4 (sheet 3 of 3): Macro MT_SM_MSC
20.3.2 Procedures in the VLR

When receiving the MAP_SEND_INFO_FOR_MT_SMS indication, the VLR will act as follows:

- the parameters and data in the primitive are checked by the macro "Check_Indication". A data failure is reported as an unexpected data value error or a data missing error depending on the nature of the failure;

- for mobile terminated short message the mobile subscriber is identified either by the IMSI only or by the IMSI accompanied by the LMSI. The subscriber identity information that may be included in the MAP_OPEN indication primitive and in the MAP service indication primitive is checked by the macro "Check_Subscr_Identity_For_MT_SMS". In the first case, the IMSI is included in the sm-RP-DA information field and the Destination Reference must not be present in the MAP_OPEN primitive. In the latter case the IMSI must be obtained from the Destination Reference of the MAP_OPEN indication primitive and an LMSI must be present in the sm-RP-DA information field of the MAP_SEND_INFO_FOR_MT_SMS indication. If the mobile subscriber is unknown, the unidentified subscriber error is returned;

- if the "Confirmed by HLR" indicator is set to "Not Confirmed", the unidentified subscriber error is returned;

- if the IMSI Detached Flag is set to detached or the LA Not Allowed Flag is set to not allowed in the VLR, an absent subscriber error with the diagnostic indication set to 'IMSI Detached' is returned and the MS not reachable flag (MNRF) is set;

- if the MAP_SEND_INFO_FOR_MT_SMS indication has passed all the tests, the VLR will initiate the paging procedure. If the location area identification is known and the "Confirmed by Radio Contact" indicator is set to "Confirmed", the MAP_PAGE service is used. Otherwise the MAP_SEARCH_FOR_MOBILE_SUBSCRIBER service is started.

The following responses to the paging procedure may be received from the MSC:

- the MAP_SEARCH_FOR_MOBILE_SUBSCRIBER confirmation indicating a successful outcome, if the search procedure is used. After that the VLR awaits the MAP_PROCESS_ACCESS_REQUEST indication from the MSC;

- the MAP_PAGE confirmation or MAP_SEARCH_FOR_MOBILE_SUBSCRIBER confirmation indicating unsuccessful outcome. If an absent subscriber error is received, the MS not reachable flag (MNRF) is set in the VLR. The errors are forwarded to the MSC in the MAP_SEND_INFO_FOR_MT_SMS response, the absent subscriber error is forwarded with the diagnostic indication set to 'No Paging Response'. If the unexpected data value or unknown location area error is received, the system failure indication is given to the MSC; if subscriber busy for MT SMS is received, this cause is given to the MSC.

- the MAP_PROCESS_ACCESS_REQUEST indication telling that the outcome of the service MAP_PAGE is successful.

If the paging procedure or process access request procedure or any other procedure invoked fails, the appropriate error is reported to the MSC.

If the process access request procedure is successful, the VLR will send the MAP_SEND_INFO_FOR_MT_SMS response to the MSC and the transaction is terminated in the VLR.

The mobile terminated short message transfer procedure in the VLR is shown in figure 20.3/5.
Figure 20.3/5 (sheet 1 of 3): Process MT_SM_VLR
Figure 20.3/5 (sheet 2 of 3): Process MT_SM_VLR
Figure 20.3/5: The mobile terminated short message service process in the VLR

Process MT_SM_VLR

Figure 20.3/5 (sheet 3 of 3): Process MT_SM_VLR
20.3.3 Procedures in the HLR

The MAP_SEND_ROUTING_INFO_FOR_SM indication is received from the GMSC. The following error cases are reported to the GMSC in the MAP_SEND_ROUTING_INFO_FOR_SM response as an unsuccessful outcome of the procedure:

- if the necessary parameters and data are not present in the primitive or they are badly formatted, the data missing or unexpected data value error is returned;
- if the mobile subscriber is unknown, i.e. it cannot be identified from the MSISDN given, an unknown subscriber error is returned;
- if the short message transfer would contravene operator determined barring, the call barred error with cause operator barring is returned;
- if the short message transfer would contravene the supplementary service barring, the call barred error with cause barring service active is returned;
- if the mobile subscription identified by the given MSISDN number does not include the short message service, the teleservice not provisioned error is returned;
- if the location registration of the mobile subscriber shows that the visited PLMN does not support the MT short message service, the facility not supported error is returned;
- if no MSC identity is stored for the mobile subscriber or the "MSC Area Restricted Flag" is set or the "MS purged" flag is set, i.e. the MS is not reachable, the MSISDN-Alert and the SC address are included in the MWD (if possible), the flag MNRF is set and the "Absent Subscriber_SM" error is returned with the appropriate absent subscriber diagnostic indication, i.e. ‘Deregistered in HLR’, ‘Roaming Restricted’ or ‘MS-Purged’.

The priority parameter (SM_RP_PRI) is processed as follows:

- if the priority is low (SM_RP_PRI = False) and the mobile station not reachable flag (MNRF) is set, an absent subscriber_SM error is returned. If a reason for the subscriber’s absence is stored in the mobile not reachable reason (MNRR) in the subscriber data, then this is returned with the absent subscriber_SM error. The SC-address given in the request will be included in the MWD if possible. The service MAP_INFORM_SERVICE_CENTRE including the parameter MW Status is invoked to indicate whether or not the SC address has been included in the MWD list.
- if the priority is low (SM_RP_PRI = False), and the MNRF is clear, the routing information is retrieved as described below;
- if the priority is high (SM_RP_PRI = True) and the MNRF is set, the HLR will send the acknowledge primitive containing the routing information to the gateway MSC. In addition the service MAP_INFORM_SERVICE_CENTRE including the parameter MW Status is invoked to indicate whether or not the SC address is already included in the MWD list.

If the MSISDN-Alert number of the mobile subscriber stored in the MWD is not the same as that received in the MAP_SEND_ROUTING_INFO_FOR_SM indication, the HLR will include in the MAP_INFORM_SERVICE_CENTRE request to the GMSC the MSISDN-Alert number stored.

The MAP_INFORM_SERVICE_CENTRE request is sent also when the MCEF and/or MNRF are set but the routing information is still sent to the GMSC. The status of the flags is indicated in the parameter MW Status.

The routing information is included in a MAP_SEND_ROUTING_INFO_FOR_SM response as follows:

- the IMSI will be returned to the GMSC together with the MSC number and may be optionally accompanied by the LMSI.

The mobile terminated short message transfer procedure in the HLR is shown in figure 20.3/6.
Figure 20.3/6 (sheet 1 of 3): Process MT_SM_HLR
Figure 20.3/6 (sheet 2 of 3): Process MT_SM_HLR
Figure 20.3/6 (sheet 3 of 3): Process MT_SM_HLR
20.3.4 Procedures in the gateway MSC

The short message handling function of the GMSC will request routing information when a mobile terminated short message is received from a Service Centre. The GMSC sends the MAP_SEND_ROUTING_INFO_FOR_SM request to the HLR containing the subscriber data of the mobile subscriber.

As an outcome of the procedure the MAP_SEND_ROUTING_INFO_FOR_SM confirmation is received indicating:

- an unsuccessful event indication containing an error;
  
  The mapping between the MAP error causes and the RP_ERROR causes is explained in TS GSM 03.40.

- a successful event indication containing following parameters:
  
  - an IMSI optionally accompanied by an LMSI; and
  
  - a routing address (a servicing MSC address).

The GMSC may also receive a MAP_INFORM_SERVICE_CENTRE indication after the MAP_SEND_ROUTING_INFO_FOR_SM confirmation. The parameter MW Status in the message indicates whether or not the Service Centre address is stored in the Message Waiting Data. It also indicates the status of the MCEF and MNRF flags in the HLR.

If the MSISDN-Alert stored in the MWD data is not the same as the one sent to the HLR, the MSISDN-Alert is received in the MAP_INFORM_SERVICE_CENTRE indication. This MSISDN number shall be transferred in a delivery failure report to the SC.

In the abnormal end or in the provider error case the system failure error is provided to the SC.

The forward short message procedure is initiated when the GMSC has obtained the routing information needed to forward a mobile terminated short message to the servicing MSC. If an LMSI has been provided in the MAP_SEND_ROUTING_INFO_FOR_SM confirmation, it can be included in the sm-RP-DA information field of the first MAP_MT_FORWARD_SHORT_MESSAGE request sent to the servicing MSC. In this case, the IMSI must be included in the Destination Reference of the MAP_OPEN request. If the LMSI is not sent by the SMS Gateway MSC, the sm-RP-DA information field in the first MAP_MT_FORWARD_SHORT_MESSAGE request sent to the servicing MSC shall contain the IMSI and the Destination Reference in the MAP_OPEN request shall not be present. The Service Centre address is sent in the parameter SM_RP_OA. The More Messages To Send flag is set to TRUE or FALSE depending on the information received from the Service Centre.

If the GMSC is the servicing MSC then the MAP service is not initiated. The procedure in the Servicing MSC is described in subclause 20.3.1 and in the figure 20.3/4.

If the grouping of MAP_OPEN request and MAP_MT_FORWARD_SHORT_MESSAGE request together would need segmenting, these primitives must not be grouped together. The MAP_OPEN request primitive is sent first without any associated MAP service request primitive and the dialogue confirmation must be received before the MAP_MT_FORWARD_SHORT_MESSAGE request is sent.

As a response to the procedure, the GMSC will receive the MAP_MT_FORWARD_SHORT_MESSAGE confirmation indicating:

- a successful forwarding of the short message. This indication is passed to the SC;

- unsuccessful forwarding of the short message. The mapping of the MAP error causes and the RP_ERROR causes is explained in TS GSM 03.40. The appropriate error indication is sent to the SC.

A provider error is indicated as a system failure error to the SC.
The GMSC invokes the procedure MAP_REPORT_SM_DELIVERY_STATUS, if an absent subscriber_SM, an unidentified subscriber or SM delivery failure with error cause MS memory capacity exceeded indication is received from the servicing MSC, and the corresponding flags received in the MAP_INFORM_SC are not already set or the SC address is not yet included in the MWD set. If absent subscriber diagnostic information is included with the absent subscriber_SM error indication then this information is relayed to the HLR using the procedure MAP_REPORT_SM_DELIVERY_STATUS. The gateway MSC may also invoke the procedure when the transfer was successful, if the MNRF and/or MCEF flags were set in the HLR. This procedure is described in detail in subclause 20.5.

Unexpected data value, system failure and unidentified subscriber errors are indicated as a system failure to the SC. Other errors are indicated using appropriate cause values and diagnostic information between the GMSC and the SC as described in TS GSM 03.40 and GSM 04.11.

If there are more short messages to send in the Service Centre and the previous short message transfer succeeded, then the gateway MSC awaits the next short message.

When receiving the next short message from the SC, the gateway MSC sets the More Messages To Send flag according to the information received and starts the service MAP_MT_FORWARD_SHORT_MESSAGE again.

If the gateway MSC is the servicing MSC, then the short message transfer to mobile subscriber is started as described in the subclause 20.3.1.

The mobile terminated short message transfer procedure in the gateway MSC is shown in figure 20.3/7.
Refer to the relevant version of GSM 09.02.

Figure 20.3/7 (sheet 1 of 6)
Figure 20.3/7 (sheet 2 of 6)
Figure 20.3/7: The mobile terminated short message service process in the GMSC.
Figure 20.3/7 (sheet 4 of 6)
Figure 20.3/7: The mobile terminated short message service process in the GMSC

Process MT_SM_GMSC

WAIT FOR MT_SMS CONFIRM, WAIT FOR A_SM CONFIRM

SC_ABORT_Ind

MAP_U_ABORT_Ind, MAP_P_ABORT_Ind, A_ABORT_Ind

MAP NOTICE_Ind

MAP CLOSE_Req

SET RP ERROR SYSTEM FAILURE

GMSC=VMSC?

Yes

A_ABORT_Req

MAP_U_ABORT_Ind

SC RP ERROR Req

NULL

No
Figure 20.3/7: The mobile terminated short message service process in the GMSC

Process MT_SM_GMSC

Figure 20.3/7 (sheet 6 of 6)
Figure 20.3/8
20.4 The Short Message Alert procedure

The Short Message Alert procedure is used for alerting the Service Centre when the mobile subscriber is active after a short message transfer has failed because the mobile subscriber is not reachable or when the MS has indicated that it has memory capacity to accept a short message.

The Short Message Alert procedure for the case when the mobile subscriber was not reachable is shown in figure 20.4/1.

![Diagram](image)

1) CM Service Request, Page response or Location Updating (BSSAP; GSM 04.08)
2) MAP_PROCESS_ACCESS_REQUEST / MAP_UPDATE_LOCATION_AREA
3) MAP_READY_FOR_SM (Mobile Present) / MAP_UPDATE_LOCATION / Supplementary Service Control Request
4) MAP_READY_FOR_SM_ACK
5) MAP_ALERT_SERVICE_CENTRE (notes 1 and 2)
6) Alert Service Centre (GSM 03.40)
7) MAP_ALERT_SERVICE_CENTRE_ACK

NOTE 1: To all Service Centres in the Message Waiting List.
NOTE 2: The HLR initiates the MAP_ALERT_SERVICE_CENTRE service only if the MS Memory Capacity Exceeded flag is clear.

**Figure 20.4/1: Short message alert procedure (Mobile is present)**

The Short Message Alert procedure for the case where the MS indicates that it has memory capacity to accept one or more short messages is shown in figure 20.4/2.
1) SM memory capacity available (BSSAP; GSM 04.11)
2) MAP_READY_FOR_SM (Memory Available)
3) MAP_READY_FOR_SM (Memory Available)
4) MAP_READY_FOR_SM_ACK
5) MAP_READY_FOR_SM_ACK
6) SM memory capacity available (Acknowledge) (BSSAP; GSM 04.11)
7) MAP_ALERT_SERVICE_CENTRE (note)
8) Alert Service Centre (GSM 03.40)
9) MAP_ALERT_SERVICE_CENTRE_ACK

NOTE: To all Service Centres in the Message Waiting List.

Figure 20.4/2: Short message alert procedure (MS memory capacity available)

In addition the following MAP services are used in the MS memory available case:

- MAP_PROCESS_ACCESS_REQUEST (see subclause 6.3);
- MAP_AUTHENTICATE (see subclause 6.5);
- MAP_SET_CIPHERING_MODE (see subclause 6.6);
- MAP_PROVIDE_IMSI (see subclause 6.9);
- MAP_CHECK_IMEI (see subclause 6.7);
- MAP_FORWARD_NEW_TMSI (see subclause 6.9);
- MAP_TRACE_SUBSCRIBER_ACTIVITY (see subclause 7.1).
The Short Message Alert procedure when the MS indicates successful transfer after polling is shown in figure 20.4/3.

Figure 20.4/3: Short message alert procedure (Successful transfer after polling)

20.4.1 Procedures in the Servicing MSC

The activation of the MAP_PROCESS_ACCESS_REQUEST service is described in the subclause 20.6.2.

After receiving the SM memory capacity available indication, the servicing MSC sends the MAP_READY_FOR_SM request to the VLR indicating memory available. The outcome of that procedure is one of the following:

- successful acknowledgment. The MSC sends the corresponding message to the MS;
- negative acknowledgment, where the error causes are treated as follows:
  - unexpected data value, data missing and system failure errors are reported as network out of order error to the MS;
  - facility not supported is reported as requested facility not implemented error to the MS;
  - procedure failure, which is reported as network out of order error to the MS if a connection to the MS still exists.

The short message alert procedure in the MSC for the MS memory capacity available case is shown in figure 20.4/4.
Figure 20.4/4
20.4.2 Procedures in the VLR

20.4.2.1 The Mobile Subscriber is present

When receiving the MAP_PROCESS_ACCESS_REQUEST indication, MAP_UPDATE_LOCATION_AREA indication while the MS not reachable flag (MNRF) is set, the VLR will send the MAP_READY_FOR_SM request towards the HLR. The Alert Reason is set to indicate that the mobile subscriber is present. If the authentication procedure is initiated and it fails, the VLR will not initiate the service. The process in VLR is described in detail in the subclause 21.10.

20.4.2.2 The Mobile Equipment has memory available

The MAP_PROCESS_ACCESS_REQUEST indication starts the MAP_PROCESS_ACCESS_REQUEST service in the VLR. The application context in the MAP_OPEN indication refers to the short message alerting procedure.

If the service MAP_PROCESS_ACCESS_REQUEST is successful, the VLR waits for the next message from the MSC. When receiving the MAP_READY_FOR_SM indication from the MSC, the VLR will check the contents. Data errors are reported to the MSC as an unexpected data value or data missing error, depending on the error. If the primitive passes the data check, the VLR forwards it to the HLR and awaits an acknowledgment.

When receiving the MAP_READY_FOR_SM confirmation from the HLR and the Alert Reason is MS memory available, the VLR will act as follows:

- the MAP_READY_FOR_SM response is sent to the MSC as follows:
  - an acknowledge in the positive case;
  - system failure error, if unexpected data value, data missing, or unknown subscriber errors are received, otherwise the error cause received from the HLR;
  - a facility not supported error, if the HLR supports MAP Vr only;
  - procedure failure is reported as a system failure error.

The short message alert procedure in the VLR is shown in figures 20.4/5.
WF_INVOKE

MAP_READY_FOR_SM_ind

Receive_Open_Cnf

OK

MAP_RESULT

Receive_Open_Cnf

OK

MAP_READY_FOR_SM_cnf

Check_Conditio

OK

Provider Error, User Error, Data Error

'SET USER ERROR'

NULL

Figure 20.4/5
20.4.3  Procedures in the HLR

When receiving the MAP_READY_FOR_SM indication, the HLR will check the contents. Data errors are reported to the VLR as an unexpected data value or a data missing error depending on the error. If the HLR does not support the MNRF, MCEF, and MWD a facility not supported error is reported to the VLR. If the IMSI is unknown an unknown subscriber error is reported to the VLR. Otherwise an acknowledgement is returned to the VLR.

If neither the MS not reachable flag (MNRF) or the memory capacity exceeded flag (MCEF) is set, the HLR sets a timer and waits for it to expire. This ensures that in the race situation the MAP_REPORT_SM_DELIVERY_STATUS service (as described in the subclause 20.6) for the same subscriber can be carried out when delayed in the GMSC.

If the Alert Reason indicates the mobile present situation, or when the update location procedure has been successfully completed or Supplementary Service Control request is received, the MS not reachable flag is cleared and the service centre alert procedure is initiated. If the memory capacity exceeded flag is set, the MS not reachable flag is and stored reason for absence are cleared but the alert procedure is not started.

If the Alert Reason indicates the memory available situation, the HLR initiates the alert procedure. The MS not reachable and memory capacity available flags are cleared.

If the MAP_REPORT_SM_DELIVERY_STATUS indication is received and it indicates the successful transfer of the mobile terminated short message, the HLR initiates the alert procedure described in the subclause 21.10 and clears MCEF and MNRF flags and stored reason for absence.

The short message alert procedure in the HLR is shown in figures 20.4/6 and 21.10/2.
Figure 20.4/6
20.4.4 Procedures in the Interworking MSC

When a MAP_ALERT_SERVICECENTRE indication is correctly received by the IWMSC, the IWMSC will forward the alerting to the given Service Centre if possible.

Data errors are reported to the HLR as an unexpected data value or a data missing error depending on the error.

The short message alert procedure is shown in figure 20.4/7.
Refer to the relevant version of GSM 09.02

Vr

Perform_MAP_Vr_Dialogue

Receive_Open_Ind

OK

Receive_Open_Ind

WF_SERVICE_PRIMITIVE

MAP_ALERT_SERVICE_CENTRE_ind

MAP_CLOSE_req

NULL

Check_Indication

OK

NULL

Error

SC_RP_ALERT_SC_req

MAP_ALERT_SERVICE_CENTRE_rsp

MAP_CLOSE_req

Figure 20.4/7
20.5 The SM delivery status report procedure

The SM delivery status report procedure is used to set the Service Centre address into the message waiting list in the HLR because the subscriber is absent or unidentified or the memory capacity is exceeded. The procedure sets the memory capacity exceeded flag in the HLR if the MS memory does not have room for more messages or the MS not reachable flag in the case of unidentified or absent subscriber.

Additionally the procedure is used to report the HLR about the successful transfer after the Service Centre has polled the subscriber. This procedure is described also in the subclause 20.4.

The SM delivery status report procedure is shown in figure 20.5/1.

![Figure 20.5/1: Short message delivery status report procedure](image)

1) MAP_MT_FORWARD_SHORT_MESSAGE_ACK/_NACK (Absent subscriber_SM, unidentified subscriber or memory capacity exceeded)
2) MAP_REPORT_SM_DELIVERY_STATUS
3) MAP_REPORT_SM_DELIVERY_STATUS_ACK
4) Short Message Negative Acknowledgement (GSM 03.40)

20.5.1 Procedures in the HLR

When the HLR receives a MAPREPORT_SM_DELIVERY_STATUS indication, it acts as described in the subclause 20.6, macro Report_SM_Delivery_Stat_HLR.

The short message delivery status report process in the HLR is shown in figure 20.5/2.
Figure 20.5/2
20.5.2 Procedures in the gateway MSC

The GMSC invokes the short message delivery status report procedure if an absent subscriber_SM indication or unidentified subscriber indication or SM delivery failure error indicating MS memory capacity exceeded is received from the servicing MSC during a mobile terminated short message transfer, and the HLR has not indicated that the SC address is included in the MWD. The unidentified subscriber indication is however processed as the absent subscriber_SM indication.

The service is invoked also when the HLR has indicated that either of the flags MCEF or MNRF is set and the SM delivery was successful or, in case of subsequent SM, the last SM delivery was successful.

The reason for unsuccessful or successful delivery of the short message is included in the SM Delivery Outcome in the MAP_REPORT_SM_DELIVERY_STATUS request. In the case of an unsuccessful delivery due to the subscriber being absent the absent subscriber diagnostic indication (if available) is also included in the MAP_REPORT_SM_DELIVERY_STATUS request.

The GMSC sends the MAP_REPORT_SM_DELIVERY_STATUS request to the HLR. As a response the GMSC will receive the MAP_REPORT_SM_DELIVERY_STATUS confirmation reporting:

- successful outcome of the procedure. The acknowledge primitive may contain the MSISDN-Alert number which is stored in the MWD List in the HLR;
- unsuccessful outcome of the procedure. The system failure indication is forwarded to the SC.

A provider error is indicated as a system failure to the SC.

The procedure towards the Service Centre may also be aborted. If so the operation towards the HLR is also aborted.

The short message delivery status report procedure in the GMSC is shown in figure 20.5/3.
Figure 20.5/3
20.6 Common procedures for the short message clause

20.6.1 The macro Report_SM_Delivery_Stat_HLR

This macro is used when the HLR receives a MAP_REPORT_SM_DELIVERY_STATUS indication from the GMSC. The HLR responses to the indication as follows:

- if invalid data content is detected, an unexpected data value error or a data missing error is returned to the GMSC;

- if the MSISDN number provided is not recognized by the HLR, an unknown subscriber error is returned to the GMSC;

- if the MAP_REPORT_SM_DELIVERY_STATUS indication reports a successful SM delivery, the Service Centres in the Message Waiting list are alerted as described in the subclause 21.10;

- if the SM Delivery Outcome reports unsuccessful delivery and the inclusion of the SC address in the MWD is not possible, a message waiting list full error is returned to the GMSC;

- if the SM Delivery Outcome reports unsuccessful delivery and the message waiting list is not full, the given Service Centre address is inserted and an acknowledgement is sent to the GMSC. If the MSISDN-Alert stored in the subscriber data is not the same as that received in the MAP_REPORT_SM_DELIVERY_STATUS indication, the MSISDN-Alert is sent in a response primitive to the GMSC;

- if the SM Delivery Outcome is MS memory capacity exceeded the HLR sets the memory capacity exceeded flag in the subscriber data and resets the MRNF;

- if the SM Delivery Outcome is absent subscriber the HLR sets the mobile station not reachable flag in the subscriber data. If a reason for absence is provided by the GMSC then this is stored in the mobile station not reachable reason (MNRR) in the subscriber data.

The short message delivery status report macro in the HLR is shown in figure 20.6/1.
Figure 20.6/1: Macro Report_SM_Delivery_Stat_HLR
21 General macro description

21.1 MAP open macros

21.1.1 Macro Receive_Open_Ind

This macro is used by a MAP service-user procedure when a peer entity requests opening of a dialogue.

If the application context received in the MAP-OPEN indication primitive indicates a context name of the MAP version one context set, the macro takes the Vr exit.

If an application-context different from version 1 is received, the presence of MAP_OPEN information is checked. If no MAP_OPEN information has been received, the MAP_OPEN response with:

- Result set to Dialogue Accepted; and
- Application Context Name set to the received value,

is returned

If the received version (Vr) is the one described in this version of MAP, the macro takes the OK exit, otherwise it takes the Vr exit.

If MAP_OPEN information is received, the macro "CHECK_REFERENCE" is called in order to check whether the received values for Destination Reference and Originating Reference correspond with the requirements of the received application-context-name. The outcome of this check is an error, the MAP_OPEN response with:

- Result set to Dialogue Refused;
- Refuse Reason set to Invalid Destination Reference or Invalid Originating Reference;
- Application Context Name set to the highest version supported,

is returned and the macro takes the error exit.

If the data values received for Destination Reference and Originating Reference are accepted for the associated application-context-name it is checked whether the Destination Reference is known if this check is required by the process that calls the macro.

If the Destination Reference (e.g. a subscribers IMSI) is unknown, the MAP_OPEN response with

- Result set to Dialogue Refused;
- Refuse Reason set to Invalid Destination Reference;
- Application Context Name set to the highest version supported,

is returned and the macro takes the error exit.

Else, if the Destination Reference is accepted or if no check is required, the MAP_OPEN response with

- Result set to Dialogue Accepted; and
- Application Context Name set to the received value,

is returned and

If the received version (Vr) is the one described in this version of MAP, the macro takes the OK exit, otherwise it takes the Vr exit.
21.1.2 Macro Receive_Open_Cnf

This macro is used by a user procedure after it requested opening of a dialogue towards a peer entity.

On receipt of a MAP_OPEN Confirmation with a "Result" parameter indicating "Dialogue Accepted", the macro takes the OK exit.

If the "Result" parameter indicates "Dialogue Refused", the "Refuse-reason" parameter is examined. If the "Refuse-reason" parameter indicates "Potential Version Incompatibility", the macro terminates in a way that causes restart of the dialogue by using the version 1 protocol.

If the "Refuse-reason" parameter indicates "Application Context Not Supported" and if the received Application Context Name indicates "Version Vr" (Vr < Vn), the macro terminates in a way that causes restart of the dialogue by using the version Vr protocol. Otherwise, the macro takes the Error exit.

If the "Refuse-reason" parameter indicates neither "Potential Version Incompatibility" nor "Application Context Not Supported", the macro takes the Error exit.

If a MAP_U_ABORT, a MAP_P_ABORT or a MAP_NOTICE Indication is received, the macro takes the Error exit.
Macrodefinition Receive_Open_Ind

Figure 21.1/1: Macro Receive_Open_Ind

![Diagram](image-url)
Figure 21.1/2
Figure 21.1/3
21.2 Macros to check the content of indication and confirmation primitives

21.2.1 Macro Check_Indication

If a parameter required by the application is missing from the indication, the macro takes the error exit, with a user error of "Data Missing".

If a parameter not expected by the application is present in the indication, or an expected parameter has a value not in the set of values permitted by the application, the macro takes the error exit, with a user error of "Unexpected Data Value".

Otherwise the macro takes the "OK" exit.

The macro is shown in figure 21.2/1.

21.2.2 Macro Check_Confirmation

If the confirmation contains a provider error the macro issues a MAP CLOSE request and takes the provider error exit.

Otherwise, if the confirmation contains a user error the macro takes the user error exit.

Otherwise, if a parameter required by the application is missing from the confirmation, or a parameter not expected by the application is present in the confirmation, or an expected parameter has a value not in the set of values permitted by the application, the macro takes the data error exit.

Otherwise the macro takes the "OK" exit.

The macro is shown in figure 21.2/2.
Figure 21.2/2
21.3 The page and search macros

21.3.1 Macro PAGE_MSC

This macro (see figure 21.3/1) is called if a mobile terminating call set-up, an unstructured SS notification, a network-initiated unstructured SS request or a mobile terminating short message is to be delivered to the MS and the current location area identity of the MS is known in the VLR.

When the MSC receives a MAP_PAGE indication, parameter checks are performed first (macro Check_Indication, see subclause 21.2). If parameter errors are detected, the MSC returns a MAP_PAGE response containing the appropriate error cause and the macro terminates with unsuccessful outcome.

Thereafter, several checks on the indication content are performed. The macro terminates by returning the MAP_PAGE response with error:

- Unknown Location Area if the LAI is not known in the MSC;
- System Failure if the call has been released by the calling subscriber or the SMS or SS transaction for this subscriber has been released by the originating entity in the meantime.

Next, the MSC checks if an MM-connection over the radio link already exists for the given IMSI. If so,

- in the case of mobile terminating call set-up the MSC determines whether the busy condition can be established (see GSM 02.01 for a definition of busy states). If the MSC determines that the MS is busy, it returns a MAP_PAGE response with error Busy Subscriber, qualified by either More Calls Allowed or No More Calls Allowed. The macro then terminates with unsuccessful outcome.
- if the service requested is short message service or an unstructured SS notification or network-initiated unstructured SS request, or if the service is mobile terminating call set-up, but the existing connection is for signalling purposes only (i.e. a service different from call set-up), the access connection status is set according to the characteristics of the existing connection (i.e. RR-connection established, ciphering mode on/off, MM-connection existing and authenticated or not), and the macro terminates with successful outcome.

If no MM-connection for the given IMSI exists, paging is initiated at the radio interface within all cells of the location area indicated by the VLR. If the VLR provided the TMSI, the MSC uses it to identify the MS at the radio interface; otherwise the MSC uses the IMSI. The IMSI will also be used to determine the page group (see GSM 04.08). There are several possible outcomes of paging:

- the MS responds to paging, causing the access connection status to be set accordingly (i.e. no RR-connection, in which case other values are not significant), and the macro terminates with successful outcome;
- the MS responds with a channel request containing an establishment cause which is not "answer to paging". The MSC sends a MAP_PAGE response primitive with user error Busy Subscriber before the macro terminates with unsuccessful outcome. This will give priority to the mobile originating request. Alternatively, as an implementation option, the MSC may treat this as a response to paging, which will give priority to the mobile terminating request.
- there is no response from the MS. The MSC sends a MAP_PAGE response primitive with user error Absent Subscriber before the macro terminates with unsuccessful outcome;
- the call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released before a response is received from the MS (indicated in the SDL by the input signal I-REL). The MAP transaction with the VLR will be released in this case by a MAP_U_ABORT request, and the unsuccessful macro termination will indicate transaction termination
- the MAP transaction with the VLR may be released by receiving a MAP_U_ABORT or MAP_P_ABORT indication. The call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released (indicated in the SDL by the output signal I-REL), and the unsuccessful macro termination will indicate transaction termination.
21.3.2 Macro Search_For_MS_MSC

This macro (see figure 21.3/2) is called if a mobile terminating call set-up, an unstructured SS notification, a network-initiated unstructured SS request or a mobile terminating short message is to be delivered to the MS and the current location area identity of the MS is not known in VLR.

When the MSC receives a MAP_SEARCH_FOR_MS Indication, parameter checks are performed first (macro Check_indication, see subclause 21.2). If parameter errors are detected, the MSC returns a MAP_SEARCH_FOR_MS response containing the appropriate error cause and the macro terminates with unsuccessful outcome.

Thereafter, the MSC checks whether the call or the SMS or SS transaction still exists in the MSC. If the call or the SMS or SS transaction has been released, the MSC returns a MAP_SEARCH_FOR_MS response with error System Failure and the macro terminates with unsuccessful outcome.

Next, the MSC checks if an MM-connection over the radio link already exists for the given IMSI. If so,

- in the case of mobile terminating call set-up the MSC determines whether the busy condition can be established (see GSM 02.01 for a definition of busy states). If the MSC determines that the MS is busy, it returns a MAP_SEARCH_FOR_MS response with error Busy Subscriber, qualified by either More Calls Allowed or No More Calls Allowed. The macro then terminates with unsuccessful outcome.

- if the service requested is short message service or an unstructured SS notification or network-initiated unstructured SS request, or if the service is mobile terminating call set-up, but the existing connection is for signalling purposes only (i.e. a service different from call set-up), a MAP_SEARCH_FOR_MS response containing the IMSI and current location area identification of the called MS is returned to the VLR. The access connection status is set according to the characteristics of the existing connection (i.e. RR-connection established, ciphering mode on/off, MM-connection existing and authenticated or not), and the macro terminates with successful outcome.

If no MM-connection for the given IMSI exists, paging is initiated at the radio interface within all cells of all location areas of the VLR, using the IMSI to identify the subscriber and the page group (see GSM 04.08). There are several possible outcomes of paging:

- the MS responds to paging, causing a MAP_SEARCH_FOR_MS response containing the IMSI and current location area identification of the called MS to be returned to the VLR. The access connection status will be set accordingly (i.e. no RR-connection, in which case other values are not significant), and the macro terminates with successful outcome.

- the MS responds with a channel request containing an establishment cause which is not "answer to paging". The MSC sends a MAP_SEARCH_FOR_MS response primitive with user error "Busy Subscriber" before the macro terminates with unsuccessful outcome. This will give priority to the mobile originating request. Alternatively, as an implementation option, the MSC may treat this as a response to paging, which will give priority to the mobile terminating request.

- there is no response from the MS. The MSC sends a MAP_SEARCH_FOR_MS response primitive with user error "Absent Subscriber" before the macro terminates with unsuccessful outcome.
- the call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released before a response is received from the MS (indicated in the SDL by the input signal I-REL). The MAP transaction with the VLR will be released in this case by a MAP_U_ABORT request, and the unsuccessful macro termination will indicate transaction termination.

- the MAP transaction with the VLR may be released by receiving a MAP_U_ABORT or MAP_P_ABORT indication. The call handling connection or MAP transaction on which the call, SMS or unstructured SS transaction is waiting for delivery, is released (indicated in the SDL by the output signal I-REL), and the unsuccessful macro termination will indicate transaction termination.
Figure 21.3/1
Figure 21.3/2
21.4 Macros for handling an Access Request

These macros are invoked when a MS accesses the network, e.g. to set up an outgoing call or when responding to paging. The macro handles identification and authentication of the mobile subscriber as well as invocation of security related features (see GSM 02.09).

21.4.1 Macro Process_Access_Request_MSC

This macro is invoked by any procedure receiving an access request from the MS, e.g. the page response at mobile terminating call set-up or the request for outgoing call set-up.

If no dialogue with the VLR exists (e.g. within the procedure for outgoing call set-up), the MSC will open a dialogue towards the VLR by sending a MAP_OPEN request without any user specific parameters.

In any case, the parameters received from the MS are mapped to a MAP_PROCESS_ACCESS_REQUEST request primitive, containing:

- the received subscriber identification (IMSI, TMSI) or - in case of emergency call set-up - an IMEI;
- the CM service type, indicating the type of request;
- the status of the access connection, i.e. whether a connection to this MS already exists and if so, whether it is already authenticated and ciphered;
- the current location area id of the MS; and
- the CKSN received from the MS.

If opening of the dialogue was required, the MSC will wait for the dialogue confirmation (see macro Receive_Open_Confirmation, subclause 21.1), leading either to:

- immediate unsuccessful exit from the macro, in case no dialogue is possible;
- reversion to MAP version one dialogue if indicated by the VLR. The macro terminates with unsuccessful outcome, as the complete dialogue will be covered by the version one procedure, so that no further action from the calling process is required;
- continuation as given below, if the dialogue is accepted by the VLR.

The MSC waits then for the MAP_PROCESS_ACCESS_REQUEST confirmation. In between, several other indications may be received from the VLR:

- the MSC may receive a MAP_PROVIDE_IMSI indication, handled by the macro Obtain_IMSI_MSC defined in subclause 21.8. In case of positive outcome, the procedure continues waiting for the MAP_PROCESS_ACCESS_REQUEST confirmation, else the macro terminates with unsuccessful outcome;
- the MSC may receive a MAP_AUTHENTICATE indication, handled by the macro Authenticate_MSC defined in subclause 21.5. In case of positive outcome, the procedure continues waiting for the MAP_PROCESS_ACCESS_REQUEST confirmation, else the macro terminates with unsuccessful outcome;
- the MSC may receive a MAP_TRACE_SUBSCRIBER_ACTIVITY indication, handled by the macro Trace_Subscriber_Activity_MSC defined in subclause 21.9;
- the MSC may receive a MAP_SET_CIPHERING_MODE indication, which will be stored for initiating ciphering later on;

- the MSC may receive a MAP_CHECK_IMEI indication, handled by the macro Check_IMEI_MSC defined in subclause 21.6. In case of positive outcome, the procedure continues waiting for the MAP_PROCESS_ACCESS_REQUEST confirmation, else the macro terminates with unsuccessful outcome;

- the MSC may receive a MAP_Obtain_IMEI indication, handled by the macro Obtain_IMEI_MSC defined in subclause 21.6. In case of positive outcome, the procedure continues waiting for the MAP_PROCESS_ACCESS_REQUEST confirmation, else the macro terminates with unsuccessful outcome;

- the MSC may receive a MAP_U_ABORT or MAP_P_ABORT indication, or a premature MAP_CLOSE indication from the VLR. In all these cases, the macro terminates with unsuccessful outcome, after sending the appropriate reject towards the MS (see GSM 09.10);

- the MSC may receive a MAP_NOTICE indication from the VLR. In this case, the dialogue towards the VLR is terminated by a MAP_CLOSE primitive, the appropriate reject is sent towards the MS (see GSM 09.10), and the macro terminates with unsuccessful outcome;

- the MSC may receive an indication for release of the radio path, in which case the dialogue towards the VLR will be terminated by a MAP_U_ABORT primitive, containing the diagnostic information Radio Channel Release.

When the MAP_PROCESS_ACCESS_REQUEST confirmation is received, the parameters of this primitive are checked first. In case of unsuccessful outcome of the service, the MAP User Error received is mapped onto the appropriate radio interface message (see GSM 09.10), before the macro terminates with unsuccessful outcome.

In case of positive outcome of the service, ciphering is initiated on the radio path, if this had been requested by the VLR (see above). Otherwise, if the access request was not triggered by a page response from the MS, the access request is accepted explicitly by sending a CM_Service_Accept message to the MS. If the access request was triggered by a page response from the MS then no CM Service Accept message is sent.

After ciphering has been initiated, the MSC will wait for the MAP_FORWARD_NEW_TMSI indication from the VLR. While waiting, the MSC may receive:

- a MAP_U_ABORT or MAP_P_ABORT indication, or a premature MAP_CLOSE indication from the VLR. In these cases, the macro terminates with unsuccessful outcome, after sending a release request towards the MS (see GSM 09.10);

- a MAP_NOTICE indication from the VLR. In this case, the dialogue towards the VLR is terminated by a MAP_CLOSE primitive, the appropriate reject is sent towards the MS (see GSM 09.10), and the macro terminates with unsuccessful outcome;

- an indication for release of the radio path, in which case the dialogue towards the VLR will be terminated by a MAP_U_ABORT primitive, containing the diagnostic information Radio Channel Release;

- a MAP_DELIMITER request from the VLR. This will be taken as a successful outcome of the macro (i.e. the VLR did not require TMSI reallocation), and it terminates successfully;

- an A_SETUP request from the MS. This will be saved for handling by the procedure which invoked the macro Process_Access_Request_MSC after the macro has terminated.
When the MAP_FORWARD_NEW_TMSI indication is received in the MSC, the TMSI Reallocation Command is sent to the MS, and the MSC waits for an acknowledgement from the MS. In case a positive acknowledgement is received, the MSC sends an empty MAP_FORWARD_NEW_TMSI response primitive to the VLR and terminates successfully. Else, the dialogue is terminated locally (MAP_CLOSE_Req with Release method Prearranged End) without any further action.

If the MSC receives an A_SETUP request while it is waiting for the TMSI acknowledgement from the MS, the A_SETUP is saved for handling by the procedure which invoked the macro Process_Access_Request_MSC after the macro has terminated.

If the dialogue is aborted by the VLR while waiting for the TMSI acknowledgement from the MS, the MSC regards the access request to be failed and terminates with unsuccessful outcome, after sending a release request towards the MS (see GSM 09.10).
Figure 21.4/1 (sheet 3 of 3)
21.4.2 Macro Process_Access_Request_VLR

When the VLR receives a MAP_PROCESS_ACCESS_REQUEST indication, the VLR will check this indication first (macro Check_Indication, see subclause 21.2). In case of negative outcome, the macro will proceed with the error handling described below.

If the indication data are correct, it is checked first whether the subscriber identification (IMSI or TMSI) is known if included:

- if the identification is not known, the IMSI may be requested from the MS, described in the macro Identification_Procedure (see below) with outcome:
  - OK, if a IMSI known in the VLR has been received;
  - Error, if the VLR did not recognize the subscriber's identity. The macro will proceed with the error handling described below;
  - Aborted, if the transaction to the MSC is released. The macro will terminate immediately with unsuccessful.

In case the identity received is an IMEI, the error System Failure is set and the macro proceeds with the error handling described below.

NOTE: Emergency Call with IMEI may be accepted within the error handling phase.

For a known subscriber the authentication check is performed next (see macro Authenticate_VLR, subclause 21.5), if required. If a negative result is received, the VLR proceeds on receipt of user error:

- illegal subscriber depending on the identity used for authentication;
  
  In case IMSI is already used or no new authentication attempt with IMSI shall not be performed (operator option), the error Illegal Subscriber is set and the macro proceeds with the error handling described below.

If a new authentication attempt with IMSI shall be performed, the IMSI is requested from the MS (macro Obtain_IMSI_VLR, see subclause 21.8):

- the authentication will be performed again if a IMSI known in the VLR is received;
- the error Unidentified Subscriber is set and the macro proceeds with the error handling described below, if the IMSI received is unknown in VLR;
- if the IMSI request procedure fails for any other reason, the error System Failure is set and the macro proceeds with the error handling described below;
- if the dialogue has been aborted during the IMSI request, the macro terminates immediately with unsuccessful outcome;
- unknown subscriber by setting the error Unidentified Subscriber and proceeding with the error handling described below.

NOTE: This can occur only in case of data inconsistency between HLR and VLR;

- procedure error by setting the error System Failure and proceeding with the error handling described below;
- null (i.e. the dialogue towards the MSC is terminated) by terminating immediately with unsuccessful outcome.
The MS access is accepted if no authentication is required or after successful authentication. Then, the indicator "Confirmed by Radio Contact" is set to "Confirmed". If the indicator "Location Information Confirmed in HLR" is set to "Not Confirmed", HLR updating will be started as an independent process (Update_Location_VLR, see subclause 16.1.1.6).

If the indicator "Confirmed by HLR" is set to "Not Confirmed", the error Unidentified Subscriber is set and the macro proceeds with the error handling described below.

If roaming is not allowed in the location area indicated in the Current Location Area Id parameter, the error Roaming Not Allowed qualified by the roaming restriction reason is set and the macro proceeds with the error handling described below.

In case roaming is allowed, the IMSI is set to attached and the process for notifying the HLR that the subscriber is present is started if required (Subscriber Present VLR, see subclause 21.10).

At next, tracing is invoked if required by the operator (macro Trace_Subscriber_Activity_VLR, see subclause 21.9). Thereafter,

if ciphering is not required, IMEI checking is invoked if required by the operator (see macro Check_IMEI_VLR defined in subclause 21.6).

The error Illegal Equipment is set in case of unsuccessful outcome of the IMEI check, the subscriber is marked as detached and the macro proceeds with the error handling described below.

The macro terminates immediately with unsuccessful outcome if the MSC dialogue has been released during the IMEI check.

Else, the macro terminates successfully by returning the MAP_PROCESS_ACCESS_REQUEST response containing the IMSI to indicate acceptance of the MS access.

if ciphering is required, the MAP_SET_CIPHERING_MODE request containing:
- the cipher mode indicating the cipher algorithm required; and
- the cipher key to be used;

is sent to the MSC.

As a further operator option, IMEI checking may be performed next.

The error Illegal Equipment is set in case of unsuccessful outcome of the IMEI check, the subscriber is marked as detached and the macro proceeds with the error handling described below.

The macro terminates immediately with unsuccessful outcome if the MSC dialogue has been released during the IMEI check.

Else, the macro terminates successfully by returning the MAP_PROCESS_ACCESS_REQUEST response containing the IMSI to indicate acceptance of the MS access.

IF no TMSI reallocation is required (again an operator option), the macro terminates thereafter. Else, TMSI reallocation is performed by sending a MAP_FORWARD_NEW_TMSI request, containing the new TMSI as parameter. The old TMSI will be frozen until an acknowledgement from the MS has been received. Before the macro terminates, the VLR will wait for the MAP_FORWARD_NEW_TMSI response, containing no parameters if reallocation has been confirmed by the MS, or a Provider Error, otherwise, in which case the old TMSI is kept frozen to avoid double allocation. In this case, both the old as the new TMSI are subsequently regarded valid when used by the MS.
Error handling

In case some error is detected during handling the access request, a respective error has been set. Before returning this error cause to the MSC in a MAP_PROCESS_ACCESS_REQUEST response, it need to be checked whether this access is for emergency call set-up, as this will require extra treatment.

If the CM Service type given in the MAP_PROCESS_ACCESS_REQUEST indication is emergency call set-up, it is checked whether EC set-up in the particular error situation is permitted (operator option). If so, it is checked whether the IMEI is required, and if so the IMEI is requested from the MS (macro Obtain_IMEI_VLR, see subclause 21.6).

The macro will terminate immediately with unsuccessful outcome if the MSC transaction has been aborted during the IMEI retrieval.

In case of an error reported back from IMEI retrieval, MAP_PROCESS_ACCESS_REQUEST response containing the error cause set previously is returned to the MSC, the dialogue is closed (MAP_CLOSE request indicating normal release) and the macro terminates with unsuccessful outcome.

When a subscriber identity required by the operator (IMSI or IMEI) is available, the user error set previously is deleted, the respective identity is returned in the MAP_PROCESS_ACCESS_REQUEST response to indicate acceptance of emergency call, and the macro terminates with successful outcome.

In all other cases, the MAP_PROCESS_ACCESS_REQUEST response containing the error cause set previously is returned to the MSC, the dialogue is closed (MAP_CLOSE request indicating normal release) and the macro terminates with unsuccessful outcome.

21.4.3 Macro Identification Procedure

This macro is invoked by the macro Process_Access_Request_VLR in case the subscribers identity is not known in the VLR.

If the identity received from the MS is an IMSI, the error Unidentified Subscriber will be set and reported back to the calling macro (to be sent in the MAP_PROCESS_ACCESS_REQUEST response). The same error is used in case a TMSI was received from the MS, but the operator does not allow open identification of the MS.

If open identification of the MS is allowed, the macro Obtain_IMSI_VLR is invoked, requesting the subscribers IMSI from the MS (see subclause 21.8), with outcome

OK, in which case it is checked whether for the IMSI received there exists a subscriber record in the VLR. If so, the macro terminates successfully, else the error Unidentified Subscriber will be set and reported back to the calling macro.

Error, in which case the error System Failure will be set and reported back to the calling macro.

Aborted, i.e. the MSC transaction is released, in which the macro terminates accordingly.
Figure 21.4/2: Macro for processing on access request in VLR

1. ID_known
   - yes
     - OK
     - Error
   - no
     - ID_Proc
     - Error
     - Aborted

2. Auth_required
   - no
     - ID_Proc
     - Error
     - Aborted
   - yes
     - Authenticate_VLR

3. MAP_PROCESS_ACCESS_REQUEST_IND
   - yes
     - Null
   - no
     - Procedure_Error
       - Unknown_Subscriber
       - Illegal_Subscriber
       - Procedure_Error

4. Check_Indication
   - ID_known
     - yes
       - OK
       - Error
     - no
       - ID_Proc
       - Error
       - Aborted

5. IMSI_used
   - yes
     - Ok
   - no
     - 'Set_error_Illegal_Subscriber'

6. IMSI_known
   - yes
     - 'Set_error_Unknown_Subscriber'
   - no
     - 'Set_error_System_Failure'

Figure 21.4/2 (sheet 1 of 3)
Figure 21.4/2 (sheet 2 of 3)
Figure 21.4/2 (sheet 3 of 3)
Figure 21.4/3
21.5 Authentication macros and processes

The following macros are used in the GSM network in order to enable authentication of a mobile subscriber.

21.5.1 Macro Authenticate_MSC

This macro is used by the MSC to relay a request for authentication transparently from the VLR to the MS, wait for a response from the MS and to relay the response from the MS back to the VLR. If, while the MSC is waiting for the authentication response, the air interface connection is released or a MAP_U_ABORT, MAP_P_ABORT or MAP_CLOSE indication is received from the VLR, then necessary connections are released and the "Error" exit is used. The macro is described in figure 21.5/1.

21.5.2 Macro Authenticate_VLR

This macro is used by the VLR to control the authentication of a subscriber. The macro proceeds as follows:

- if there are not enough authentication triplets in the VLR to perform the authentication, then the macro "Obtain_Authent_Para_VLR" described below is invoked. If this macro fails, then the corresponding error (Unknown Subscriber or Procedure Error) is returned to the calling process;

- if there are enough authentication triplets in the VLR, or the Obtain_Authent_Para_VLR macro was successful, then a MAP_AUTHENTICATE request is sent to the MSC. This request contains the RAND and CKSN parameters as indicated in the service description;

- the VLR then waits for a response from the MSC;

- if a MAP_U_ABORT, MAP_P_ABORT or MAP_CLOSE indication is received from the MSC in this wait state, the VLR checks whether authentication sets are available. If no sets are available the process Obtain_Authent_Sets_VLR is invoked to fetch authentication sets from the HLR. The "Null" exit is then used;

- if a MAP_NOTICE indication is received from the MSC in this wait state, the VLR closes the dialogue with the MSC, then checks whether authentication sets are available. If no sets are available the process Obtain_Authent_Sets_VLR is invoked to fetch authentication sets from the HLR. The "Null" exit is then used;

- if a MAP_AUTHENTICATE confirmation is received by the VLR, it checks whether the received Signed Result (SRES) is identical to the stored one (see GSM 03.20). If this is not the case, the "Illegal Subscriber" exit is used. If the SRES values are identical, then the "OK" exit is used;

- before exit, the VLR may fetch a new set of triplets from the HLR. This is done by initiating a separate Obtain_Authent_Sets_VLR process described below.

The macro is described in figure 21.5/2.

21.5.3 Process Obtain_Authentication_Sets_VLR

This process is initiated by the VLR to fetch triplets from a subscriber's HLR in a stand-alone, independent manner. The Obtain_Authent_Para_VLR macro described below is simply called; the process is described in figure 21.5/3.
21.5.4 Macro Obtain_Authent_Para_VLR

This macro is used by the VLR to request authentication triplets from the HLR. The macro proceeds as follows:

- a connection is opened, and a MAP_SEND_AUTHENTICATION_INFO request sent to the HLR;
- if the HLR indicates that a MAP version 1 dialogue is to be used, the VLR performs the equivalent MAP version 1 dialogue, which can return a positive result containing authentication sets, an empty positive result, or an error;
- if the dialogue opening fails, the "Procedure Error" exit is used. Otherwise, the VLR waits for the response from the HLR;
- if a MAP_SEND_AUTHENTICATION_INFO confirmation is received from the HLR, the VLR checks the received data.

One of the following positive responses may be received from a MAP version 1 or MAP version 2 dialogue with the HLR:

- Authentication triplets, in which case the outcome is successful;
- Empty response, in which case the VLR may re-use old triplets, if allowed by the PLMN operator.

If the VLR cannot re-use old triplets (or no such triplets are available) then the "Procedure Error" exit is used.

If the outcome was successful or re-use of old parameters in the VLR is allowed, then the "OK" exit is used.

If an "Unknown Subscriber" error is included in the MAP_SEND_AUTHENTICATION_INFO confirm or is returned by the MAP version 1 dialogue, then the "Unknown Subscriber" exit is used.

- if a MAP-U-ABORT, MAP_P_ABORT, MAP_NOTICE or unexpected MAP_CLOSE service indication is received from the MSC, then open connections are terminated, and the macro takes the "Null" exit;
- if a MAP-U-ABORT, MAP_P_ABORT or unexpected MAP_CLOSE service indication is received from the HLR, then the VLR checks whether old authentication parameters can be re-used. If old parameters cannot be re-used the macro takes the "Procedure Error" exit; otherwise it takes the "OK" exit;
- if a MAP_NOTICE service indication is received from the HLR, then the dialogue with the HLR is closed. The VLR then checks whether old authentication parameters can be re-used. If old parameters cannot be re-used the macro takes the "Procedure Error" exit; otherwise it takes the "OK" exit.

The macro is described in figure 21.5/4.
21.5.5 Process Obtain_Auth_Sets_HLR

Opening of the dialogue is described in the macro Receive_Open_Ind in subclause 21.1, with outcomes:

- reversion to version one procedure;
- procedure termination; or
- dialogue acceptance, with proceeding as below.

This process is used by the HLR to obtain authentication triplets from the AuC, upon request from the VLR. The process acts as follows:

- a MAP_SEND_AUTHENTICATION_INFO indication is received by the HLR;
- the HLR checks the service indication for errors. If any, they are reported to the VLR in the MAP_SEND_AUTHENTICATION_INFO response. If no errors are detected, authentication triplets are fetched from the AuC. Further details are found in GSM 03.20;
- if errors are detected they are reported to the VLR in the MAP_SEND_AUTHENTICATION_INFO response. Otherwise the authentication triplets are returned.

The process is described in figure 21.5/5.
Figure 21.5/2
Figure 21.5/3
Macrodefinition Obtain_Authent_Para_VLR

Figure 21.5/4: Macro to obtain authentication parameters from the HLR to the VLR

Section 21.1

MAP_OPEN_Req
MAP_SEND_AUTHENTICATION_INFO_Req
MAP_DELIMITER_Req

Receive_Open_Cnf

OK

V1

Error

Perform MAP V1 Dialogue

Positive result

no

yes

Empty result

no

yes

Procedure_Error

Figure 21.2/2

MAP_SEND_AUTHENTICATION_INFO_Cnf

Figure 21.5/4 (sheet 1 of 2)
Macrodefinition Obtain_Authent_Para_VLR

Figure 21.5/4: Macro to obtain authentication parameters from the HLR to the VLR

215_42(2)

WAIT FOR AUTHENTICATION INFO

MAP P_ABORT_ind, MAP U_ABORT_ind, MAP CLOSE_ind from MSC

MAP U_ABORT_req to HLR

Null

MAP NOTICE_ind from MSC

MAP CLOSE_req to MSC

MAP U_ABORT_req to HLR

Null

MAP P_ABORT_ind, MAP U_ABORT_ind, MAP CLOSE_ind from HLR

MAP NOTICE_ind from HLR

Re-use old parameters

no

yes

OK

Procedure_Error

Figure 21.5/4 (sheet 2 of 2)
Perform MAP V1 Dialogue

MAP_CLOSE_Req

Null

Wait_For Service_Primitive

MAP_NOTICE_Ind

Check_Indication

Null

Null

Null

Null

Null

Figure 21.2/1

MAP_SEND_AUTHENTICATION_INFO

MAP_CLOSE_Req

Null

Null

Null

Null

Figure 21.1/1

MAP_SEND_AUTHENTICATION_INFO

MAP_CLOSE_Req

Figure 21.5/5
21.6 IMEI Handling Macros

The following macros are used in the GSM network in order to enable handling and checking of the mobile equipment identity.

21.6.1 Macro Check_IMEI_MSC

This macro is used by the MSC to receive a request from the VLR, relay it to the EIR, and pass the result from the EIR back to the VLR. The macro proceeds as follows:

- a MAP_CHECK_IMEI service indication containing only the Invoke Id is received from the VLR;
- if the IMEI is not available in the MSC, it is requested from the MS using the IDENTITY REQUEST message;
- if the MS releases the radio resources, a MAP_U_ABORT request indicating "Application procedure Cancellation" is sent to the VLR, and the "Error" exit of the macro is used;
- when the IMEI is known, a connection is set up towards the EIR, and a MAP_CHECK_IMEI service request is sent including the IMEI;
- if the opening of the dialogue fails, a System Failure is reported to the VLR. Otherwise, the MSC waits for a response from the EIR;
- when the MAP_CHECK_IMEI service confirm is received, it is checked for errors. Any errors discovered in the MSC lead to the System Failure error to be reported to the VLR in the MAP_CHECK_IMEI response. Any errors reported from the EIR are sent directly to the VLR in the MAP_CHECK_IMEI service response. If no errors are detected by or reported to the MSC, the IMEI is added to the MAP_CHECK_IMEI service response returned to the VLR. The "OK" exit is used in all cases;
- if a MAP_P_ABORT, MAP_U_ABORT, MAP_CLOSE or MAP_NOTICE service indication is received from the EIR, the MSC closes the transaction with the EIR (if necessary), reports a System Failure error back to the VLR in the MAP_CHECK_IMEI response, and uses the macro's "OK" exit;
- if a MAP_P_ABORT, MAP_U_ABORT, MAP_CLOSE or MAP_NOTICE indication is received from the VLR, the MSC closes the transaction with the VLR (if necessary) and aborts the connections towards the EIR and the MS; the macro takes the "Error" exit.

If the dialogue with the EIR drops back to version 1, the result or error returned by the EIR is checked. The use of the "Check_Confirmation" macro in the SDL diagram indicates that the checks carried out on the result returned by the EIR in a MAP v1 dialogue are functionally equivalent to those carried out on the parameters of the MAP_CHECK_IMEI confirm received from the EIR in a MAP v2 dialogue.

The macro is described in figure 21.6/1.

21.6.2 Macro Check_IMEI_VLR

This macro is used by the VLR to control the check of a mobile equipment’s IMEI. The macro proceeds as follows:

- a MAP_CHECK_IMEI service request is sent to the MSC, including only the Invoke Id;
- the VLR then waits for the response from the MSC;
- if a MAP_CHECK_IMEI service confirm including either:
  - the IMEI and the Equipment Status; or
  - an error;

  is received, the VLR checks whether the response requires that an alarm be generated on the Operation and Maintenance interface. The criteria for such alarms are PLMN operator dependent;

- the VLR then checks whether the response from the MSC means that service is granted to the MS. The criteria for granting service depending on the equipment status or errors received in the MAP_CHECK_IMEI service response are also PLMN operator dependent;

- if a MAP_P_ABORT, MAP_U_ABORT, MAP_CLOSE or MAP_NOTICE indication is received from the MSC, then the MSC connection is closed (if necessary) and the macro takes the "Aborted" exit.

The macro is described in figure 21.6/2.

### 21.6.3 Process Check_IMEI_EIR

This process is used by the EIR to obtain the status of a piece of mobile equipment, upon request from the MSC. The process acts as follows:

- a MAP_OPEN service indication is received (macro Receive_Open_Ind, subclause 21.1.1). If the dialogue opening fails, the process terminates;

- otherwise, a MAP_CHECK_IMEI indication is received by the EIR, containing the IMEI to be checked;

- the EIR checks the service indication for errors. If there are any, they are reported to the MSC in the MAP-CHECK_IMEI response. If no errors are detected, the EIR data base function is interrogated for the status of the given equipment. Further details are found in GSM 02.16;

- the status of the equipment (white-listed, grey-listed, black-listed or unknown) is returned to the MSC in the MAP_CHECK_IMEI service response;

- if a MAP_U_ABORT, MAP_P_ABORT, MAP_NOTICE or MAP_CLOSE indication is received from the MSC at any time during this process, the process in the EIR terminates.

The process is described in figure 21.6/3.

### 21.6.4 Macro Obtain_IMEI_MSC

This macro is used by the MSC to respond to a request from the VLR to provide the IMEI. The macro proceeds as follows:

- a MAP_OBTAIN_IMEI service indication containing only the Invoke Id is received from the VLR;

- if the IMEI is not available in the MSC, it is requested from the MS using the IDENTITY REQUEST message;

- when the IMEI is known, it is returned to the VLR in the MAP_OBTAIN_IMEI service response. The macro terminates at the "OK" exit;

- if the IMEI cannot be obtained by the MSC, the System Failure error is reported back to the VLR in the MAP_OBTAIN_IMEI service response. The macro terminates at the "OK" exit;

- if a MAP_P_ABORT, MAP_U_ABORT or MAP_CLOSE indication is received from the VLR, the macro terminates at the "Error" exit.

The macro is described in figure 21.6/4.
21.6.5 Macro Obtain_IMEI_VLR

This macro is used by the VLR to obtain the IMEI from the MSC, e.g. to enable handling of emergency calls in case of authentication failure (in which case the IMEI may be used by some operators as an alternative to the IMSI). It proceeds as follows:

- the MAP_OBTAIN_IMEI service request is sent to the MSC, including only the Invoke Id;
- the VLR then waits for the response from the MSC;
- if the IMEI is received in the MAP_OBTAIN_IMEI service response, the macro terminates at the "OK" exit;
- if the System Failure error is reported in the MAP_OBTAIN_IMEI service response, the "Error" exit is used;
- if the MSC terminates the dialogue using a MAP_P_ABORT, MAP_U_ABORT, MAP_CLOSE or MAP_NOTICE service indication, the necessary connections are released, and the "Aborted" exit is used for termination of the macro.

The macro is shown in figure 21.6/5.
Figure 21.6/1 (sheet 1 of 2)
Figure 21.6/1 (sheet 2 of 2)
Figure 21.6/2
Figure 21.6/3
Figure 21.6/4
MAP_OBTAIN_IMEI_req

MAP_DELIMITER_req

Wait for IMEI

MAP_OBTAIN_IMEI_con

MAP_NOTICE_ind

MAP_CLOSE_req

MAP_U_ABORT_ind,
MAP_P_ABORT_ind,
MAP_CLOSE_ind

From MSC

Error ?

yes

OK

Eor

Error

Aborted

MAP_U_ABORT_ind,
MAP_P_ABORT_ind,
MAP_CLOSE_ind

From MSC

MAP_CLOSE_req

MAP_NOTICE_ind

MAP_OBTAIN_IMEI_con

MAP_DELIMITER_req

MAP_OBTAIN_IMEI_req

To MSC

To MSC

To MSC

To MSC

To MSC

To MSC

Figure 21.6/5
21.7 Insert Subscriber Data Macros

21.7.1 Macro Insert_Subs_Data_VLR

This macro describes the reception of the InsertSubscriberData service indication. This macro is used by any procedure that triggers the reception of subscriber data (e.g. Update Location or Restore Data).

If the VLR does not support any basic or supplementary service or the network feature Operator Determined Barring, or there is a problem with Regional Subscription Data then it reports it to the HLR.

If the entire MSC area is restricted due to regional subscription this is reported to the HLR.

The SDL diagram is shown in figure 21.7/1.
21.7.2 Process Insert_Subs_Data_Stand_Alone_HLR

This process is used by HLR to transfer subscriber data to VLR in a stand alone mode, i.e. in its own dialogue. This is done whenever a change of subscriber data is performed either by the operator or by the subscriber and this change has to be reported to VLR.

The process, after opening the dialogue with VLR, sends as many requests of the InsertSubscriberData service as necessary to transfer the subscriber data. The call to the process "Send_Insert_Subs_Data" (see subclause 21.7.4) is meant to describe two possible behaviours of the HLR when more than one service request has to be sent:

- either the HLR handles the requests and the confirmations in parallel; or
- the HLR sends every request after receiving the confirmation to the previous one.

The macro "Wait_for_Insert_Subs_Data_Cnf" (see subclause 21.7.3) is also called in order to handle every single confirmation.

If the result of a primitive received from the VLR is unsuccessful, the HLR may initiate re-attempts; the number of repeat attempts and the time in between are HLR operator options, depending on the error returned by the VLR.

If certain services required for a subscriber are not supported by the VLR (e.g. Advice of Charge Charging Level), this may result in one of the following outcomes:

- the HLR stores and sends "Roaming Restriction Due To Unsupported Feature" in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. If "Roaming Restriction Due To Unsupported Feature" is stored in the HLR, the "MSC Area Restricted Flag" shall be set to "restricted". This will prevent MT calls, MT SM and MT USSD from being forwarded to the MSC/VLR.

- the HLR stores and sends other induced subscriber data (e.g. a specific barring program) in a subsequent MAP_INSERT_SUBSCRIBER_DATA service. This will cause rejection of mobile originated service requests, except emergency calls.

When the VLR receives regional subscription data (Zone Code List) it may respond with "MSC Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response. In this case the "MSC Area Restricted Flag" shall be set to "restricted" in the HLR. This will prevent MT calls, MT SM and MT USSD from being forwarded to the MSC/VLR.

If the HLR does not store "Roaming Restriction Due To Unsupported Feature" as a consequence of the stand alone Insert Subscriber Data procedure and the HLR does not receive "MSC Area Restricted" in the MAP_INSERT_SUBSCRIBER_DATA response and "Roaming Restriction Due To Unsupported Feature" has not been stored in the HLR in the course of a previous subscriber data retrieval procedure, the "MSC Area Restricted Flag" in the HLR shall be set to "not restricted".

The SDL diagram is shown in figure 21.7/2.
Process Insert_Subs_Data_Stand_Alon_HLR

Figure 21.7/2 (sheet 1 of 2)
Process Insert_Subs_Data_Stand_A lone_HLR

Figure 21.7/2: Independent process to open a new dialog and transfer subscriber data from HLR to VLR.

Error

Replace_Service

OK

Abort

1. Count:= Count-1

2. Wait_for_Insert_Subs_Data_Cnf

3. Set_replacement_service(s)

4. MAP_CLOSE_req

5. Repeat update of VLR

Figure 21.7/3: Roaming_restriction: due_to_unsupported_feature_or_MSC_area_restricted_received

Error

Replace_Service

OK

Abort

1. Count:= Count-1

2. Wait_for_repetition_timer_expiry

3. MAP_U_ABORT_req

4. MAP_U_ABORT_req

5. MAP_U_ABORT_req

Figure 21.7/2 (sheet 2 of 2)
21.7.3 Macro Wait_for_Insert_Subsc_Data_Cnf

This macro is used by any process or macro that describes the handling of the reception of the Insert_Subscriber_Data service in HLR (e.g. Update Location or Restore Data).

If the VLR reports the non-support of some basic or supplementary service or the network feature Operator Determined Barring then three actions are possible:

- to ignore the information received;
- to replace the not supported service;
- or to perform any other internal action.

The SDL diagram is shown in figure 21.7/3.
Figure 21.2/2

Figure 21.7/3
21.7.4 Process Send_Insert_Subscriber_Data

This process is used by any process or macro where the Insert_Subscriber_Data request is sent to VLR.

The SDL diagram is shown in figure 21.7/4.

![Figure 21.7/4]
21.8 Request IMSI Macros

21.8.1 Macro Obtain_IMSI_MSC

This macro describes the handling of the request received from the VLR to provide the IMSI of a subscriber (e.g. at Location Updating).

The SDL diagram is shown in figure 21.8/1.

![SDL Diagram](image-url)
21.8.2 Macro Obtain_IMSI_VLR

This macro describes the way VLR requests the MSC the IMSI of a subscriber (e.g. at Location Updating).

The SDL diagram is shown in figure 21.8/2.
21.9 Tracing macros

21.9.1 Macro Trace_Subscriber_Activity_MSC

The Trace_Subscriber_Activity_MSC is invoked in the MSC, when the MSC receives the MAP_TRACE_SUBSCRIBER_ACTIVITY indication from the VLR. The data of the primitive is checked and the tracing in the MSC is started if the content includes no errors. No response is returned to the VLR.

The Trace_Subscriber_Activity_MSC macro is described in the figure 21.9/1.
21.9.2 Macro Trace_Subscriber_Activity_VLR

The macro Trace_Subscriber_Activity_VLR is invoked, if the subscriber activity is detected by the VLR and the tracing is active. The VLR sends MAP_TRACE_SUBSCRIBER_ACTIVITY request to the MSC. No answer is awaited from the MSC.

The Trace_Subscriber_Activity_VLR macro is shown in the figure 21.9/2.

Figure 21.9/2
21.9.3 Macro Activate_Tracing_VLR

The Activate_Tracing_VLR macro is invoked, when the MAP_ACTIVATE_TRACE_MODE indication is received from the HLR. The primitive is processed in the VLR as follows:

- if the data contains errors, a data missing or unexpected data value indication is returned to the HLR;
- if the tracing is not supported, a facility not supported indication is returned to the HLR;
- if the tracing buffer does not have any space left for the data, a tracing buffer full indication is returned to the HLR;
- if no errors is detected, the tracing is set active and a positive acknowledge is returned to the HLR.

The Activate_Tracing_VLR macro is described in the figure 21.9/3.

![Figure 21.9/3](image-url)
21.9.4 Macro Control_Tracing_HLR

The Control_Tracing_HLR macro may be invoked in the HLR, if subscriber related activity is detected. If the tracing is active in the HLR and not active in the VLR, the MAP_ACTIVATE_TRACE_MODE request is sent to the VLR.

The MAP_ACTIVATE_TRACE_MODE confirmation from the VLR is processed as follows:

- if the primitive contains a successful acknowledge, the tracing in VLR is set active;
- if the primitive contains errors, the tracing in VLR is set deactive.

The Control_Tracing_HLR macro is shown in the figure 21.9/4.
Figure 21.2/2

Figure 21.9/4
21.10 Short Message Alert procedures

21.10.1 Subscriber_Present_VLR process

The Subscriber_Present_VLR process is invoked by the VLR, when the mobile subscriber becomes active and the MNRF flag is set. The general description of the short message alert procedures is in the subclause 20.4.

The VLR sends the MAP_READY_FOR_SM request to the HLR and waits for the HLR to answer. When receiving the answer, the VLR will act as follows:

- the MNRF flag is cleared if the procedure is successful;
- the MNRF flag is not cleared if the procedure is not successful.

The Subscriber_Present_VLR process is shown in the figure 21.10/1.

![Diagram](image-url)
21.10.2 Macro Alert_Service_Centre_HLR

The Alert_Service_Centre_HLR macro is initiated when the HLR notices that the Service Centre(s) shall be alerted. The macro starts process Alert_Service_Centre_HLR for every SC address in the MWD list.

In the process Alert_Service_Centre_HLR the HLR sends MAP_ALERT_SERVICE_CENTRE request to the appropriate IWMSC. The MWD entry is deleted when the positive acknowledge is received from the IWMSC. The unsuccessful alert may be repeated. The MWD entry should be purged in the unsuccessful case, at least when a suitable time period has expired.

The Alert_Service_Centre_HLR macro is shown in the figure 21.10/2 and the Alert_Service_Centre_HLR process is shown in the figure 21.10/3.

![Diagram of Alert_Service_Centre_HLR process]
Annex A (informative): Cross-reference for abstract syntaxes of MAP

Annex A is not part of the standard, it is included for information purposes only.

For every ASN.1 item such as identifier, type-reference or value-reference the cross-reference allows to locate all occurrences by means of module-name and line numbers. For that purpose line numbers are printed at the left margin in front of each ASN.1 source line starting with 1 for every module.

The items are sorted alphabetically in the cross-reference in a case-insensitive manner. Occurrences of an item are its definition and all its usages such as in exports, imports or within a type or value assignment.

For every item additional information is provided such as kind of item (identifier, value reference, type reference), and tag, associated type and value if applicable.

The cross-reference for a root module includes all modules referred to directly or indirectly via imports. The cross-references for the root modules MAP-Protocol/TCAPMessages and MAP-DialoguePDU are included.

<table>
<thead>
<tr>
<th>TAG</th>
<th>P3.90M</th>
<th>Cross Reference Listing for MAP-Protocol</th>
<th>98-07-23 09:08:55</th>
<th>1</th>
</tr>
</thead>
</table>

| &extensionId............................identifier of Fieldspec | DEFINED in MAP-ExtensionDataTypes : | 24 |
| | USED in MAP-ExtensionDataTypes : | 41 |

| &ExtensionType..........................identifier of Fieldspec | DEFINED in MAP-ExtensionDataTypes : | 23 |
| | USED in MAP-ExtensionDataTypes : | 43 |

| abort...................................identifier of [APPLICATION 7] IMPLICIT Abort | DEFINED in TCAPMessages : | 56 |

| Abort...................................type reference SEQUENCE | DEFINED in TCAPMessages : | 74 |
| | USED in TCAPMessages : | 56 |

| absentSubscriber.......................value reference AbsentSubscriber, CHOICE VALUE | DEFINED in MAP-Protocol : | 259 |

| AbsentSubscriber.......................type reference ERROR | DEFINED in MAP-Errors : | 222 |
| | USED in MAP-Protocol : | 259 |
| | USED in MAP-CallHandlingOperat : | 33 70 88 |
| | USED in MAP-SupplementaryServi : | 46 184 198 |
| | USED in MAP-ShortMessageServic : | 36 |
| | USED in MAP-Errors : | 45 |

| absentSubscriber.......................identifier of Named Number, 1 | DEFINED in MAP-SM-DataTypes : | 117 |

| absentSubscriberDiagnosticSM............identifier of [0] AbsentSubscriberDiagnosticSM | DEFINED in MAP-SM-DataTypes : | 110 |

| absentSubscriberDiagnosticSM............identifier of AbsentSubscriberDiagnosticSM | DEFINED in MAP-ER-DataTypes : | 135 |

| AbsentSubscriberDiagnosticSM............type reference INTEGER | DEFINED in MAP-ER-DataTypes : | 139 |
| | USED in MAP-SM-DataTypes : | 39 110 |
| | USED in MAP-ER-DataTypes : | 43 135 |

| absentSubscriberParam..................identifier of AbsentSubscriberParam | DEFINED in MAP-Errors : | 224 |

| absentSubscriberParam..................type reference SEQUENCE | DEFINED in MAP-Errors : | 206 |
| | USED in MAP-Errors : | 101 224 |
| | USED in MAP-ER-DataTypes : | 34 |

| absentsubscriberSM.....................value reference AbsentSubscriberSM, CHOICE VALUE | | |
DEFINED in MAP-Protocol : 293

AbsentSubscriberSM...............type reference ERROR
DEFINED in MAP-Errors : 319
USED in MAP-Protocol : 119 293
USED in MAP-ShortMessageService : 41 79 109
USED in MAP-Errors : 70

absentSubscriberSM-Param........identifier of AbsentSubscriberSM-Param
DEFINED in MAP-Errors : 321

AbsentSubscriberSM-Param...............type reference SEQUENCE
DEFINED in MAP-Error-DataTypes : 134
USED in MAP-Errors : 111 321
USED in MAP-Error-DataTypes : 42
activateSS.................................value reference ActivateSS, CHOICE VALUE
DEFINED in MAP-Protocol : 186

ActivateSS.................................type reference OPERATION
DEFINED in MAP-SupplementaryService : 108
USED in MAP-Protocol : 54 186
USED in MAP-SupplementaryService : 15
activateTraceMode.........................value reference ActivateTraceMode, CHOICE VALUE
DEFINED in MAP-Protocol : 170

ActivateTraceMode.........................type reference OPERATION
DEFINED in MAP-OperationAndMainte : 50
USED in MAP-Protocol : 36 170
USED in MAP-OperationAndMainte : 13
activateTraceModeArg......................identifier of ActivateTraceModeArg
DEFINED in MAP-OperationAndMainte : 52

ActivateTraceModeArg......................type reference SEQUENCE
DEFINED in MAP-OM-DataTypes : 36
USED in MAP-OperationAndMainte : 34 52
USED in MAP-OM-DataTypes : 14
activateTraceModeRes......................identifier of ActivateTraceModeRes
DEFINED in MAP-OperationAndMainte : 54

ActivateTraceModeRes......................type reference SEQUENCE
DEFINED in MAP-OM-DataTypes : 50
USED in MAP-OperationAndMainte : 35 54
USED in MAP-OM-DataTypes : 15
AddressString...........................type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 73
USED in MAP-CommonDataTypes : 16 117
USED in MAP-OM-DataTypes : 21 40
USED in MAP-SS-DataTypes : 37 55
USED in MAP-SM-DataTypes : 30 54 98 103 108 128

ageOfLocationInformation...............identifier of AgeOfLocationInformation
DEFINED in MAP-MS-DataTypes : 632

AgeOfLocationInformation...............type reference INTEGER
DEFINED in MAP-MS-DataTypes : 640
USED in MAP-MS-DataTypes : 632

alertReason.............................identifier of AlertReason
DEFINED in MAP-MS-DataTypes : 146

AlertReason.............................type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 149
USED in MAP-MS-DataTypes : 26 146

alertServiceCentre.......................value reference AlertServiceCentre, CHOICE VALUE
DEFINED in MAP-Protocol : 204

AlertServiceCentre.......................type reference OPERATION
DEFINED in MAP-ShortMessageService : 123
USED in MAP-Protocol : 71 204
USED in MAP-ShortMessageService : 17

alertServiceCentreArg..................identifier of AlertServiceCentreArg
AlertServiceCentreArg....................type reference SEQUENCE
  DEFINED in MAP-SM-DataTypes : 126
  USED in MAP-ShortMessageService : 54 125
  USED in MAP-SM-DataTypes : 22

allAdditionalInfoTransferSS..............value reference SS-Code, '10000000'B
  DEFINED in MAP-SS-Code : 94

allAlternateSpeech-DataCDA..............value reference BearerServiceCode, '00110000'B
  DEFINED in MAP-BS-Code : 82

allAlternateSpeech-DataCDS..............value reference BearerServiceCode, '00111000'B
  DEFINED in MAP-BS-Code : 84

allAsynchronousServices.................value reference BearerServiceCode, '01100000'B
  DEFINED in MAP-BS-Code : 95

allBarringSS..............................value reference SS-Code, '10010000'B
  DEFINED in MAP-SS-Code : 101

allBearerServices.......................value reference BearerServiceCode, '00000000'B
  DEFINED in MAP-BS-Code : 49

allCallCompletionSS........................value reference SS-Code, '01000000'B
  DEFINED in MAP-SS-Code : 63

allCallOfferingSS........................value reference SS-Code, '00110000'B
  DEFINED in MAP-SS-Code : 54

allCallPrioritySS........................value reference SS-Code, '10100000'B
  DEFINED in MAP-SS-Code : 137

allChargingSS..............................value reference SS-Code, '01110000'B
  DEFINED in MAP-SS-Code : 86

allCommunityOfInterest-SS...............value reference SS-Code, '01100000'B
  DEFINED in MAP-SS-Code : 80

allCondForwardingSS............................value reference SS-Code, '00101000'B
  DEFINED in MAP-SS-Code : 45

allDataCDA-Services.....................value reference BearerServiceCode, '00010000'B
  DEFINED in MAP-BS-Code : 51

allDataCDS-Services.....................value reference BearerServiceCode, '00011000'B
  DEFINED in MAP-BS-Code : 60

allDataCircuitAsynchronous.............value reference BearerServiceCode, '01010000'B
  DEFINED in MAP-BS-Code : 92

allDataCircuitSynchronous.............value reference BearerServiceCode, '01011000'B
  DEFINED in MAP-BS-Code : 98

allDataPDS-Services.....................value reference BearerServiceCode, '00101000'B
  DEFINED in MAP-BS-Code : 76

allDataTeleservices.....................value reference TeleserviceCode, '01110000'B
  DEFINED in MAP-TS-Code : 55

allECT-Barred..............................identifier of Named Number, 9
  DEFINED in MAP-MS-DataTypes : 262

allFacsimileTransmissionServices........value reference TeleserviceCode, '01100000'B
  DEFINED in MAP-TS-Code : 48

allForwardingSS........................value reference SS-Code, '00100000'B
  DEFINED in MAP-SS-Code : 41

allLineIdentificationSS.................value reference SS-Code, '00010000'B
  DEFINED in MAP-SS-Code : 25

allMultiPartySS............................value reference SS-Code, '01010000'B
  DEFINED in MAP-SS-Code : 74

allOG-CallsBarred............................identifier of Named Number, 0
  DEFINED in MAP-MS-DataTypes : 253
allPadAccessCA-Services.................value reference BearerServiceCode, '00100000'B
DEFINED in MAP-BS-Code : 67

allPLMN-specificBS......................value reference BearerServiceCode, '11010000'B
DEFINED in MAP-BS-Code : 110

allPLMN-specificBS......................value reference SS-Code, '11110000'B
DEFINED in MAP-SS-Code : 120

allPLMN-specificTS......................value reference TeleserviceCode, '11010000'B
DEFINED in MAP-TS-Code : 72

allShortMessageServices..................value reference TeleserviceCode, '00100000'B
DEFINED in MAP-TS-Code : 44

allSpeechFollowedByDataCDA..............value reference BearerServiceCode, '01000000'B
DEFINED in MAP-BS-Code : 86

allSpeechFollowedByDataCDS..............value reference BearerServiceCode, '01001000'B
DEFINED in MAP-BS-Code : 88

allSpeechTransmissionServices..........value reference TeleserviceCode, '00010000'B
DEFINED in MAP-TS-Code : 40

allSS.....................................value reference SS-Code, '00000000'B
DEFINED in MAP-SS-Code : 21

allSynchronousServices..................value reference BearerServiceCode, '01101000'B
DEFINED in MAP-BS-Code : 101

allTeleservices.........................value reference TeleserviceCode, '00000000'B
DEFINED in MAP-TS-Code : 38

allTeleservices-ExceptSMS..............value reference TeleserviceCode, '10000000'B
DEFINED in MAP-TS-Code : 58

allVoiceGroupCallServices..............value reference TeleserviceCode, '10010000'B
DEFINED in MAP-TS-Code : 67

anyTimeInterrogation....................value reference AnyTimeInterrogation, CHOICE VALUE
DEFINED in MAP-Protocol : 213

AnyTimeInterrogation....................type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 154
USED in MAP-Protocol : 29 213
USED in MAP-MobileServiceOpera : 24

anyTimeInterrogationArg..................identifier of AnyTimeInterrogationArg
DEFINED in MAP-MobileServiceOpera : 156

AnyTimeInterrogationArg..................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 676
USED in MAP-MobileServiceOpera : 89 156
USED in MAP-MS-DataTypes : 67

anyTimeInterrogationRes..................identifier of AnyTimeInterrogationRes
DEFINED in MAP-MobileServiceOpera : 158

AnyTimeInterrogationRes..................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 683
USED in MAP-MobileServiceOpera : 90 158
USED in MAP-MS-DataTypes : 68

aocc.....................................value reference SS-Code, '01110010'B
DEFINED in MAP-SS-Code : 91

aoci.....................................value reference SS-Code, '01110001'B
DEFINED in MAP-SS-Code : 89

assumedIdle..............................identifier of [0] NULL
DEFINED in MAP-MS-DataTypes : 663

ati-NotAllowed.........................value reference ATI-NotAllowed, CHOICE VALUE
DEFINED in MAP-Protocol : 270

ATI-NotAllowed.........................type reference ERROR
DEFINED in MAP-Errors : 265
USED in MAP-Protocol : 105 270
USED in MAP-MobileServiceOpera : 62 161
USED in MAP-Errors : 52

ati-NotAllowedParam.....................identifier of ATI-NotAllowedParam
DEFINED in MAP-Errors : 267

ATI-NotAllowedParam.....................type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 226
USED in MAP-Errors : 108 267
USED in MAP-ER-DataTypes : 39

AuthenticationSet......................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 156
USED in MAP-MS-DataTypes : 154

authenticationSetList....................identifier of AuthenticationSetList
DEFINED in MAP-MS-DataTypes : 150

AuthenticationSetList...................type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 153
USED in MAP-MS-DataTypes : 150 191

automaticFacsimileGroup3.................value reference TeleserviceCode, '01100010'B
DEFINED in MAP-TS-Code : 50

badlyFormattedTransactionPortion........identifier of Named Number, 2
DEFINED in TCAPMessages : 105

badlyStructuredComponent.................identifier of Named Number, 2
DEFINED in TCAPMessages : 181

baic....................................value reference SS-Code, '10011010'B
DEFINED in MAP-SS-Code : 114

baoc....................................value reference SS-Code, '10010010'B
DEFINED in MAP-SS-Code : 105

barringOfIncomingCalls....................value reference SS-Code, '10011001'B
DEFINED in MAP-SS-Code : 112

barringOfOutgoingCalls...................value reference SS-Code, '10010001'B
DEFINED in MAP-SS-Code : 103

barringServiceActive......................identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes : 88

basicCall..............................identifier of Named Number, 0
DEFINED in MAP-CH-DataTypes : 88

basicService..........................identifier of Ext-BasicServiceCode
DEFINED in MAP-MS-DataTypes : 320

basicService..........................identifier of Ext-BasicServiceCode
DEFINED in MAP-MS-DataTypes : 395

basicService..........................identifier of Ext-BasicServiceCode
DEFINED in MAP-MS-DataTypes : 438

DEFINED in MAP-CH-DataTypes : 111

basicService..........................identifier of BasicServiceCode
DEFINED in MAP-SS-DataTypes : 54

basicService..........................identifier of BasicServiceCode
DEFINED in MAP-SS-DataTypes : 77

basicService..........................identifier of BasicServiceCode
DEFINED in MAP-SS-DataTypes : 131

basicService..........................identifier of BasicServiceCode
DEFINED in MAP-SS-DataTypes : 157

basicService..........................identifier of BasicServiceCode
DEFINED in MAP-ER-DataTypes : 109

BasicServiceCode........................type reference CHOICE
DEFINED in MAP-CH-DataTypes : 78

basicServiceGroup.......................identifier of [1] Ext-BasicServiceCode
DEFINED in MAP-CH-DataTypes : 150

basicServiceGroupList...................identifier of Ext-BasicServiceGroupList
DEFINED in MAP-MS-DataTypes : 413

basicServiceGroupList...................identifier of Ext-BasicServiceGroupList
DEFINED in MAP-MS-DataTypes : 460

basicServiceGroupList...................identifier of BasicServiceGroupList
DEFINED in MAP-SS-DataTypes : 139

basicServiceGroupList...................identifier of [2] BasicServiceGroupList
DEFINED in MAP-SS-DataTypes : 167

BasicServiceGroupList...................type reference SEQUENCE OF
DEFINED in MAP-SS-DataTypes : 212

basicServiceList........................identifier of [1] BasicServiceList
DEFINED in MAP-MS-DataTypes : 491

BasicServiceList........................type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 503

DEFINED in MAP-MS-DataTypes : 214

DEFINED in MAP-CommonDataTypes : 267

BearerServiceCode.......................type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 11

bearerServNotProvParam..................identifier of BearerServNotProvParam
DEFINED in MAP-Errors : 189

bearerServiceNotProvisioned...........value reference BearerServiceNotProvisioned, CHOICE
VALUE
DEFINED in MAP-Protocol : 237

BearerServiceNotProvisioned..............type reference ERROR
DEFINED in MAP-Errors : 187

bearerServNotProvParam..................identifier of BearerServNotProvParam
DEFINED in MAP-Errors : 189

BearerServNotProvParam....................type reference SEQUENCE
DEFINED in MAP-Errors : 190

begin.................................identifier of [APPLICATION 2] IMPLICIT Begin
DEFINED in TCAPMessages : 53
BEGIN......................type reference SEQUENCE
DEFINEd in TCAPMessages : 61
USED in TCAPMessages : 53

bicRoam.......................value reference SS-Code, '10011011'B
DEFINEd in MAP-SS-Code : 116

blackListed.....................identifier of Named Number, 1
DEFINEd in MAP-MS-DataTypes : 198

boic................................value reference SS-Code, '10010011'B
DEFINEd in MAP-SS-Code : 107

boicExHC............................value reference SS-Code, '10010100'B
DEFINEd in MAP-SS-Code : 109

broadcastInitEntitlement..............identifier of NULL
DEFINEd in MAP-MS-DataTypes : 597

bss-APDU................................identifier of ExternalSignalInfo
DEFINEd in MAP-MobileServiceOpera : 181

bss-APDU................................identifier of ExternalSignalInfo
DEFINEd in MAP-MobileServiceOpera : 186

bss-APDU................................identifier of ExternalSignalInfo
DEFINEd in MAP-MobileServiceOpera : 190

bss-APDU................................identifier of ExternalSignalInfo
DEFINEd in MAP-MobileServiceOpera : 196

bss-APDU................................identifier of ExternalSignalInfo
DEFINEd in MAP-MS-DataTypes : 173

bss-APDU................................identifier of ExternalSignalInfo
DEFINEd in MAP-MS-DataTypes : 178

bss-APDU................................identifier of ExternalSignalInfo
DEFINEd in MAP-MS-DataTypes : 184

busy...................................identifier of Named Number, 1
DEFINEd in MAP-CH-DataTypes : 97

busySubscriber......................value reference BusySubscriber, CHOICE VALUE
DEFINEd in MAP-Protocol : 260

BusySubscriber......................type reference ERROR
DEFINEd in MAP-Errors : 228
USED in MAP-Protocol : 99 260
USED in MAP-CallHandlingOperat : 34 71
USED in MAP-Errors : 43

busySubscriberParam....................identifier of BusySubscriberParam
DEFINEd in MAP-Errors : 230

BusySubscriberParam....................type reference SEQUENCE
DEFINEd in MAP-ER-DataTypes : 210
USED in MAP-Errors : 102 230
USED in MAP-ER-DataTypes : 35

callBarred.............................value reference CallBarred, CHOICE VALUE
DEFINEd in MAP-Protocol : 262

CallBarred.............................type reference ERROR
DEFINEd in MAP-Errors : 238
USED in MAP-Protocol : 101 262
USED in MAP-CallHandlingOperat : 36 73
USED in MAP-SupplementaryServ : 35 86 103 120 140 158 172 213
USED in MAP-ShortMessageServic : 37 78
USED in MAP-Errors : 46

callBarredParam.......................identifier of CallBarredParam
DEFINEd in MAP-Errors : 240

CallBarredParam.......................type reference CHOICE
DEFINEd in MAP-ER-DataTypes : 80
USED in MAP-Errors : 104 240
USED in MAP-ER-DataTypes : 15

callBarringCause................................identifier of CallBarringCause
DEFINED in MAP-ER-DataTypes : 81

CallBarringCause................................type reference ENUMERATED
DEFINED in MAP-ER-DataTypes : 87
USED in MAP-ER-DataTypes : 81 92

callBarringCause................................identifier of CallBarringCause
DEFINED in MAP-ER-DataTypes : 92

CallBarringFeature.............................type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 130
USED in MAP-SS-DataTypes : 128

callBarringFeatureList........................identifier of Ext-CallBarFeatureList
DEFINED in MAP-MS-DataTypes : 386

callBarringFeatureList........................identifier of CallBarringFeatureList
DEFINED in MAP-SS-DataTypes : 123

CallBarringFeatureList........................type reference SEQUENCE OF
DEFINED in MAP-SS-DataTypes : 126
USED in MAP-SS-DataTypes : 123

callBarringInfo...............................identifier of [1] Ext-CallBarInfo
DEFINED in MAP-MS-DataTypes : 284

callBarringInfo...............................identifier of [1] CallBarringInfo
DEFINED in MAP-SS-DataTypes : 64

CallBarringInfo...............................type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 121
USED in MAP-SS-DataTypes : 64

calledPartySS-InteractionViolation.............identifier of Named Number, 7
DEFINED in MAP-ER-DataTypes : 105

callReferenceNumber...........................identifier of [7] CallReferenceNumber
DEFINED in MAP-CH-DataTypes : 76

CallReferenceNumber.........................type reference OCTET STRING
DEFINED in MAP-CH-DataTypes : 93
USED in MAP-CH-DataTypes : 22 76 137 149

callReferenceNumber...........................identifier of [9] CallReferenceNumber
DEFINED in MAP-CH-DataTypes : 137

callReferenceNumber...........................identifier of [0] CallReferenceNumber
DEFINED in MAP-CH-DataTypes : 149

camelBusy........................................identifier of [1] NULL
DEFINED in MAP-MS-DataTypes : 664

DEFINED in MAP-CH-DataTypes : 80

CamelInfo........................................type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 162
USED in MAP-CH-DataTypes : 80

camelRoutingInfo..............................identifier of [8] CamelRoutingInfo
DEFINED in MAP-CH-DataTypes : 170

CamelRoutingInfo..............................type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 172
USED in MAP-CH-DataTypes : 170

camelSubscriptionInfoWithdraw...............identifier of [9] NULL
DEFINED in MAP-MS-DataTypes : 499

cancelLocation...............................value reference CancelLocation, CHOICE VALUE
DEFINED in MAP-Protocol : 129

CancelLocation...............................type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 119
USED in MAP-Protocol : 13 129
cancelLocationArg.......................identifier of CancelLocationArg
DEFINED in MAP-MobileServiceOpera : 121

CancelLocationArg.......................type reference CHOICE
DEFINED in MAP-MS-DataTypes : 133
USED in MAP-MobileServiceOpera : 71 121
USED in MAP-MS-DataTypes : 18

category................................identifier of [2] Category
DEFINED in MAP-MS-DataTypes : 212

Category................................type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 229
USED in MAP-MS-DataTypes : 212

type reference OCTET STRING

ccbs.................................value reference SS-Code, '01000011'B
DEFINED in MAP-SS-Code : 70

cellIdFixedLength.......................identifier of [0] CellIdFixedLength
DEFINED in MAP-CommonDataTypes : 241

CellIdFixedLength.......................type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 244
USED in MAP-CommonDataTypes : 241

ccIdOrLAI.............................identifier of [3] CellIdOrLAI
DEFINED in MAP-MS-DataTypes : 636

CellIdOrLAI.............................type reference CHOICE
DEFINED in MAP-CommonDataTypes : 240
USED in MAP-MS-DataTypes : 636 103
USED in MAP-CommonDataTypes : 34

cfb.................................value reference SS-Code, '00101001'B
DEFINED in MAP-SS-Code : 47

cfnrc.................................value reference SS-Code, '00101011'B
DEFINED in MAP-SS-Code : 51

cfry.................................value reference SS-Code, '00101010'B
DEFINED in MAP-SS-Code : 49

cfu.................................value reference SS-Code, '00100001'B
DEFINED in MAP-SS-Code : 43

chargeableECT-Barred....................identifier of Named Number, 10
DEFINED in MAP-MS-DataTypes : 263

checkIMEI...............................value reference CheckIMEI, CHOICE VALUE
DEFINED in MAP-Protocol : 151

CheckIMEI...............................type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 219
USED in MAP-Protocol : 22 151
USED in MAP-MobileServiceOpera : 37

clip.................................value reference SS-Code, '00010001'B
DEFINED in MAP-SS-Code : 28

crir.................................value reference SS-Code, '00010100'B
DEFINED in MAP-SS-Code : 30

cLiRestrictionOption....................identifier of [2] CliRestrictionOption
DEFINED in MAP-SS-DataTypes : 143

CliRestrictionOption....................type reference ENUMERATED
DEFINED in MAP-SS-DataTypes : 146
USED in MAP-SS-DataTypes : 27 143 162

cliRestrictionOption....................identifier of CliRestrictionOption
DEFINED in MAP-SS-DataTypes : 162

Cli-RestrictionInfo.....................type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 160
USED in MAP-SS-DataTypes : 169
cli-RestrictionInfo.....................identifier of [4] Cli-RestrictionInfo
  DEFINED in MAP-SS-DataTypes : 169

collectedInfo..........................identifier of Named Number, 2
  DEFINED in MAP-MS-DataTypes : 540

colp....................................value reference SS-Code, '00010011'B
  DEFINED in MAP-SS-Code : 32

colr....................................value reference SS-Code, '00010100'B
  DEFINED in MAP-SS-Code : 34

Component...............................type reference CHOICE
  DEFINED in TCAPMessages : 124
  USED in TCAPMessages : 47 115

ComponentPortion........................type reference [APPLICATION 12] IMPLICIT SEQUENCE OF
  DEFINED in TCAPMessages : 115
  USED in TCAPMessages : 59 63 67 72

components..............................identifier of ComponentPortion
  DEFINED in TCAPMessages : 59

components..............................identifier of ComponentPortion
  DEFINED in TCAPMessages : 63

components..............................identifier of ComponentPortion
  DEFINED in TCAPMessages : 67

components..............................identifier of ComponentPortion
  DEFINED in TCAPMessages : 72

Continue.................................type reference SEQUENCE
  DEFINED in TCAPMessages : 69
  USED in TCAPMessages : 55

continueCall............................identifier of Named Number, 0
  DEFINED in MAP-MS-DataTypes : 548

continue-ME.............................identifier of [APPLICATION 5] IMPLICIT Continue
  DEFINED in TCAPMessages : 55

controllingMSC..........................identifier of Named Number, 4
  DEFINED in MAP-CommonDataTypes : 233

cug.....................................value reference SS-Code, '01100001'B
  DEFINED in MAP-SS-Code : 83

cugIC-CallBarred........................identifier of Named Number, 1
  DEFINED in MAP-MS-DataTypes : 424

cugOG-CallBarred........................identifier of Named Number, 2
  DEFINED in MAP-MS-DataTypes : 425

cugSubscriptionFlag.....................identifier of [6] NULL
  DEFINED in MAP-CH-DataTypes : 108

CUG-CheckInfo...........................type reference SEQUENCE
  DEFINED in MAP-CH-DataTypes : 60
  USED in MAP-CH-DataTypes : 70 107 153

cug-CheckInfo...........................identifier of CUG-CheckInfo
  DEFINED in MAP-CH-DataTypes : 70

CUG-CheckInfo.........................identifier of CUG-CheckInfo
  DEFINED in MAP-CH-DataTypes : 107

CUG-CheckInfo.........................identifier of CUG-CheckInfo
  DEFINED in MAP-CH-DataTypes : 153

CUG-Feature............................type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes : 437
  USED in MAP-MS-DataTypes : 430

CUG-FeatureList.........................identifier of CUG-FeatureList
  DEFINED in MAP-MS-DataTypes : 402

CUG-FeatureList..........................type reference SEQUENCE OF
<table>
<thead>
<tr>
<th>Definition</th>
<th>Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>cug-Index</strong></td>
<td>Identifier</td>
<td>CUG-Index</td>
</tr>
<tr>
<td><strong>CUG-Index</strong></td>
<td>Type Reference</td>
<td>INTEGER</td>
</tr>
<tr>
<td><strong>cug-Info</strong></td>
<td>Identifier</td>
<td>CUG-Info</td>
</tr>
<tr>
<td><strong>CUG-Info</strong></td>
<td>Type Reference</td>
<td>SEQUENCE</td>
</tr>
<tr>
<td><strong>cug-Interlock</strong></td>
<td>Identifier</td>
<td>CUG-Interlock</td>
</tr>
<tr>
<td><strong>CUG-Interlock</strong></td>
<td>Type Reference</td>
<td>OCTET STRING</td>
</tr>
<tr>
<td><strong>cug-OutgoingAccess</strong></td>
<td>Identifier</td>
<td>NULL</td>
</tr>
<tr>
<td><strong>cug-Reject</strong></td>
<td>Value Reference</td>
<td>CUG-Reject, CHOICE VALUE</td>
</tr>
<tr>
<td><strong>CUG-Reject</strong></td>
<td>Type Reference</td>
<td>ERROR</td>
</tr>
<tr>
<td><strong>cug-RejectCause</strong></td>
<td>Identifier</td>
<td>CUG-RejectCause</td>
</tr>
<tr>
<td><strong>cug-RejectParam</strong></td>
<td>Identifier</td>
<td>CUG-RejectParam</td>
</tr>
<tr>
<td><strong>CUG-Subscription</strong></td>
<td>Type Reference</td>
<td>SEQUENCE</td>
</tr>
<tr>
<td><strong>CUG-SubscriptionList</strong></td>
<td>Identifier</td>
<td>CUG-SubscriptionList</td>
</tr>
<tr>
<td><strong>currentPassword</strong></td>
<td>Identifier</td>
<td>Password</td>
</tr>
<tr>
<td><strong>cw</strong></td>
<td>Value Reference</td>
<td>SS-Code, '01000001'B</td>
</tr>
<tr>
<td><strong>dataCDA-1200bps</strong></td>
<td>Value Reference</td>
<td>BearerServiceCode, '00010010'B</td>
</tr>
<tr>
<td><strong>dataCDA-1200-75bps</strong></td>
<td>Value Reference</td>
<td>BearerServiceCode, '00010011'B</td>
</tr>
</tbody>
</table>
dataCDA-2400bps......value reference BearerServiceCode, '00010100'B
  DEFINED in MAP-BS-Code : 55

dataCDA-300bps..............value reference BearerServiceCode, '00010001'B
  DEFINED in MAP-BS-Code : 52

dataCDA-4800bps..............value reference BearerServiceCode, '00010101'B
  DEFINED in MAP-BS-Code : 56

dataCDA-9600bps..............value reference BearerServiceCode, '00010110'B
  DEFINED in MAP-BS-Code : 57

dataCDS-1200bps..............value reference BearerServiceCode, '00011010'B
  DEFINED in MAP-BS-Code : 61

dataCDS-2400bps..............value reference BearerServiceCode, '00011100'B
  DEFINED in MAP-BS-Code : 62

dataCDS-4800bps..............value reference BearerServiceCode, '00011101'B
  DEFINED in MAP-BS-Code : 63

dataCDS-9600bps..............value reference BearerServiceCode, '00011110'B
  DEFINED in MAP-BS-Code : 64

dataMissing..................value reference DataMissing, CHOICE VALUE
  DEFINED in MAP-Protocol : 218

DataMissing..................type reference ERROR
  DEFINED in MAP-Errors : 126
  USED in MAP-Protocol : 80 218
  USED in MAP-MobileServiceOpera : 55 114 124 138 149 162 175 199 213
  226 238 249 268
  USED in MAP-OperationAndMainte : 24 58 72 83
  Used in MAP-CallHandlingOperat : 24 62 84
  Used in MAP-SupplementaryServi : 31 82 99 116 136 154 169 182 196
  211
  USED in MAP-ShortMessageServic : 28 73 101 118 129 141
  USED in MAP-Errors : 15

dataMissingParam...............identifier of DataMissingParam
  DEFINED in MAP-Errors : 128

DataMissingParam...............type reference SEQUENCE
  DEFINED in MAP-ER-DataTypes : 154
  USED in MAP-Errors : 87 128
  USED in MAP-ER-DataTypes : 21

dataPDS-2400bps..............value reference BearerServiceCode, '00101100'B
  DEFINED in MAP-BS-Code : 77

dataPDS-4800bps..............value reference BearerServiceCode, '00101101'B
  DEFINED in MAP-BS-Code : 78

dataPDS-9600bps..............value reference BearerServiceCode, '00101110'B
  DEFINED in MAP-BS-Code : 79

deactivateSS..................value reference DeactivateSS, CHOICE VALUE
  DEFINED in MAP-Protocol : 187

DeactivateSS..................type reference OPERATION
  DEFINED in MAP-SupplementaryServi : 128
  USED in MAP-Protocol : 55 187
  USED in MAP-SupplementaryServi : 16

deactivateTraceMode............value reference DeactivateTraceMode, CHOICE VALUE
  DEFINED in MAP-Protocol : 171

DeactivateTraceMode............type reference OPERATION
  DEFINED in MAP-OperationAndMainte : 64
  USED in MAP-Protocol : 37 171
  USED in MAP-OperationAndMainte : 14

deactivateTraceModeArg............identifier of DeactivateTraceModeArg
  DEFINED in MAP-OperationAndMainte : 66

DeactivateTraceModeArg............type reference SEQUENCE
  DEFINED in MAP-OM-DataTypes : 54
  USED in MAP-OperationAndMainte : 36 66
deactivateTraceModeRes..................identifier of DeactivateTraceModeRes
DEFINED in MAP-OperationAndMainte : 68
DeactivateTraceModeRes..................type reference SEQUENCE
DEFINED in MAP-OM-DataTypes : 60
USED in MAP-OperationAndMainte : 37 68
USED in MAP-OM-DataTypes : 17
defaultCallHandling.....................identifier of [1] DefaultCallHandling
DEFINED in MAP-MS-DataTypes : 533
DefaultCallHandling.....................type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 547
USED in MAP-MS-DataTypes : 46 533
USED in MAP-CH-DataTypes : 29 196
defaultCallHandling.....................identifier of [1] DefaultCallHandling
DEFINED in MAP-CH-DataTypes : 196
defaultPriority.........................identifier of EMLPP-Priority
DEFINED in MAP-MS-DataTypes : 291
deleteSubscriberData....................value reference DeleteSubscriberData, CHOICE VALUE
DEFINED in MAP-Protocol : 157
DeleteSubscriberData.....................type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 242
USED in MAP-Protocol : 24 157
USED in MAP-MobileServiceOpera : 41
deleteSubscriberDataArg..................identifier of DeleteSubscriberDataArg
DEFINED in MAP-MobileServiceOpera : 244
DeleteSubscriberDataArg..................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 489
USED in MAP-MobileServiceOpera : 82 244
USED in MAP-MS-DataTypes : 37
deleteSubscriberDataRes..................identifier of DeleteSubscriberDataRes
DEFINED in MAP-MobileServiceOpera : 246
DeleteSubscriberDataRes..................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 508
USED in MAP-MobileServiceOpera : 83 246
USED in MAP-MS-DataTypes : 38
derivable...............................identifier of InvokeIdType
DEFINED in TCAPMessages : 167
DestTransactionID.......................type reference [APPLICATION 9] IMPLICIT
TransactionID
DEFINED in TCAPMessages : 98
USED in TCAPMessages : 65 70 74
diagnosticInfo..........................identifier of SignalInfo
DEFINED in MAP-ER-DataTypes : 130
dialoguePortion..........................identifier of DialoguePortion
DEFINED in TCAPMessages : 58
dialoguePortion..........................identifier of DialoguePortion
DEFINED in TCAPMessages : 62
dialoguePortion..........................identifier of DialoguePortion
DEFINED in TCAPMessages : 66
dialoguePortion..........................identifier of DialoguePortion
DEFINED in TCAPMessages : 71
dialoguePortion..........................identifier of DialoguePortion
DEFINED in TCAPMessages : 77
DialoguePortion.........................type reference [APPLICATION 11] EXTERNAL
DEFINED in TCAPMessages : 82
USED in TCAPMessages : 58 62 66 71 77
doublyChargeableECT-Barred.............identifier of Named Number, 13
  DEFINED in MAP-MS-DataTypes : 266
dtid.....................................identifier of DestTransactionID
  DEFINED in TCAPMessages : 65
dtid.....................................identifier of DestTransactionID
  DEFINED in TCAPMessages : 70
dtid.....................................identifier of DestTransactionID
  DEFINED in TCAPMessages : 74
duplicateInvokeID....................identifier of Named Number, 0
  DEFINED in TCAPMessages : 183
ect.....................................value reference SS-Code, '00110001'B
  DEFINED in MAP-SS-Code : 57
eir.....................................identifier of Named Number, 6
  DEFINED in MAP-CommonDataTypes : 235
emergencyCalls.......................value reference TeleserviceCode, '00010010'B
  DEFINED in MAP-TS-Code : 42
emplpp..................................value reference SS-Code, '10100001'B
  DEFINED in MAP-SS-Code : 140
  DEFINED in MAP-MS-DataTypes : 287
EMLPP-Info................................type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes : 289
  USED in MAP-MS-DataTypes : 287
EMLPP-Priority..........................type reference INTEGER
  DEFINED in MAP-MS-DataTypes : 295
  USED in MAP-MS-DataTypes : 290 291 301 302 303 304 305 306 307
End....................................type reference SEQUENCE
  DEFINED in TCAPMessages : 65
  USED in TCAPMessages : 54
end-ME.................................identifier of [APPLICATION 4] IMPLICIT End
  DEFINED in TCAPMessages : 54
enterNewPW..............................identifier of Named Number, 1
  DEFINED in MAP-SS-DataTypes : 198
enterNewPW-Again........................identifier of Named Number, 2
  DEFINED in MAP-SS-DataTypes : 199
enterPW..................................identifier of Named Number, 0
  DEFINED in MAP-SS-DataTypes : 197
equipmentNotSM-Equipped................identifier of Named Number, 2
  DEFINED in MAP-ER-DataTypes : 122
equipmentProtocolError..................identifier of Named Number, 1
  DEFINED in MAP-ER-DataTypes : 121
equipmentStatus........................identifier of EquipmentStatus
  DEFINED in MAP-MobileServiceOpera : 223
EquipmentStatus........................type reference ENUMERATED
  DEFINED in MAP-MS-DataTypes : 196
  USED in MAP-MobileServiceOpera : 79 223
  USED in MAP-MS-DataTypes : 32
eraseSS..................................value reference EraseSS, CHOICE VALUE
  DEFINED in MAP-Protocol : 185
EraseSS..................................type reference OPERATION
  DEFINED in MAP-SupplementaryServic : 91
  USED in MAP-Protocol : 53 185
  USED in MAP-SupplementaryServic : 14
erErrorCode............................identifier of ERROR
  DEFINED in TCAPMessages : 158
ets-300102-1..........................identifier of Named Number, 4
DEFINED in MAP-CommonDataTypes : 186

extendedRoutingInfo.....................identifier of ExtendedRoutingInfo
DEFINED in MAP-CH-DataTypes : 106

ExtendedRoutingInfo.....................type reference CHOICE
DEFINED in MAP-CH-DataTypes : 168
USED in MAP-CH-DataTypes : 106

tableContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 125

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 131

extensionContainer......................identifier of [14] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 207

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 249

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 292

extensionContainer......................identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 312

extensionContainer......................identifier of [9] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 326

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 387

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 397

extensionContainer......................identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 403

extensionContainer......................identifier of [0] ExtensionContainer
DEFINED in MAP-MS-DataTypes : 414

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 441

DEFINED in MAP-MS-DataTypes : 461

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 480

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 500

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 511

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 516
DEFINED in MAP-MS-DataTypes : 534

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 569

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 575

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 592

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 598

DEFINED in MAP-MS-DataTypes : 611

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 616

DEFINED in MAP-MS-DataTypes : 622

DEFINED in MAP-MS-DataTypes : 628

DEFINED in MAP-MS-DataTypes : 637

DEFINED in MAP-MS-DataTypes : 680

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-MS-DataTypes : 685

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-CommonDataTypes : 167

DEFINED in MAP-OM-DataTypes : 41

extensionContainer......................identifier of [0] ExtensionContainer
DEFINED in MAP-OM-DataTypes : 51

DEFINED in MAP-OM-DataTypes : 57

extensionContainer......................identifier of [0] ExtensionContainer
DEFINED in MAP-OM-DataTypes : 61

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes : 63

DEFINED in MAP-CH-DataTypes : 82

extensionContainer......................identifier of [0] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 114

extensionContainer......................identifier of [7] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 125

DEFINED in MAP-CH-DataTypes : 139

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes : 144

extensionContainer......................identifier of [7] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 155

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes : 159

extensionContainer......................identifier of ExtensionContainer
extensionContainer......................identifier of [1] ExtensionContainer
DEFINED in MAP-CH-DataTypes : 165

DEFINED in MAP-CH-DataTypes : 175

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes : 181

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-CH-DataTypes : 186

DEFINED in MAP-CH-DataTypes : 197

DEFINED in MAP-SM-DataTypes : 55

DEFINED in MAP-SM-DataTypes : 61

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes : 67

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes : 74

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes : 79

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes : 87

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes : 92

extensionContainer......................identifier of [1] ExtensionContainer
DEFINED in MAP-SM-DataTypes : 112

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes : 122

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-SM-DataTypes : 134

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 73

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 93

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 98

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 131

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 136

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 151

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 155

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 159

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 163

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 167

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 171
extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 175

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 179

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 183

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 187

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 191

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 195

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 199

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 203

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 207

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 211

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 215

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 219

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 223

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 227

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 231

extensionContainer......................identifier of ExtensionContainer
DEFINED in MAP-ER-DataTypes : 235

ExtensionContainer......................type reference SEQUENCE
DEFINED in MAP-ExtensionDataTypes : 32
USED in MAP-MS-DataTypes : 110 125 131 207 249 292 312 326 387
397 403 414 441 461 480 500 511 516
521 534 569 575 592 598 611 616 622
628 637 680 685

USED in MAP-CommonDataTypes : 54 167

USED in MAP-OM-DataTypes : 27 41 51 57 61

USED in MAP-CH-DataTypes : 53 63 82 114 125 139 144 155 159
165 175 181 186 197

USED in MAP-SM-DataTypes : 44 55 61 67 74 79 87 92 112
122 134

USED in MAP-ER-DataTypes : 65 73 93 98 131 136 151 155 159
163 167 171 175 179 183 187 191 195
199 203 207 211 215 219 223 227 231
235

USED in MAP-ExtensionDataTypes : 16

ExtensionSet.............information object set reference MAP-EXTENSION, Information Object
Set
DEFINED in MAP-ExtensionDataTypes : 48
USED in MAP-ExtensionDataTypes : 42 44

ExternalSignalInfo......................type reference SEQUENCE
DEFINED in MAP-CommonDataTypes : 162
USED in MAP-MobileServiceOpera : 96 181 186 190 196

USED in MAP-MS-DataTypes : 98 173 178 184

USED in MAP-CommonDataTypes : 19

USED in MAP-CH-DataTypes : 45 79 133 134
extId.................................identifier of InformationObjectClassField
DEFINED in MAP-ExtensionDataTypes : 41

extType...............................identifier of InformationObjectClassField
DEFINED in MAP-ExtensionDataTypes : 43

Ext-BasicServiceCode.....................type reference CHOICE
DEFINED in MAP-CommonDataTypes : 270
USED in MAP-MS-DataTypes : 104 320 395 433 438 504
USED in MAP-CommonDataTypes : 38
USED in MAP-CH-DataTypes : 48 78 111 150

Ext-BasicServiceGroupList................type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 432
USED in MAP-MS-DataTypes : 413 460

DEFINED in MAP-CommonDataTypes : 271

Ext-BearerServiceCode......................type reference OCTET STRING
DEFINED in MAP-BS-Code : 25
USED in MAP-MS-DataTypes : 85 237
USED in MAP-CommonDataTypes : 49 271

Ext-CallBarFeatureList....................type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 390
USED in MAP-MS-DataTypes : 386

Ext-CallBarInfo........................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 384
USED in MAP-MS-DataTypes : 284

Ext-CallBarringFeature....................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 394
USED in MAP-MS-DataTypes : 392

Ext-ForwFeature..........................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 319
USED in MAP-MS-DataTypes : 317

Ext-ForwFeatureList......................type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 315
USED in MAP-MS-DataTypes : 311

Ext-ForwInfo............................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 309
USED in MAP-MS-DataTypes : 283

Ext-ForwOptions.........................type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 350
USED in MAP-MS-DataTypes : 324

Ext-NoRepCondTime.......................type reference INTEGER
DEFINED in MAP-MS-DataTypes : 377
USED in MAP-MS-DataTypes : 325

Ext-SS-Data..............................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 456
USED in MAP-MS-DataTypes : 286

Ext-SS-Info.............................type reference CHOICE
DEFINED in MAP-MS-DataTypes : 282
USED in MAP-MS-DataTypes : 280

Ext-SS-InfoList.........................type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 279
USED in MAP-MS-DataTypes : 220

Ext-SS-Status...........................type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 329
USED in MAP-MS-DataTypes : 321 396 458

DEFINED in MAP-CommonDataTypes : 272

Ext-TeleserviceCode.......................type reference OCTET STRING
DEFINED in MAP-TS-Code : 20
USED in MAP-MS-DataTypes : 90 242
facilityNotSupParam.........................identifier of FacilityNotSupParam
DEFINEd in MAP-Errors : 140

FacilityNotSupParam.........................type reference SEQUENCE
DEFINEd in MAP-ER-DataTypes : 162
USED in MAP-Errors : 89 140
USED in MAP-ER-DataTypes : 23

collectedNameSupported......................value reference FacilityNotSupported, CHOICE VALUE
DEFINEd in MAP-Protocol : 220

FacilityNotSupported.......................type reference ERROR
DEFINEd in MAP-Errors : 138
USED in MAP-Protocol : 82 220
USED in MAP-OperationAndMainte : 26 60 74
USED in MAP-CallHandlingOperat : 26 64 86
USED in MAP-ShortMessageServic : 30 75 90 103 143
USED in MAP-Errors : 17

facsimileGroup3AndAlterSpeech...............value reference TeleserviceCode, '01100001'B
DEFINEd in MAP-TS-Code : 49

facsimileGroup4.............................value reference TeleserviceCode, '01100011'B
DEFINEd in MAP-TS-Code : 51

forwardAccessSignalling.....................value reference ForwardAccessSignalling, CHOICE VALUE
DEFINEd in MAP-Protocol : 139

ForwardAccessSignalling.....................type reference OPERATION
DEFINEd in MAP-MobileServiceOpera : 188
USED in MAP-Protocol : 19 139
USED in MAP-MobileServiceOpera : 30

forwardCheckSS-Indication....................value reference ForwardCheckSS-Indication, CHOICE VALUE
DEFINEd in MAP-Protocol : 163

ForwardCheckSS-Indication....................type reference OPERATION
DEFINEd in MAP-MobileServiceOpera : 259
USED in MAP-Protocol : 26 163
USED in MAP-MobileServiceOpera : 45

forwardedToNumber..........................identifier of [5] ISDN-AddressString
DEFINEd in MAP-MS-DataTypes : 322

forwardedToNumber..........................identifier of [5] ISDN-AddressString
DEFINEd in MAP-CH-DataTypes : 122

forwardedToNumber..........................identifier of [4] AddressString
DEFINEd in MAP-SS-DataTypes : 55

forwardedToNumber..........................identifier of [5] ISDN-AddressString
DEFINEd in MAP-SS-DataTypes : 79

forwardedToSubaddress.......................identifier of [8] ISDN-SubaddressString
DEFINEd in MAP-MS-DataTypes : 323

forwardedToSubaddress.......................identifier of [4] ISDN-SubaddressString
DEFINEd in MAP-CH-DataTypes : 123

forwardedToSubaddress.......................identifier of [6] ISDN-SubaddressString
DEFINEd in MAP-SS-DataTypes : 56

forwardedToSubaddress.......................identifier of [8] ISDN-SubaddressString
DEFINEd in MAP-SS-DataTypes : 80

forwarding.................................identifier of Named Number, 1
DEFINEd in MAP-CH-DataTypes : 89

forwardingData..............................identifier of ForwardingData
DEFINEd in MAP-CH-DataTypes : 119

ForwardingData.............................type reference SEQUENCE
DEFINEd in MAP-CH-DataTypes : 121
USED in MAP-CH-DataTypes : 119 151 173
forwardingData........................identifier of [2] ForwardingData
DEFINED in MAP-CH-DataTypes : 151

forwardingData........................identifier of ForwardingData
DEFINED in MAP-CH-DataTypes : 173

forwardingFailed........................value reference ForwardingFailed, CHOICE VALUE
DEFINED in MAP-Protocol : 263

ForwardingFailed.........................type reference ERROR
DEFINED in MAP-Errors : 248
USED in MAP-Protocol : 103 263
USED in MAP-CallHandlingOperat : 38 97
USED in MAP-Errors : 48

forwardingFailedParam...................identifier of ForwardingFailedParam
DEFINED in MAP-Errors : 250

ForwardingFailedParam...................type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 222
USED in MAP-Errors : 106 250
USED in MAP-ER-DataTypes : 38

ForwardingFeature.......................type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 76
USED in MAP-SS-DataTypes : 74

forwardingFeatureList...................identifier of Ext-ForwFeatureList
DEFINED in MAP-MS-DataTypes : 311

forwardingFeatureList...................identifier of ForwardingFeatureList
DEFINED in MAP-SS-DataTypes : 69

ForwardingFeatureList...................type reference SEQUENCE OF
DEFINED in MAP-SS-DataTypes : 72
USED in MAP-SS-DataTypes : 69 168

forwardingFeatureList...................identifier of [3] ForwardingFeatureList
DEFINED in MAP-SS-DataTypes : 168

forwardingInfo..........................identifier of [0] Ext-ForwInfo
DEFINED in MAP-MS-DataTypes : 283

forwardingInfo..........................identifier of [0] ForwardingInfo
DEFINED in MAP-SS-DataTypes : 63

ForwardingInfo.........................type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 67
USED in MAP-SS-DataTypes : 63

forwardingInterrogationRequired........identifier of [4] NULL
DEFINED in MAP-CH-DataTypes : 112

forwardingOptions.......................identifier of [6] Ext-ForwOptions
DEFINED in MAP-MS-DataTypes : 324

forwardingOptions.......................identifier of [6] ForwardingOptions
DEFINED in MAP-CH-DataTypes : 124

forwardingOptions.......................identifier of [6] ForwardingOptions
DEFINED in MAP-SS-DataTypes : 81

ForwardingOptions.......................type reference OCTET STRING
DEFINED in MAP-SS-DataTypes : 100
USED in MAP-CH-DataTypes : 37 124
USED in MAP-SS-DataTypes : 29 81

DEFINED in MAP-CH-DataTypes : 77

ForwardingReason.........................type reference ENUMERATED
DEFINED in MAP-CH-DataTypes : 95
USED in MAP-CH-DataTypes : 77

forwardingViolation.....................value reference ForwardingViolation, CHOICE VALUE
DEFINED in MAP-Protocol : 265

ForwardingViolation......................type reference ERROR
Defined in MAP-Errors: 243
Used in MAP-Protocol: 102 265
Used in MAP-CallHandlingOperat: 37 75
Used in MAP-Errors: 47

forwadingViolationParam: identifier of ForwardingViolationParam
Defined in MAP-Errors: 245

ForwardingViolationParam: type reference SEQUENCE
Defined in MAP-ER-DataTypes: 218
Used in MAP-Errors: 105 245
Used in MAP-ER-DataTypes: 37

geographicalInformation: identifier of [0] IMPLICIT GeographicalInformation
Defined in MAP-MS-DataTypes: 633
Used in MAP-MS-DataTypes: 102 171 182
Used in MAP-CommonDataTypes: 30

getPassword: value reference GetPassword, CHOICE VALUE
Defined in MAP-Protocol: 194

GlobalCellId: type reference OCTET STRING
Defined in MAP-CommonDataTypes: 218
Used in MAP-MS-DataTypes: 102 171 182
Used in MAP-CommonDataTypes: 30

greyListed: identifier of Named Number, 2
Defined in MAP-MS-DataTypes: 199

groupId: identifier of GroupId
Defined in MAP-MS-DataTypes: 591

GroupId: type reference OCTET STRING
Defined in MAP-MS-DataTypes: 601
Used in MAP-MS-DataTypes: 591 596
gsmSCF-Address....................identifier of [0] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 532

7

gsmSCF-Address....................identifier of [3] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 679

8

gsmSCF-Address....................identifier of [0] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 195

10

gsm-0408.............................identifier of Named Number, 1
DEFINED in MAP-CommonDataTypes : 182

12

gsm-0806.............................identifier of Named Number, 2
DEFINED in MAP-CommonDataTypes : 183

14

DEFINED in MAP-CH-DataTypes : 133

16

gsm-BSSMAP.............................identifier of Named Number, 3
DEFINED in MAP-CommonDataTypes : 184

18

guidanceInfo........................identifier of GuidanceInfo
DEFINED in MAP-SupplementaryServi : 223

20

GuidanceInfo........................type reference ENUMERATED
DEFINED in MAP-SS-DataTypes : 196
USED in MAP-SupplementaryServi : 60 223
USED in MAP-SS-DataTypes : 23

24

handoverNumber.....................identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 177

26

hlr.....................................identifier of Named Number, 1
DEFINED in MAP-CommonDataTypes : 230

28

HLR-Id................................type reference IMSI
DEFINED in MAP-CommonDataTypes : 207
USED in MAP-CommonDataTypes : 212

30

hlr-List................................identifier of HLR-List
DEFINED in MAP-MS-DataTypes : 563

32

HLR-List...............................type reference HLR-List SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 211
USED in MAP-MS-DataTypes : 100 563
USED in MAP-CommonDataTypes : 28

36

hlr-Number............................identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 129

38

hlr-Number............................identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 562

40

hlr-Number............................identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 573

42

hold.................................value reference SS-Code, '01000010'B
DEFINED in MAP-SS-Code : 68

44

ho-NumberNotRequired...............identifier of NULL
DEFINED in MAP-MS-DataTypes : 172

46

illegalEquipment....................value reference IllegalEquipment, CHOICE VALUE
DEFINED in MAP-Protocol : 236

48

IllegalEquipment....................type reference ERROR
DEFINED in MAP-Errors : 181
USED in MAP-Protocol : 90 236
USED in MAP-SupplementaryServi : 48 186 200
USED in MAP-ShortMessageServi : 34 106
USED in MAP-Errors : 29

52

illegalEquipmentParam.............identifier of IllegalEquipmentParam
DEFINED in MAP-Errors : 183

54

IllegalEquipmentParam.............type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 186
USED in MAP-Errors : 95 183
<table>
<thead>
<tr>
<th>Identifier</th>
<th>Type Reference</th>
<th>Definition Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>illegalSS-Operation</td>
<td>value reference IllegalSS-Operation, CHOICE VALUE</td>
<td>defined in MAP-Protocol: 275</td>
</tr>
<tr>
<td>IllegalSS-Operation</td>
<td>type reference ERROR</td>
<td>defined in MAP-Errors: 273, used in MAP-Protocol: 106, 275, MAP-SupplementaryService: 36, 87, 104, 121, 141, 159</td>
</tr>
<tr>
<td>illegalSubscriber</td>
<td>value reference IllegalSubscriber, CHOICE VALUE</td>
<td>defined in MAP-Protocol: 235</td>
</tr>
<tr>
<td>IllegalSubscriber</td>
<td>type reference ERROR</td>
<td>defined in MAP-Errors: 175, used in MAP-Protocol: 89, 235, MAP-SupplementaryService: 47, 185, 199</td>
</tr>
<tr>
<td>IllegalSubscriberParam</td>
<td>identifier of IllegalSubscriberParam</td>
<td>defined in MAP-Errors: 94, 177, used in MAP-Protocol: 89, 235, MAP-SupplementaryService: 47, 185, 199</td>
</tr>
<tr>
<td>imei</td>
<td>identifier of IMEI</td>
<td>defined in MAP-MobileServiceOperation: 221, used in MAP-Protocol: 98, 221</td>
</tr>
<tr>
<td>IMEI</td>
<td>type reference TBCD-STRING</td>
<td>defined in MAP-CommonDataTypes: 200, used in MAP-MobileServiceOperation: 98, 221</td>
</tr>
<tr>
<td>imsi</td>
<td>identifier of IMSI</td>
<td>defined in MAP-OperationAndMaintenance: 81, used in MAP-Protocol: 99, 121, 134, 143, 149, 189, 205, 490</td>
</tr>
</tbody>
</table>

From the table, it appears that identifiers are used extensively in various protocols and supplementary services, with some appearing multiple times across different fields. This indicates a complex interplay of identifiers in the communication protocols discussed.
USED in MAP-SM-DataTypes : 33 59 96 145

imsi.................................identifier of [0] IMSI
DEFINED in MAP-CommonDataTypes : 197

imsi.................................identifier of [0] IMSI
DEFINED in MAP-OM-DataTypes : 37

imsi.................................identifier of [0] IMSI
DEFINED in MAP-OM-DataTypes : 55

imsi.................................identifier of [9] IMSI
DEFINED in MAP-CH-DataTypes : 102

imsi.................................identifier of [0] IMSI
DEFINED in MAP-CH-DataTypes : 129

imsi.................................identifier of [3] IMSI
DEFINED in MAP-CH-DataTypes : 152

imsi.................................identifier of IMSI
DEFINED in MAP-SM-DataTypes : 59

imsi.................................identifier of [0] IMSI
DEFINED in MAP-SM-DataTypes : 96

imsi.................................identifier of [0] IMSI
DEFINED in MAP-SM-DataTypes : 145

imsiDetached.........................identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 670

IMSI-WithLMSI.........................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 135

imsi-WithLMSI............................identifier of IMSI-WithLMSI
DEFINED in MAP-MS-DataTypes : 142

incomingCallsBarredWithinCUG............identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes : 102

incorrectTransactionPortion.............identifier of Named Number, 3
DEFINED in TCAPMessages : 106

informServiceCentre.....................value reference InformServiceCentre, CHOICE VALUE
DEFINED in MAP-Protocol : 203

InformServiceCentre....................type reference OPERATION
DEFINED in MAP-ShortMessageServic : 132

InformServiceCentre....................type reference OPERATION
USED in MAP-Protocol : 72 203

InformServiceCentre....................type reference OPERATION
USED in MAP-ShortMessageServic : 18

informServiceCentreArg...................identifier of InformServiceCentreArg
DEFINED in MAP-SM-DataTypes : 134

InformServiceCentreArg...................type reference SEQUENCE
DEFINED in MAP-SM-DataTypes : 131

informServiceCentreArg...................type reference SEQUENCE
USED in MAP-ShortMessageServic : 55 134

initiatingRelease......................identifier of Named Number, 4
DEFINED in TCAPMessages : 187

insertSubscriberData....................value reference InsertSubscriberData, CHOICE VALUE
DEFINED in MAP-Protocol : 156

InsertSubscriberData....................type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 231

InsertSubscriberData....................type reference OPERATION
USED in MAP-Protocol : 23 156

InsertSubscriberData....................type reference OPERATION
USED in MAP-MobileServiceOpera : 40

insertSubscriberDataArg..................identifier of InsertSubscriberDataArg
DEFINED in MAP-MobileServiceOpera : 233

InsertSubscriberDataArg..................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 204

insertSubscriberDataArg..................type reference SEQUENCE
USED in MAP-MobileServiceOpera : 80 233
USED in MAP-MS-DataTypes : 35

insertSubscriberDataRes.................identifier of InsertSubscriberDataRes
DEFINED in MAP-MobileServiceOpera : 235

InsertSubscriberDataRes.................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 472
USED in MAP-MobileServiceOpera : 81 235
USED in MAP-MS-DataTypes : 36

interCUG-Restrictions.................identifier of InterCUG-Restrictions
DEFINED in MAP-MS-DataTypes : 440

InterCUG-Restrictions.................type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 444
USED in MAP-MS-DataTypes : 51 440

internationalECT-Barred.................identifier of Named Number, 11
DEFINED in MAP-MS-DataTypes : 264

internationalOGCallsBarred..............identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 254

internationalOGCallsNotToHPLMN-CountryBanidentifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 255

interrogateSS...........................value reference InterrogateSS, CHOICE VALUE
DEFINED in MAP-Protocol : 188

InterrogateSS...........................type reference OPERATION
DEFINED in MAP-SupplementaryServi : 147
USED in MAP-Protocol : 56 188
USED in MAP-SupplementaryServi : 17

interrogateSS-Res.......................identifier of InterrogateSS-Res
DEFINED in MAP-SupplementaryServi : 151

InterrogateSS-Res.......................type reference CHOICE
DEFINED in MAP-SS-DataTypes : 165
USED in MAP-SupplementaryServi : 56 151
USED in MAP-SS-DataTypes : 19

interrogationType.......................identifier of [3] InterrogationType
DEFINED in MAP-CH-DataTypes : 72

InterrogationType.......................type reference ENUMERATED
DEFINED in MAP-CH-DataTypes : 87
USED in MAP-CH-DataTypes : 72

interzonalECT-Barred....................identifier of Named Number, 12
DEFINED in MAP-MS-DataTypes : 265

interzonalOGCallsAndInternationalOGCallsidentifier of Named Number, 8
DEFINED in MAP-MS-DataTypes : 258

interzonalOGCallsBarred...................identifier of Named Number, 6
DEFINED in MAP-MS-DataTypes : 256

interzonalOGCallsNotToHPLMN-CountryBarreidentifier of Named Number, 7
DEFINED in MAP-MS-DataTypes : 257

intraCUG-Options........................identifier of IntraCUG-Options
DEFINED in MAP-MS-DataTypes : 412

IntraCUG-Options.......................type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 422
USED in MAP-MS-DataTypes : 52 412

invalidFormat...........................identifier of Named Number, 1
DEFINED in MAP-ER-DataTypes : 115

invalidSME-Address......................identifier of Named Number, 5
DEFINED in MAP-ER-DataTypes : 125

invoke..................................identifier of [1] IMPLICIT Invoke
DEFINED in TCAPMessages : 125

Invoke..................................type reference SEQUENCE
invokeID................................identifier of InvokeIdType
DEFINEd in TCAPMessages : 133
USED in TCAPMessages : 125

invokeID................................identifier of InvokeIdType
DEFINEd in TCAPMessages : 134

invokeID................................identifier of InvokeIdType
DEFINEd in TCAPMessages : 145

invokeID................................identifier of InvokeIdType
DEFINEd in TCAPMessages : 157

InvokeIdType............................type reference INTEGER
DEFINEd in TCAPMessages : 166

invokeProblem...........................identifier of [1] IMPLICIT InvokeProblem
DEFINEd in TCAPMessages : 171

InvokeProblem...........................type reference INTEGER
DEFINEd in TCAPMessages : 183
USED in TCAPMessages : 171

ISDN-AddressString......................type reference AddressString
DEFINEd in MAP-CommonDataTypes : 116
USED in MAP-OperationAndMainte : 42 79
USED in MAP-MS-DataTypes : 96 123 123 129 139 177 183 211 322 532 562 573 634 679 690
USED in MAP-CommonDataTypes : 17
USED in MAP-Ch-DataTypes : 43 69 75 113 118 122 130 131 136 143 195
USED in MAP-SS-DataTypes : 38 79
USED in MAP-SM-DataTypes : 31 52 65 102 107 121 127 132

ISDN-SubaddressString.....................type reference OCTET STRING
DEFINEd in MAP-CommonDataTypes : 122
USED in MAP-CommonDataTypes : 159
USED in MAP-MS-DataTypes : 159
USED in MAP-Ch-DataTypes : 18
USED in MAP-SS-DataTypes : 39 56 80

kc......................................identifier of Kc
DEFINEd in MAP-MS-DataTypes : 159

Kc......................................type reference OCTET STRING
DEFINEd in MAP-MS-DataTypes : 166
USED in MAP-MS-DataTypes : 159

laiFixedLength..........................identifier of [1] LAIFixedLength
DEFINEd in MAP-CommonDataTypes : 242

LAIFixedLength..........................type reference OCTET STRING
DEFINEd in MAP-CommonDataTypes : 254
USED in MAP-CommonDataTypes : 242

linkedID................................identifier of [0] IMPLICIT InvokeIdType
DEFINEd in TCAPMessages : 135

linkedResponseUnexpected................identifier of Named Number, 6
DEFINEd in TCAPMessages : 189

lmsi....................................identifier of [10] LMSI
DEFINEd in MAP-MS-DataTypes : 124

lmsi....................................identifier of LMSI
DEFINEd in MAP-MS-DataTypes : 144

lmsi....................................identifier of LMSI
DEFINEd in MAP-MS-DataTypes : 568

lmsi....................................identifier of [1] LMSI
DEFINEd in MAP-MS-DataTypes : 609

LMSI....................................type reference OCTET STRING
DEFINEd in MAP-CommonDataTypes : 216
USED in MAP-MS-DataTypes : 101 124 144 568 609
USED in MAP-CommonDataTypes : 29
USED in MAP-CH-DataTypes : 47 132
USED in MAP-SM-DataTypes : 34 66 97

lmsi..............................identifier of [4] LMSI
DEFINED in MAP-CH-DataTypes : 132
lmsi..............................identifier of LMSI
DEFINED in MAP-SM-DataTypes : 66
lmsi..............................identifier of [1] LMSI
DEFINED in MAP-SM-DataTypes : 97

locationInformation.....................identifier of [0] LocationInformation
DEFINED in MAP-MS-DataTypes : 620

LocationInformation.....................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 631
USED in MAP-MS-DataTypes : 63 620

LocationInfoWithLMSI.....................type reference SEQUENCE
DEFINED in MAP-SM-DataTypes : 64
USED in MAP-SM-DataTypes : 60

LocationInfoWithLMSI.....................type reference SEQUENCE
DEFINED in MAP-SM-DataTypes : 64
USED in MAP-SM-DataTypes : 60

locationNumber..........................identifier of [2] LocationNumber
DEFINED in MAP-MS-DataTypes : 635

LocationNumber..........................type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 659
USED in MAP-MS-DataTypes : 635

mah.................................value reference SS-Code, '00110010'B
DEFINED in MAP-SS-Code : 59

MAP-BS-Code.........................module reference
DEFINED in MAP-BS-Code : 1
USED in MAP-MS-DataTypes : 86
USED in MAP-CommonDataTypes : 50

MAP-CallHandlingOperations..............module reference
DEFINED in MAP-CallHandlingOperat : 1
USED in MAP-Protocol : 47

MAP-CH-DataTypes.......................module reference
DEFINED in MAP-CH-DataTypes : 1
USED in MAP-CallHandlingOperat : 49

MAP-CommonDataTypes....................module reference
DEFINED in MAP-CommonDataTypes : 1
USED in MAP-MobileServiceOpera : 99
USED in MAP-OperationAndMainte : 44
USED in MAP-MS-DataTypes : 106
USED in MAP-OM-DataTypes : 23
USED in MAP-CH-DataTypes : 49
USED in MAP-SS-DataTypes : 41
USED in MAP-SM-DataTypes : 35
USED in MAP-ER-DataTypes : 56

MAP-Errors............................module reference
DEFINED in MAP-Errors : 1
USED in MAP-Protocol : 120
USED in MAP-MobileServiceOpera : 65
USED in MAP-OperationAndMainte : 30
USED in MAP-CallHandlingOperat : 40
USED in MAP-SupplementaryServ : 49
USED in MAP-ShortMessageServic : 42

MAP-ER-DataTypes......................module reference
DEFINED in MAP-ER-DataTypes : 1
USED in MAP-Errors : 113
USED in MAP-SM-DataTypes : 40
MAP-EXTENSION.......................information object class reference CLASS
  DEFINED in MAP-ExtensionDataTypes :  22
  USED in MAP-ExtensionDataTypes :  41 43 48

MAP-ExtensionDataTypes....................module reference
  DEFINED in MAP-ExtensionDataTypes :  1
  USED in MAP-MS-DataTypes :  111
  USED in MAP-CommonDataTypes :  55
  USED in MAP-OM-DataTypes :  28
  USED in MAP-CH-DataTypes :  54
  USED in MAP-SM-DataTypes :  45
  USED in MAP-ER-DataTypes :  66

MAP-MobileServiceOperations............module reference
  DEFINED in MAP-MobileServiceOpera :  1
  USED in MAP-Protocol :  31

MAP-MS-DataTypes.......................module reference
  DEFINED in MAP-MS-DataTypes :  1
  USED in MAP-MobileServiceOpera :  92
  USED in MAP-CH-DataTypes :  33

MAP-OM-DataTypes.......................module reference
  DEFINED in MAP-OM-DataTypes :  1
  USED in MAP-OperationAndMainte :  38

MAP-OperationAndMaintenanceOperations...module reference
  DEFINED in MAP-OperationAndMainte :  1
  USED in MAP-Protocol :  39

MAP-Protocol............................module reference
  DEFINED in MAP-Protocol :  1
  USED in MAP-ShortMessageServic :  74

MAP-ShortMessageServiceOperations.......module reference
  DEFINED in MAP-ShortMessageServic :  1
  USED in MAP-Protocol :  62

MAP-SM-DataTypes.......................module reference
  DEFINED in MAP-SM-DataTypes :  1
  USED in MAP-ShortMessageServic :  57

MAP-SS-Code............................module reference
  DEFINED in MAP-SS-Code :  1
  USED in MAP-SupplementaryServi :  61
  USED in MAP-MS-DataTypes :  81
  USED in MAP-SS-DataTypes :  46
  USED in MAP-ER-DataTypes :  61

MAP-SS-DataTypes.......................module reference
  DEFINED in MAP-SS-DataTypes :  1
  USED in MAP-SupplementaryServi :  61
  USED in MAP-Errors :  79
  USED in MAP-MS-DataTypes :  76
  USED in MAP-CH-DataTypes :  39
  USED in MAP-ER-DataTypes :  49

MAP-SupplementaryServiceOperations......module reference
  DEFINED in MAP-SupplementaryServi :  1
  USED in MAP-Protocol :  62

MAP-TS-Code............................module reference
  DEFINED in MAP-TS-Code :  1
  USED in MAP-MS-DataTypes :  91
  USED in MAP-CommonDataTypes :  44

maxAddressLength........................value reference INTEGER, 20
  DEFINED in MAP-CommonDataTypes :  114
  USED in MAP-CommonDataTypes :  73

maxISDN-AddressLength.....................value reference INTEGER, 9
  DEFINED in MAP-CommonDataTypes :  120
  USED in MAP-CommonDataTypes :  117

maxISDN-SubaddressLength...............value reference INTEGER, 21
maxNumOfBasicServiceGroups..............value reference INTEGER, 13
maxNumOfBasicServices...................value reference INTEGER, 70
maxNumOfBearerServices..................value reference INTEGER, 50
maxNumOfCamel1TDPPData..................value reference INTEGER, 10
maxNumOfCUG.............................value reference INTEGER, 10
maxNumOfExt-BasicServiceGroups..........value reference INTEGER, 32
maxNumOfHLR-Id.........................value reference INTEGER, 50
maxNumOfPrivateExtensions...............value reference INTEGER, 10
maxNumOfSS..............................value reference INTEGER, 30
maxNumOfTeleservices....................value reference INTEGER, 20
maxNumOfVBSGroupIds.....................value reference INTEGER, 50
maxNumOfVGCSGroupIds....................value reference INTEGER, 50
maxNumOfZoneCodes.......................value reference INTEGER, 10
maxSignalInfoLength.....................value reference INTEGER, 200
mcef-Set................................identifier of Named Number, 2
mci.....................................value reference SS-Code, '00010101'B
memoryAvailable.........................identifier of Named Number, 1
memoryCapacityExceeded..................identifier of Named Number, 0
memoryCapacityExceeded..................identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes : 120

MessageType..............................type reference CHOICE
DEFINED in TCAPMessages : 51
USED in TCAPMessages : 47

messageWaitingListFull.................value reference MessageWaitingListFull, CHOICE VALUE
DEFINED in MAP-Protocol : 292

MessageWaitingListFull.................type reference ERROR
DEFINED in MAP-Errors : 313
USED in MAP-Protocol : 118 292
USED in MAP-ShortMessageServic : 40 121
USED in MAP-Errors : 69

messageWaitListFullParam................identifier of MessageWaitListFullParam
DEFINED in MAP-Errors : 315

MessageWaitListFullParam..............type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 234
USED in MAP-Errors : 110 315
USED in MAP-ER-DataTypes : 41

mistypedComponent......................identifier of Named Number, 1
DEFINED in TCAPMessages : 180

mistypedParameter......................identifier of Named Number, 2
DEFINED in TCAPMessages : 185

mistypedParameter......................identifier of Named Number, 2
DEFINED in TCAPMessages : 194

mistypedParameter......................identifier of Named Number, 4
DEFINED in TCAPMessages : 200

mnrf-Set.................................identifier of Named Number, 1
DEFINED in MAP-SM-DataTypes : 139

moreMessagesToSend....................identifier of NULL
DEFINED in MAP-SM-DataTypes : 86

mo-forwardSM................................value reference MO-ForwardSM, CHOICE VALUE
DEFINED in MAP-Protocol : 200

MO-ForwardSM.............................type reference OPERATION
DEFINED in MAP-ShortMessageServic : 81
USED in MAP-Protocol : 68 200
USED in MAP-ShortMessageServic : 14

mo-forwardSM-Arg........................identifier of MO-ForwardSM-Arg
DEFINED in MAP-ShortMessageServic : 83

MO-ForwardSM-Arg........................type reference SEQUENCE
DEFINED in MAP-SM-DataTypes : 70
USED in MAP-ShortMessageServic : 48 83
USED in MAP-SM-DataTypes : 16

mo-forwardSM-Res........................identifier of MO-ForwardSM-Res
DEFINED in MAP-ShortMessageServic : 85

MO-ForwardSM-Res........................type reference SEQUENCE
DEFINED in MAP-SM-DataTypes : 77
USED in MAP-ShortMessageServic : 49 85
USED in MAP-SM-DataTypes : 17

msc-AreaRestricted......................identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 484

msc-Number..............................identifier of [1] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 123

msc-Number..............................identifier of [1] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 130

msc-Number..............................identifier of [1] ISDN-AddressString
DEFINED in MAP-SM-DataTypes : 65

msisdn...................................identifier of ISDN-AddressString
msisdn............................identifier of [1] ISDN-AddressString
DEFINED in MAP-MS-DataTypes      : 211
msisdn............................identifier of [1] ISDN-AddressString
DEFINED in MAP-CH-DataTypes       : 690
msisdn............................identifier of [0] ISDN-AddressString
DEFINED in MAP-SM-DataTypes       : 69
msisdn............................identifier of [2] ISDN-AddressString
DEFINED in MAP-SM-DataTypes       : 131
msisdn............................identifier of [0] ISDN-AddressString
DEFINED in MAP-SM-DataTypes       : 52
msisdn............................identifier of [2] ISDN-AddressString
DEFINED in MAP-SM-DataTypes       : 102
msisdn............................identifier of ISDN-AddressString
DEFINED in MAP-SM-DataTypes       : 107
msisdn............................identifier of ISDN-AddressString
DEFINED in MAP-SM-DataTypes       : 127
msNotReachable.....................identifier of NULL
DEFINED in MAP-MS-DataTypes       : 574
msPurged...........................identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes       : 669
ms-Present..........................identifier of Named Number, 0
DEFINED in MAP-SM-DataTypes       : 150
mt-forwardSM.......................value reference MT-ForwardSM, CHOICE VALUE
DEFINED in MAP-Protocol           : 201
MT-ForwardSM........................type reference OPERATION
DEFINED in MAP-ShortMessageServic : 93
USED in MAP-Protocol             : 69 201
USED in MAP-ShortMessageServic   : 15
mt-forwardSM-Arg....................identifier of MT-ForwardSM-Arg
DEFINED in MAP-ShortMessageServic : 95
MT-ForwardSM-Arg....................type reference SEQUENCE
DEFINED in MAP-SM-DataTypes       : 82
USED in MAP-ShortMessageServic    : 50 95
USED in MAP-SM-DataTypes          : 18
mt-forwardSM-Res....................identifier of MT-ForwardSM-Res
DEFINED in MAP-ShortMessageServic : 97
MT-ForwardSM-Res....................type reference SEQUENCE
DEFINED in MAP-SM-DataTypes       : 90
USED in MAP-ShortMessageServic    : 51 97
USED in MAP-SM-DataTypes          : 19
multipleECT-Barred...................identifier of Named Number, 14
DEFINED in MAP-MS-DataTypes       : 267
multiPTY............................value reference SS-Code, '01010001'B
DEFINED in MAP-SS-Code            : 77
mw-Status..........................identifier of MW-Status
DEFINED in MAP-SM-DataTypes       : 133
MW-Status..........................type reference BIT STRING
DEFINED in MAP-SM-DataTypes       : 137
USED in MAP-SM-DataTypes          : 133
negativePW-Check...................value reference NegativePW-Check, CHOICE VALUE
DEFINED in MAP-Protocol           : 283
NegativePW-Check...................type reference ERROR
DEFINED in MAP-Errors             : 297
USED in MAP-Protocol              : 114 283
USED in MAP-SupplementaryServi : 42 125 144 216
USED in MAP-Errors : 63

netDetNotReachable.........................identifier of NotReachableReason
DEFINEd in MAP-MS-DataTypes : 665

NetworkResource.........................type reference ENUMERATED
DEFINEd in MAP-CommonDataTypes : 228
USED in MAP-CommonDataTypes : 31
USED in MAP-ER-DataTypes : 55 143 150

networkResource.........................identifier of NetworkResource
DEFINEd in MAP-ER-DataTypes : 143

networkResource.........................identifier of NetworkResource
DEFINEd in MAP-ER-DataTypes : 150

networkSignalInfo.......................identifier of [10] ExternalSignalInfo
DEFINEd in MAP-CH-DataTypes : 79

DEFINEd in MAP-CH-DataTypes : 134

ewPassword.........................identifier of Password
DEFINEd in MAP-SupplementaryServi : 208

newPasswordsMismatch.....................identifier of Named Number, 2
DEFINEd in MAP-ER-DataTypes : 116

noCUG-Restrictions.......................identifier of Named Number, 0
DEFINEd in MAP-MS-DataTypes : 423

noHandoverNumberAvailable...............value reference NoHandoverNumberAvailable, CHOICE

VALUE
DEFINEd in MAP-Protocol : 245

NoHandoverNumberAvailable...............type reference ERROR
DEFINEd in MAP-Errors : 202
USED in MAP-Protocol : 93 245
USED in MAP-MobileServiceOpera : 63 177
USED in MAP-Errors : 34

noReply.........................identifier of Named Number, 2
DEFINEd in MAP-CH-DataTypes : 98

DEFINEd in MAP-MS-DataTypes : 325

noReplyConditionTime.....................identifier of [5] NoReplyConditionTime
DEFINEd in MAP-SS-DataTypes : 57

NoReplyConditionTime.....................type reference INTEGER
DEFINEd in MAP-SS-DataTypes : 60
USED in MAP-SS-DataTypes : 28 57 82

noReplyConditionTime.....................identifier of [7] NoReplyConditionTime
DEFINEd in MAP-SS-DataTypes : 82

noRoamingNbParam.......................identifier of NoRoamingNbParam
DEFINEd in MAP-Errors : 219

NoRoamingNbParam.......................type reference SEQUENCE
DEFINEd in MAP-ER-DataTypes : 202
USED in MAP-Errors : 99 219
USED in MAP-ER-DataTypes : 33

noRoamingNumberAvailable...............value reference NoRoamingNumberAvailable, CHOICE

VALUE
DEFINEd in MAP-Protocol : 258

NoRoamingNumberAvailable...............type reference ERROR
DEFINEd in MAP-Errors : 217
USED in MAP-Protocol : 97 258
USED in MAP-CallHandlingOperat : 32 89
USED in MAP-Errors : 42

DEFINEd in MAP-SM-DataTypes : 99
   DEFINED in MAP-SM-DataTypes : 104

noSubscriberReply.......................value reference NoSubscriberReply, CHOICE VALUE
   DEFINED in MAP-Protocol : 261

   NoSubscriberReply.......................type reference ERROR
   DEFINED in MAP-Errors : 233
   USED in MAP-Protocol : 100 261
   USED in MAP-CallHandlingOperat : 35 72
   USED in MAP-Errors : 44

noSubscriberReplyParam..................identifier of NoSubscriberReplyParam
   DEFINED in MAP-Errors : 235

   NoSubscriberReplyParam..................type reference SEQUENCE
   DEFINED in MAP-ER-DataTypes : 214
   USED in MAP-Errors : 103 235
   USED in MAP-ER-DataTypes : 36

notProvidedFromVLR.......................identifier of [2] NULL
   DEFINED in MAP-MS-DataTypes : 666

notReachable.............................identifier of Named Number, 0
   DEFINED in MAP-CH-DataTypes : 96

   NotReachableReason......................type reference ENUMERATED
   DEFINED in MAP-MS-DataTypes : 668
   USED in MAP-MS-DataTypes : 665

notRegistered...........................identifier of Named Number, 3
   DEFINED in MAP-MS-DataTypes : 672

not-derivable...........................identifier of NULL
   DEFINED in TCAPMessages : 168

numberChanged...........................value reference NumberChanged, CHOICE VALUE
   DEFINED in MAP-Protocol : 226

   NumberChanged.........................type reference ERROR
   DEFINED in MAP-Errors : 153
   USED in MAP-Protocol : 84 226
   USED in MAP-CallHandlingOperat : 29 67
   USED in MAP-Errors : 21

   numberChangedParam...................identifier of NumberChangedParam
   DEFINED in MAP-Errors : 155

   NumberChangedParam...................type reference SEQUENCE
   DEFINED in MAP-ER-DataTypes : 174
   USED in MAP-Errors : 91 155
   USED in MAP-ER-DataTypes : 26

   NumberOfForwarding......................type reference INTEGER
   DEFINED in MAP-CH-DataTypes : 66
   USED in MAP-CH-DataTypes : 20 71

   numberOfForwarding......................identifier of [2] NumberOfForwarding
   DEFINED in MAP-CH-DataTypes : 71

   numberOfPW-AttemptsViolation.........value reference NumberOfPW-AttemptsViolation, CHOICE VALUE
   DEFINED in MAP-Protocol : 284

   NumberOfPW-AttemptsViolation.........type reference ERROR
   DEFINED in MAP-Errors : 299
   USED in MAP-Protocol : 115 284
   USED in MAP-SupplementaryServi : 43 126 145 217
   USED in MAP-Errors : 64

odbc-Data................................identifier of [8] ODB-Data
   DEFINED in MAP-MS-DataTypes : 221

   ODB-Data................................type reference SEQUENCE
   DEFINED in MAP-MS-DataTypes : 246
   USED in MAP-MS-DataTypes : 40 221
**odb-GeneralData**..........................identifier of ODB-GeneralData
Defined in MAP-MS-DataTypes: 247

**ODB-GeneralData**..........................typedef reference BIT STRING
Defined in MAP-MS-DataTypes: 252
Used in MAP-MS-DataTypes: 247 476

**odb-GeneralData**..........................identifier of [4] ODB-GeneralData
Defined in MAP-MS-DataTypes: 476

**odb-HPLMN-Data**..........................identifier of ODB-HPLMN-Data
Defined in MAP-MS-DataTypes: 248

**ODB-HPLMN-Data**..........................typedef reference BIT STRING
Defined in MAP-MS-DataTypes: 271
Used in MAP-MS-DataTypes: 248

**omc-Id**..................................identifier of [3] AddressString
Defined in MAP-OM-DataTypes: 40

**operationCode**............................identifier of OPERATION
Defined in TCAPMessages: 136
Used in TCAPMessages: 137

**operationCode**............................identifier of OPERATION
Defined in TCAPMessages: 147
Used in TCAPMessages: 148

**operatorBarring**..........................identifier of Named Number, 1
Defined in MAP-ER-DataTypes: 89

**operatorDeterminedBarring**............identifier of Named Number, 1
Defined in MAP-MS-DataTypes: 234

**operatorDeterminedBarring**............identifier of Named Number, 3
Defined in MAP-ER-DataTypes: 78

**OrigTransactionID**..........................typedef reference [APPLICATION 8] IMPLICIT TransactionID
Defined in TCAPMessages: 97
Used in TCAPMessages: 61 69

**or-Capability**..........................identifier of [5] OR-Phase
Defined in MAP-CH-DataTypes: 74

**or-Interrogation**.........................identifier of [4] NULL
Defined in MAP-CH-DataTypes: 73

**or-Interrogation**.........................identifier of [10] NULL
Defined in MAP-CH-DataTypes: 138

**or-NotAllowed**..........................value reference OR-NotAllowed, CHOICE VALUE
Defined in MAP-Protocol: 264

**OR-NotAllowed**..........................typedef reference ERROR
Defined in MAP-Errors: 258
Used in MAP-Protocol: 96 264
Used in MAP-CallHandlingOperat: 27 65 87 98
Used in MAP-Errors: 41

**or-NotAllowedParam**......................identifier of OR-NotAllowedParam
Defined in MAP-Errors: 260

**OR-NotAllowedParam**......................typedef reference SEQUENCE
Defined in MAP-ER-DataTypes: 166
Used in MAP-Errors: 100 260
Used in MAP-ER-DataTypes: 24

**OR-Phase**.................................typedef reference INTEGER
Defined in MAP-CH-DataTypes: 91
Used in MAP-CH-DataTypes: 74

**otid**....................................identifier of OrigTransactionID
Defined in TCAPMessages: 61

**otid**....................................identifier of OrigTransactionID
Defined in TCAPMessages: 69
overrideCategory.........................identifier of [1] OverrideCategory
DEFINED in MAP-SS-DataTypes : 144

OverrideCategory.........................type reference ENUMERATED
DEFINED in MAP-SS-DataTypes : 151
USED in MAP-SS-DataTypes : 26 144

overrideDisabled.........................identifier of Named Number, 1
DEFINED in MAP-SS-DataTypes : 153

overrideEnabled.........................identifier of Named Number, 0
DEFINED in MAP-SS-DataTypes : 152

O-BcsmCamelTDPData.......................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 529
USED in MAP-MS-DataTypes : 525

O-BcsmCamelTDPDataList...................identifier of O-BcsmCamelTDPDataList
DEFINED in MAP-MS-DataTypes : 520

O-BcsmCamelTDPDataList...................type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 524
USED in MAP-MS-DataTypes : 520

O-BcsmTriggerDetectionPoint..............identifier of O-BcsmTriggerDetectionPoint
DEFINED in MAP-MS-DataTypes : 530
USED in MAP-MS-DataTypes : 530

O-CSI....................................identifier of [0] O-CSI
DEFINED in MAP-MS-DataTypes : 515

O-CSI....................................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 519
USED in MAP-MS-DataTypes : 44 515
USED in MAP-CH-DataTypes : 32 154 180

DEFINED in MAP-CH-DataTypes : 154

O-CSI....................................identifier of [1] O-CSI
DEFINED in MAP-CH-DataTypes : 180

padAccessCA-1200bps....................value reference BearerServiceCode, '00100010'B
DEFINED in MAP-BS-Code : 69

padAccessCA-1200-75bps..................value reference BearerServiceCode, '00100011'B
DEFINED in MAP-BS-Code : 70

padAccessCA-2400bps....................value reference BearerServiceCode, '00100100'B
DEFINED in MAP-BS-Code : 71

padAccessCA-300bps......................value reference BearerServiceCode, '00100001'B
DEFINED in MAP-BS-Code : 68

padAccessCA-4800bps....................value reference BearerServiceCode, '00100101'B
DEFINED in MAP-BS-Code : 72

padAccessCA-9600bps....................value reference BearerServiceCode, '00100110'B
DEFINED in MAP-BS-Code : 73

parameter...............................identifier of ANY DEFINED BY operationCode
DEFINED in TCAPMessages : 137

parameter...............................identifier of ANY DEFINED BY operationCode
DEFINED in TCAPMessages : 148

Password................................type reference NumericString
DEFINED in MAP-SS-DataTypes : 192
USED in MAP-SupplementaryService : 59 208 225
USED in MAP-SS-DataTypes : 22

pcs-Extensions..........................identifier of [1] PCS-Extensions
DEFINED in MAP-ExtensionDataTypes: 34

PCS-Extensions.................type reference SEQUENCE
DEFINED in MAP-ExtensionDataTypes: 56
USED in MAP-ExtensionDataTypes: 34

permanent..........................identifier of Named Number, 0
DEFINED in MAP-SS-DataTypes: 147

phasel.............................identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes: 556

plmn...............................identifier of Named Number, 0
DEFINED in MAP-CommonDataTypes: 229

plmnRoamingNotAllowed..............identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes: 77

plmn-SpecificBarringType1..........identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes: 272

plmn-SpecificBarringType2..........identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes: 273

plmn-SpecificBarringType3..........identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes: 274

plmn-SpecificBarringType4..........identifier of Named Number, 3
DEFINED in MAP-MS-DataTypes: 275

plmn-specificBS-1...............value reference BearerServiceCode, '11010001'B
DEFINED in MAP-BS-Code: 111

plmn-specificBS-2...............value reference BearerServiceCode, '11010010'B
DEFINED in MAP-BS-Code: 112

plmn-specificBS-3...............value reference BearerServiceCode, '11010011'B
DEFINED in MAP-BS-Code: 113

plmn-specificBS-4...............value reference BearerServiceCode, '11010100'B
DEFINED in MAP-BS-Code: 114

plmn-specificBS-5...............value reference BearerServiceCode, '11010101'B
DEFINED in MAP-BS-Code: 115

plmn-specificBS-6...............value reference BearerServiceCode, '11010110'B
DEFINED in MAP-BS-Code: 116

plmn-specificBS-7...............value reference BearerServiceCode, '11010111'B
DEFINED in MAP-BS-Code: 117

plmn-specificBS-8...............value reference BearerServiceCode, '11011000'B
DEFINED in MAP-BS-Code: 118

plmn-specificBS-9...............value reference BearerServiceCode, '11011001'B
DEFINED in MAP-BS-Code: 119

plmn-specificBS-A...............value reference BearerServiceCode, '11011010'B
DEFINED in MAP-BS-Code: 120

plmn-specificBS-B...............value reference BearerServiceCode, '11011011'B
DEFINED in MAP-BS-Code: 121

plmn-specificBS-C...............value reference BearerServiceCode, '11011100'B
DEFINED in MAP-BS-Code: 122

plmn-specificBS-D...............value reference BearerServiceCode, '11011101'B
DEFINED in MAP-BS-Code: 123

plmn-specificBS-E...............value reference BearerServiceCode, '11011110'B
DEFINED in MAP-BS-Code: 124

plmn-specificBS-F...............value reference BearerServiceCode, '11011111'B
DEFINED in MAP-BS-Code: 125

plmn-specificSS-1...............value reference SS-Code, '11110001'B
DEFINED in MAP-SS-Code: 121
<table>
<thead>
<tr>
<th>plmn-specificSS-2</th>
<th>value reference SS-Code, '11110010'B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>122</td>
</tr>
<tr>
<td>plmn-specificSS-3</td>
<td>value reference SS-Code, '11110011'B</td>
</tr>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>123</td>
</tr>
<tr>
<td>plmn-specificSS-4</td>
<td>value reference SS-Code, '11110100'B</td>
</tr>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>124</td>
</tr>
<tr>
<td>plmn-specificSS-5</td>
<td>value reference SS-Code, '11110101'B</td>
</tr>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>125</td>
</tr>
<tr>
<td>plmn-specificSS-6</td>
<td>value reference SS-Code, '11110110'B</td>
</tr>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>126</td>
</tr>
<tr>
<td>plmn-specificSS-7</td>
<td>value reference SS-Code, '11110111'B</td>
</tr>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>127</td>
</tr>
<tr>
<td>plmn-specificSS-8</td>
<td>value reference SS-Code, '11111000'B</td>
</tr>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>128</td>
</tr>
<tr>
<td>plmn-specificSS-9</td>
<td>value reference SS-Code, '11111001'B</td>
</tr>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>129</td>
</tr>
<tr>
<td>plmn-specificSS-A</td>
<td>value reference SS-Code, '11111010'B</td>
</tr>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>130</td>
</tr>
<tr>
<td>plmn-specificSS-B</td>
<td>value reference SS-Code, '11111011'B</td>
</tr>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>131</td>
</tr>
<tr>
<td>plmn-specificSS-C</td>
<td>value reference SS-Code, '11111100'B</td>
</tr>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>132</td>
</tr>
<tr>
<td>plmn-specificSS-D</td>
<td>value reference SS-Code, '11111101'B</td>
</tr>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>133</td>
</tr>
<tr>
<td>plmn-specificSS-E</td>
<td>value reference SS-Code, '11111110'B</td>
</tr>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>134</td>
</tr>
<tr>
<td>plmn-specificSS-F</td>
<td>value reference SS-Code, '11111111'B</td>
</tr>
<tr>
<td>Defined in MAP-SS-Code</td>
<td>135</td>
</tr>
<tr>
<td>plmn-specificTS-1</td>
<td>value reference TeleserviceCode, '11010001'B</td>
</tr>
<tr>
<td>Defined in MAP-TS-Code</td>
<td>73</td>
</tr>
<tr>
<td>plmn-specificTS-2</td>
<td>value reference TeleserviceCode, '11010010'B</td>
</tr>
<tr>
<td>Defined in MAP-TS-Code</td>
<td>74</td>
</tr>
<tr>
<td>plmn-specificTS-3</td>
<td>value reference TeleserviceCode, '11010011'B</td>
</tr>
<tr>
<td>Defined in MAP-TS-Code</td>
<td>75</td>
</tr>
<tr>
<td>plmn-specificTS-4</td>
<td>value reference TeleserviceCode, '11010100'B</td>
</tr>
<tr>
<td>Defined in MAP-TS-Code</td>
<td>76</td>
</tr>
<tr>
<td>plmn-specificTS-5</td>
<td>value reference TeleserviceCode, '11010101'B</td>
</tr>
<tr>
<td>Defined in MAP-TS-Code</td>
<td>77</td>
</tr>
<tr>
<td>plmn-specificTS-6</td>
<td>value reference TeleserviceCode, '11010110'B</td>
</tr>
<tr>
<td>Defined in MAP-TS-Code</td>
<td>78</td>
</tr>
<tr>
<td>plmn-specificTS-7</td>
<td>value reference TeleserviceCode, '11010111'B</td>
</tr>
<tr>
<td>Defined in MAP-TS-Code</td>
<td>79</td>
</tr>
<tr>
<td>plmn-specificTS-8</td>
<td>value reference TeleserviceCode, '11011000'B</td>
</tr>
<tr>
<td>Defined in MAP-TS-Code</td>
<td>80</td>
</tr>
<tr>
<td>plmn-specificTS-9</td>
<td>value reference TeleserviceCode, '11011001'B</td>
</tr>
<tr>
<td>Defined in MAP-TS-Code</td>
<td>81</td>
</tr>
<tr>
<td>plmn-specificTS-A</td>
<td>value reference TeleserviceCode, '11011010'B</td>
</tr>
<tr>
<td>Defined in MAP-TS-Code</td>
<td>82</td>
</tr>
<tr>
<td>plmn-specificTS-B</td>
<td>value reference TeleserviceCode, '11011011'B</td>
</tr>
<tr>
<td>Defined in MAP-TS-Code</td>
<td>83</td>
</tr>
<tr>
<td>plmn-specificTS-C</td>
<td>value reference TeleserviceCode, '11011100'B</td>
</tr>
<tr>
<td>Defined in MAP-TS-Code</td>
<td>84</td>
</tr>
</tbody>
</table>
plmn-specificTS-D..................value reference TeleserviceCode, '1101101' B
 DEFINED in MAP-TS-Code : 85
plmn-specificTS-E..................value reference TeleserviceCode, '1101110' B
 DEFINED in MAP-TS-Code : 86
plmn-specificTS-F..................value reference TeleserviceCode, '1101111' B
 DEFINED in MAP-TS-Code : 87
preferentialCUG-Indicator.............identifier of CUG-Index
 DEFINED in MAP-MS-DataTypes : 439
premiumRateEntertainmentOGCallsBarred..identifier of Named Number, 4
 DEFINED in MAP-MS-DataTypes : 260
premiumRateInformationOGCallsBarred.....identifier of Named Number, 3
 DEFINED in MAP-MS-DataTypes : 259
prepareHandover..................value reference PrepareHandover, CHOICE VALUE
 DEFINED in MAP-Protocol : 136
PrepareHandover..................type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 168
 USED in MAP-Protocol : 16
 USED in MAP-MobileServiceOpera : 27
prepareHO-Arg..................identifier of PrepareHO-Arg
 DEFINED in MAP-MobileServiceOpera : 170
PrepareHO-Arg..................type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 170
 USED in MAP-MobileServiceOpera : 74 170
 USED in MAP-MS-DataTypes : 23
prepareHO-Res..................identifier of PrepareHO-Res
 DEFINED in MAP-MobileServiceOpera : 172
PrepareHO-Res..................type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 176
 USED in MAP-MobileServiceOpera : 75 172
 USED in MAP-MS-DataTypes : 24
prepareSubsequentHandover...........value reference PrepareSubsequentHandover, CHOICE VALUE
 DEFINED in MAP-Protocol : 140
PrepareSubsequentHandover...........type reference OPERATION
 DEFINED in MAP-MobileServiceOpera : 192
 USED in MAP-Protocol : 20 140
 USED in MAP-MobileServiceOpera : 31
prepareSubsequentHO-Arg..................identifier of PrepareSubsequentHO-Arg
 DEFINED in MAP-MobileServiceOpera : 194
PrepareSubsequentHO-Arg..................type reference SEQUENCE
 DEFINED in MAP-MS-DataTypes : 181
 USED in MAP-MobileServiceOpera : 76 194
 USED in MAP-MS-DataTypes : 25
priorityLevel0..................value reference EMLPP-Priority, 0
 DEFINED in MAP-MS-DataTypes : 303
priorityLevel1..................value reference EMLPP-Priority, 1
 DEFINED in MAP-MS-DataTypes : 304
priorityLevel2..................value reference EMLPP-Priority, 2
 DEFINED in MAP-MS-DataTypes : 305
priorityLevel3..................value reference EMLPP-Priority, 3
 DEFINED in MAP-MS-DataTypes : 306
priorityLevel4..................value reference EMLPP-Priority, 4
 DEFINED in MAP-MS-DataTypes : 307
priorityLevel6..................value reference EMLPP-Priority, 6
 DEFINED in MAP-MS-DataTypes : 301
priorityLevelB..................value reference EMLPP-Priority, 5
DEFINED in MAP-MS-DataTypes : 302

PrivateExtension.........................type reference SEQUENCE
DEFINED in MAP-ExtensionDataTypes : 40
USED in MAP-ExtensionDataTypes : 15 38

privateExtensionList....................identifier of [0] PrivateExtensionList
DEFINED in MAP-ExtensionDataTypes : 33

PrivateExtensionList....................type reference SEQUENCE OF
DEFINED in MAP-ExtensionDataTypes : 37
USED in MAP-ExtensionDataTypes : 33

problem.........................identifier of CHOICE
DEFINED in TCAPMessages : 169

processAccessSignalling...............value reference ProcessAccessSignalling, CHOICE VALUE
DEFINED in MAP-Protocol : 138

ProcessAccessSignalling....................type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 184
USED in MAP-Protocol : 18 138
USED in MAP-MobileServiceOpera : 29

processUnstructuredSS-Request.......value reference ProcessUnstructuredSS-Request, CHOICE VALUE
DEFINED in MAP-Protocol : 189

ProcessUnstructuredSS-Request.............type reference OPERATION
DEFINED in MAP-SupplementaryServi : 162
USED in MAP-Protocol : 57 189
USED in MAP-SupplementaryServi : 18

protocolId..............................identifier of ProtocolId
DEFINED in MAP-CommonDataTypes : 163

ProtocolId..............................type reference ENUMERATED
DEFINED in MAP-CommonDataTypes : 181
USED in MAP-CommonDataTypes : 163

provideRoamingNumber...................value reference ProvideRoamingNumber, CHOICE VALUE
DEFINED in MAP-Protocol : 178

ProvideRoamingNumber....................type reference OPERATION
DEFINED in MAP-CallHandlingOperat : 77
USED in MAP-Protocol : 45 178
USED in MAP-CallHandlingOperat : 14

provideRoamingNumberArg................identifier of ProvideRoamingNumberArg
DEFINED in MAP-CallHandlingOperat : 79

ProvideRoamingNumberArg...................type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 128
USED in MAP-CallHandlingOperat : 45 79
USED in MAP-CH-DataTypes : 16

provideRoamingNumberRes................identifier of ProvideRoamingNumberRes
DEFINED in MAP-CallHandlingOperat : 81

ProvideRoamingNumberRes...................type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 142
USED in MAP-CallHandlingOperat : 46 81
USED in MAP-CH-DataTypes : 17

provideSubscriberInfo...................value reference ProvideSubscriberInfo, CHOICE VALUE
DEFINED in MAP-Protocol : 209

ProvideSubscriberInfo....................type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 143
USED in MAP-Protocol : 28 209
USED in MAP-MobileServiceOpera : 21

provideSubscriberInfoArg................identifier of ProvideSubscriberInfoArg
DEFINED in MAP-MobileServiceOpera : 145

ProvideSubscriberInfoArg...................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 607
USED in MAP-MobileServiceOpera : 87 145
provideSubscriberInfoRes................identifier of ProvideSubscriberInfoRes
DEFINEd in MAP-MobileServiceOpera : 147

ProvideSubscriberInfoRes................type reference SEQUENCE
DEFINEd in MAP-MS-DataTypes : 614
USED in MAP-MobileServiceOpera : 88 147
USED in MAP-MS-DataTypes : 61

DEFINEd in MAP-MS-DataTypes : 220

purgeMS................................value reference PurgeMS, CHOICE VALUE
DEFINEd in MAP-Protocol : 130

PurgeMS.................................type reference OPERATION
DEFINEd in MAP-MobileServiceOpera : 127
USED in MAP-Protocol : 14 130
USED in MAP-MobileServiceOpera : 17

purgeMS-Arg............................identifier of PurgeMS-Arg
DEFINEd in MAP-MobileServiceOpera : 129

PurgeMS-Arg............................type reference SEQUENCE
DEFINEd in MAP-MS-DataTypes : 137
USED in MAP-MobileServiceOpera : 72 129
USED in MAP-MS-DataTypes : 19

pvlr....................................identifier of Named Number, 3
DEFINEd in MAP-CommonDataTypes : 232

pw-RegistrationFailure.................value reference PW-RegistrationFailure, CHOICE VALUE
DEFINEd in MAP-Protocol : 282

PW-RegistrationFailure.................type reference ERROR
DEFINEd in MAP-Errors : 293
USED in MAP-Protocol : 113 282
USED in MAP-SupplementaryService : 41 215
USED in MAP-Errors : 62

pw-RegistrationFailureCause............identifier of PW-RegistrationFailureCause
DEFINEd in MAP-Errors : 295

PW-RegistrationFailureCause............type reference ENUMERATED
DEFINEd in MAP-ER-DataTypes : 113
USED in MAP-Errors : 84 295
USED in MAP-ER-DataTypes : 18

p-abortCause............................identifier of P-AbortCause
DEFINEd in TCAPMessages : 76

P-AbortCause............................type reference [APPLICATION 10] IMPLICIT INTEGER
DEFINEd in TCAPMessages : 102
USED in TCAPMessages : 76

rand....................................identifier of RAND
DEFINEd in MAP-MS-DataTypes : 157

RAND..................................type reference OCTET STRING
DEFINEd in MAP-MS-DataTypes : 162
USED in MAP-MS-DataTypes : 157

readyForSM.........................value reference ReadyForSM, CHOICE VALUE
DEFINEd in MAP-Protocol : 205

ReadyForSM.............................type reference OPERATION
DEFINEd in MAP-ShortMessageService : 136
USED in MAP-Protocol : 73 205
USED in MAP-ShortMessageService : 19

readyForSM-Arg........................identifier of ReadyForSM-Arg
DEFINEd in MAP-ShortMessageService : 138

ReadyForSM-Arg........................type reference SEQUENCE
DEFINEd in MAP-SM-DataTypes : 144
USED in MAP-ShortMessageService : 56 138
USED in MAP-SM-DataTypes : 24
reason..................................identifier of CHOICE
  DEFINED in TCAPMessages : 75

regionalSubscNotSupported.............identifier of Named Number, 3
  DEFINED in MAP-MS-DataTypes : 487

regionalSubscriptionData..............identifier of [10] ZoneCodeList
  DEFINED in MAP-MS-DataTypes : 223

regionalSubscriptionIdentifier........identifier of [5] ZoneCode
  DEFINED in MAP-MS-DataTypes : 496

  DEFINED in MAP-MS-DataTypes : 477

RegionalSubscriptionResponse............type reference ENUMERATED
  DEFINED in MAP-MS-DataTypes : 483
  USED in MAP-MS-DataTypes : 478 510

regionalSubscriptionResponse............identifier of [0] RegionalSubscriptionResponse
  DEFINED in MAP-MS-DataTypes : 509

registerPassword.........................value reference RegisterPassword, CHOICE VALUE
  DEFINED in MAP-Protocol : 193

RegisterPassword.........................type reference OPERATION
  DEFINED in MAP-SupplementaryServi : 204
  USED in MAP-Protocol : 60 193
  USED in MAP-SupplementaryServi : 21

registerSS..............................value reference RegisterSS, CHOICE VALUE
  DEFINED in MAP-Protocol : 184

RegisterSS..............................type reference OPERATION
  DEFINED in MAP-SupplementaryServi : 74
  USED in MAP-Protocol : 52 184
  USED in MAP-SupplementaryServi : 13

registerSS-Arg..........................identifier of RegisterSS-Arg
  DEFINED in MAP-SupplementaryServi : 76

RegisterSS-Arg..........................type reference SEQUENCE
  DEFINED in MAP-SM-DataTypes : 106
  USED in MAP-SM-DataTypes : 52 113
  USED in MAP-SM-DataTypes : 20

reject..................................identifier of [4] IMPLICIT Reject
  DEFINED in TCAPMessages : 128

Reject..................................type reference SEQUENCE
  DEFINED in TCAPMessages : 165
  USED in TCAPMessages : 128

releaseCall.............................identifier of Named Number, 1
  DEFINED in MAP-MS-DataTypes : 549

reportSM-DeliveryStatus...............value reference ReportSM-DeliveryStatus, CHOICE VALUE
  DEFINED in MAP-Protocol : 202

ReportSM-DeliveryStatus.................type reference OPERATION
  DEFINED in MAP-ShortMessageServic : 111
  USED in MAP-Protocol : 70 202
  USED in MAP-ShortMessageServic : 16

reportSM-DeliveryStatusArg...............identifier of ReportSM-DeliveryStatusArg
  DEFINED in MAP-ShortMessageServic : 113

ReportSM-DeliveryStatusArg...............type reference SEQUENCE
  DEFINED in MAP-SM-DataTypes : 106
  USED in MAP-SM-DataTypes : 52 113
  USED in MAP-SM-DataTypes : 20

reportSM-DeliveryStatusRes...............identifier of ReportSM-DeliveryStatusRes
  DEFINED in MAP-ShortMessageServic : 115

ReportSM-DeliveryStatusRes...............type reference SEQUENCE
  DEFINED in MAP-SM-DataTypes : 120
USED in MAP-ShortMessageService : 53 115
USED in MAP-SM-DataTypes : 21

requestedBasicServiceViolatesCUG-Constrain_identifier of Named Number, 5
DEFINED in MAP-ER-DataTypes : 104

requestedInfo.....................................identifier of [2] RequestedInfo
DEFINED in MAP-MS-DataTypes : 610

RequestedInfo....................................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 625
USED in MAP-MS-DataTypes : 610 678

requestedInfo.....................................identifier of [1] RequestedInfo
DEFINED in MAP-MS-DataTypes : 678

reset.............................................value reference Reset, CHOICE VALUE
DEFINED in MAP-Protocol : 162

Reset..............................................type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 255
USED in MAP-Protocol : 25 162
USED in MAP-MobileServiceOpera : 44

resetArg..........................................identifier of ResetArg
DEFINED in MAP-MobileServiceOpera : 257

ResetArg............................................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 561
USED in MAP-MobileServiceOpera : 84 257
USED in MAP-MS-DataTypes : 55

resourceLimitation.........................identifier of Named Number, 4
DEFINED in TCAPMessages : 107

resourceLimitation.........................identifier of Named Number, 3
DEFINED in TCAPMessages : 186

restoreData.................................value reference RestoreData, CHOICE VALUE
DEFINED in MAP-Protocol : 165

RestoreData.................................type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 261
USED in MAP-Protocol : 27 165
USED in MAP-MobileServiceOpera : 46

restoreDataArg...............................identifier of RestoreDataArg
DEFINED in MAP-MobileServiceOpera : 263

RestoreDataArg...............................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 566
USED in MAP-MobileServiceOpera : 85 263
USED in MAP-MS-DataTypes : 56

restoreDataRes...............................identifier of RestoreDataRes
DEFINED in MAP-MobileServiceOpera : 265

RestoreDataRes...............................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 572
USED in MAP-MobileServiceOpera : 86 265
USED in MAP-MS-DataTypes : 57

restrictedArea..............................identifier of Named Number, 2
DEFINED in MAP-MS-DataTypes : 671

result-RR.................................identifier of SEQUENCE
DEFINED in TCAPMessages : 146

resumeCallHandling.........................value reference ResumeCallHandling, CHOICE VALUE
DEFINED in MAP-Protocol : 179

ResumeCallHandling.........................type reference OPERATION
DEFINED in MAP-CallHandlingOperat : 91
USED in MAP-Protocol : 46 179
USED in MAP-CallHandlingOperat : 15

resumeCallHandlingArg.....................identifier of ResumeCallHandlingArg
DEFINED in MAP-CallHandlingOperat : 93
ResumeCallHandlingArg...................type reference SEQUENCE
DEFINEd in MAP-CH-DataTypes : 148
USEd in MAP-CallHandlingOperat : 47 93
USEd in MAP-CH-DataTypes : 18

ResumeCallHandlingRes...................identifier of ResumeCallHandlingRes
DEFINEd in MAP-CallHandlingOperat : 95

ResumeCallHandlingRes...................type reference SEQUENCE
DEFINEd in MAP-CH-DataTypes : 158
USEd in MAP-CallHandlingOperat : 48 95
USEd in MAP-CH-DataTypes : 19

returnError...................identifier of [3] IMPLICIT ReturnError
DEFINEd in TCAPMessages : 127

ReturnError...................type reference SEQUENCE
DEFINEd in TCAPMessages : 156
USEd in TCAPMessages : 127

returnErrorProblem......................identifier of [3] IMPLICIT ReturnErrorProblem
DEFINEd in TCAPMessages : 173

ReturnErrorProblem......................type reference INTEGER
DEFINEd in TCAPMessages : 196
USEd in TCAPMessages : 173

returnErrorUnexpected..................identifier of Named Number, 1
DEFINEd in TCAPMessages : 197

ReturnResult............................type reference SEQUENCE
DEFINEd in TCAPMessages : 144
USEd in TCAPMessages : 126 129

returnResultLast.......................identifier of [2] IMPLICIT ReturnResult
DEFINEd in TCAPMessages : 126

returnResultNotLast.....................identifier of [7] IMPLICIT ReturnResult
DEFINEd in TCAPMessages : 129

returnResultProblem.....................identifier of [2] IMPLICIT ReturnResultProblem
DEFINEd in TCAPMessages : 172

ReturnResultProblem.....................type reference INTEGER
DEFINEd in TCAPMessages : 192
USEd in TCAPMessages : 172

returnResultUnexpected..................identifier of Named Number, 1
DEFINEd in TCAPMessages : 193

roamingNotAllowed......................value reference RoamingNotAllowed, CHOICE VALUE
DEFINEd in MAP-Protocol : 234

RoamingNotAllowed......................type reference ERROR
DEFINEd in MAP-Errors : 171
USEd in MAP-Protocol : 80 234
USEd in MAP-MobileServiceOpere : 61 117
USEd in MAP-Errors : 27

roamingNotAllowedCause..................identifier of RoamingNotAllowedCause
DEFINEd in MAP-ER-DataTypes : 72

RoamingNotAllowedCause..................type reference ENUMERATED
DEFINEd in MAP-ER-DataTypes : 76
USEd in MAP-ER-DataTypes : 72

roamingNotAllowedParam..................identifier of RoamingNotAllowedParam
DEFINEd in MAP-Errors : 173

RoamingNotAllowedParam..................type reference SEQUENCE
DEFINEd in MAP-ER-DataTypes : 71
USEd in MAP-Errors : 93 173
USEd in MAP-ER-DataTypes : 14

roamingNumber.........................identifier of ISDN-AddressString
DEFINEd in MAP-CH-DataTypes : 118
roamingNumber.....................identifier of ISDN-AddressString
  DEFINED in MAP-CH-DataTypes : 143
roamingRestrictionDueToUnsupportedFeatureidentifier of [9] NULL
  DEFINED in MAP-MS-DataTypes : 222
roamingRestrictionDueToUnsupportedFeatureidentifier of [4] NULL
  DEFINED in MAP-MS-DataTypes : 495
RoutingInfo.........................type reference CHOICE
  DEFINED in MAP-CH-DataTypes : 117
  USED in MAP-CH-DataTypes : 169
routingInfo........................identifier of RoutingInfo
  DEFINED in MAP-CH-DataTypes : 169
routingInfoForSM-Arg................identifier of RoutingInfoForSM-Arg
  DEFINED in MAP-ShortMessageServic : 68
RoutingInfoForSM-Arg................type reference SEQUENCE
  DEFINED in MAP-ShortMessageServic : 51
  USED in MAP-ShortMessageServic : 46 68
  USED in MAP-MS-DataTypes : 14
routingInfoForSM-Res................identifier of RoutingInfoForSM-Res
  DEFINED in MAP-ShortMessageServic : 70
RoutingInfoForSM-Res................type reference SEQUENCE
  DEFINED in MAP-ShortMessageServic : 58
  USED in MAP-ShortMessageServic : 47 70
  USED in MAP-MS-DataTypes : 15
rss..................................identifier of Named Number, 7
  DEFINED in MAP-CommonDataTypes : 236
sc-AddressNotIncluded...............identifier of Named Number, 0
  DEFINED in MAP-SM-DataTypes : 138
sc-Congestion.........................identifier of Named Number, 4
  DEFINED in MAP-ER-DataTypes : 124
sendAuthenticationInfo...............value reference SendAuthenticationInfo, CHOICE VALUE
  DEFINED in MAP-Protocol : 146
SendAuthenticationInfo...............type reference OPERATION
  DEFINED in MAP-MobileServiceOpera : 205
  USED in MAP-Protocol : 21 146
  USED in MAP-MobileServiceOpera : 34
sendAuthenticationInfoArg...............identifier of SendAuthenticationInfoArg
  DEFINED in MAP-MobileServiceOpera : 207
SendAuthenticationInfoArg...............type reference IMSI
  DEFINED in MAP-MS-DataTypes : 189
  USED in MAP-MobileServiceOpera : 77 207
  USED in MAP-MS-DataTypes : 28
sendAuthenticationInfoRes...............identifier of SendAuthenticationInfoRes
  DEFINED in MAP-MobileServiceOpera : 209
SendAuthenticationInfoRes...............type reference AuthenticationSetList
  DEFINED in MAP-MS-DataTypes : 191
  USED in MAP-MobileServiceOpera : 78 209
  USED in MAP-MS-DataTypes : 29
sendEndSignal.........................value reference SendEndSignal, CHOICE VALUE
  DEFINED in MAP-Protocol : 137
SendEndSignal.........................type reference OPERATION
  DEFINED in MAP-MobileServiceOpera : 179
  USED in MAP-Protocol : 17 137
  USED in MAP-MobileServiceOpera : 28
sendIdentification..................value reference SendIdentification, CHOICE VALUE
  DEFINED in MAP-Protocol : 131
SendIdentification..................type reference OPERATION
  DEFINED in MAP-MobileServiceOpera : 132
sendIdentificationRes............................identifier of SendIdentificationRes
DEFINED in MAP-MobileServiceOpera : 136

SendIdentificationRes............................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 148
USED in MAP-MobileServiceOpera : 73 136
USED in MAP-MS-DataTypes : 20

sendIMSI................................value reference SendIMSI, CHOICE VALUE
DEFINED in MAP-Protocol : 172

SendIMSI................................type reference OPERATION
DEFINED in MAP-OperationAndMainte : 77
USED in MAP-Protocol : 38 172
USED in MAP-OperationAndMainte : 15

sendRoutingInfo.........................value reference SendRoutingInfo, CHOICE VALUE
DEFINED in MAP-Protocol : 177

SendRoutingInfo............................type reference OPERATION
DEFINED in MAP-CallHandlingOperat : 55
USED in MAP-Protocol : 44 177
USED in MAP-CallHandlingOperat : 13

sendRoutingInfoArg......................identifier of SendRoutingInfoArg
DEFINED in MAP-CallHandlingOperat : 57

SendRoutingInfoArg............................type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 68
USED in MAP-CallHandlingOperat : 43 57
USED in MAP-CH-DataTypes : 14

sendRoutingInfoForSM....................value reference SendRoutingInfoForSM, CHOICE VALUE
DEFINED in MAP-Protocol : 199

SendRoutingInfoForSM............................type reference OPERATION
DEFINED in MAP-ShortMessageServic : 66
USED in MAP-Protocol : 67 199
USED in MAP-ShortMessageServic : 13

sendRoutingInfoRes......................identifier of SendRoutingInfoRes
DEFINED in MAP-CallHandlingOperat : 59

DEFINED in MAP-CH-DataTypes : 101
USED in MAP-CallHandlingOperat : 44 59
USED in MAP-CH-DataTypes : 15

serviceCentreAddress....................identifier of [2] AddressString
DEFINED in MAP-SM-DataTypes : 54

serviceCentreAddress....................identifier of AddressString
DEFINED in MAP-SM-DataTypes : 108

serviceCentreAddress....................identifier of AddressString
DEFINED in MAP-SM-DataTypes : 128

serviceCentreAddressDA..................identifier of [4] AddressString
DEFINED in MAP-SM-DataTypes : 98

serviceCentreAddressOA..................identifier of [4] AddressString
DEFINED in MAP-SM-DataTypes : 103

serviceGranted..........................identifier of Named Number, 0
DEFINED in MAP-MS-DataTypes : 233

serviceKey..............................identifier of ServiceKey
DEFINED in MAP-MS-DataTypes : 531

ServiceKey..............................type reference INTEGER
DEFINED in MAP-MS-DataTypes : 537
USED in MAP-MS-DataTypes : 45 531
USED in MAP-CH-DataTypes : 28 194
DEFINED in MAP-CH-DataTypes : 194

shortMessageMO-PP.........................value reference TeleserviceCode, '00100010'B
DEFINED in MAP-TS-Code : 46

shortMessageMT-PP.........................value reference TeleserviceCode, '00100001'B
DEFINED in MAP-TS-Code : 45

signalInfo..............................identifier of SignalInfo
DEFINED in MAP-CommonDataTypes : 164

SignalInfo..............................type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 171
USED in MAP-CommonDataTypes : 20 164
USED in MAP-SM-DataTypes : 32 73 78 85 91
USED in MAP-ER-DataTypes : 53 130

sm-DeliveryFailure.......................value reference SM-DeliveryFailure, CHOICE VALUE
DEFINED in MAP-Protocol : 291

SM-DeliveryFailure.......................type reference ERROR
DEFINED in MAP-Errors : 309
USED in MAP-Protocol : 117 291
USED in MAP-ShortMessageService : 39 91 108
USED in MAP-Errors : 68

sm-DeliveryFailureCause.....................identifier of SM-DeliveryFailureCause
DEFINED in MAP-Errors : 311

SM-DeliveryFailureCause.....................type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 128
USED in MAP-Errors : 85 311
USED in MAP-ER-DataTypes : 19

sm-DeliveryOutcome......................identifier of SM-DeliveryOutcome
DEFINED in MAP-SM-DataTypes : 109

SM-DeliveryOutcome......................type reference ENUMERATED
DEFINED in MAP-SM-DataTypes : 115
USED in MAP-SM-DataTypes : 25 109

SM-EnumeratedDeliveryFailureCause........type reference ENUMERATED
DEFINED in MAP-ER-DataTypes : 119
USED in MAP-ER-DataTypes : 129

sm-EnumeratedDeliveryFailureCause.......identifier of SM-EnumeratedDeliveryFailureCause
DEFINED in MAP-ER-DataTypes : 129

sm-RP-DA................................identifier of SM-RP-DA
DEFINED in MAP-SM-DataTypes : 71

sm-RP-DA................................identifier of SM-RP-DA
DEFINED in MAP-SM-DataTypes : 83

SM-RP-DA................................type reference CHOICE
DEFINED in MAP-SM-DataTypes : 95
USED in MAP-SM-DataTypes : 71 83

sm-RP-OA................................identifier of SM-RP-OA
DEFINED in MAP-SM-DataTypes : 72

sm-RP-OA................................identifier of SM-RP-OA
DEFINED in MAP-SM-DataTypes : 84

SM-RP-OA................................type reference CHOICE
DEFINED in MAP-SM-DataTypes : 101
USED in MAP-SM-DataTypes : 72 84

sm-RP-PRI............................identifier of [1] BOOLEAN
DEFINED in MAP-SM-DataTypes : 53

sm-RP-UI................................identifier of SignalInfo
DEFINED in MAP-SM-DataTypes : 73

sm-RP-UI................................identifier of SignalInfo
DEFINED in MAP-SM-DataTypes : 78

sm-RP-UI................................identifier of SignalInfo
sm-RP-UI................................identifier of SignalInfo
DEFINED in MAP-SM-DataTypes : 91
sres....................................identifier of SRES
DEFINED in MAP-MS-DataTypes : 158
SRES....................................type reference OCTET STRING
DEFINED in MAP-MS-DataTypes : 164
Used in MAP-MS-DataTypes : 158
ss-AccessBarred........................identifier of Named Number, 5
DEFINED in MAP-MS-DataTypes : 261
ss-Code..................................identifier of SS-Code
DEFINED in MAP-SupplementaryServi : 206
ss-Code..................................identifier of SS-Code
DEFINED in MAP-MS-DataTypes : 310
ss-Code..................................identifier of SS-Code
DEFINED in MAP-MS-DataTypes : 385
ss-Code..................................identifier of SS-Code
DEFINED in MAP-MS-DataTypes : 457
ss-Code..................................identifier of SS-Code
DEFINED in MAP-SS-DataTypes : 53
ss-Code..................................identifier of SS-Code
DEFINED in MAP-SS-DataTypes : 68
ss-Code..................................identifier of SS-Code
DEFINED in MAP-SS-DataTypes : 122
ss-Code..................................identifier of SS-Code
DEFINED in MAP-SS-DataTypes : 136
ss-Code..................................identifier of SS-Code
DEFINED in MAP-SS-DataTypes : 156
SS-Code.................................type reference OCTET STRING
DEFINED in MAP-SS-Code : 11
Used in MAP-SupplementaryServi : 65 206
Used in MAP-MS-DataTypes : 80 310 385 457
Used in MAP-SS-DataTypes : 45 53 68 122 136 156 205
Used in MAP-SS-Code : 21 25 28 30 32 34 36 41 43
45 47 49 51 54 57 59 63 66
68 70 74 77 80 83 86 89 91
94 97 101 103 105 107 109 112 114
116 120 121 122 123 124 125 126 127
128 129 130 131 132 133 134 135 137
140
Used in MAP-ER-DataTypes : 60 108
DEFINED in MAP-ER-DataTypes : 108
ss-Data.................................identifier of [3] Ext-SS-Data
DEFINED in MAP-MS-DataTypes : 286
ss-Data.................................identifier of [3] SS-Data
DEFINED in MAP-SS-DataTypes : 65
SS-Data..................................type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 135
Used in MAP-SS-DataTypes : 31 65
ss-ErrorStatus..........................value reference SS-ErrorStatus, CHOICE VALUE
DEFINED in MAP-Protocol : 276
SS-ErrorStatus..........................type reference ERROR
DEFINED in MAP-Errors : 275
Used in MAP-Protocol : 107 276
Used in MAP-SupplementaryServi : 37 88 105 122 142
Used in MAP-Errors : 56
ss-ForBS..............................identifier of SS-ForBS-Code
DEFINED in MAP-SupplementaryServi :  93

ss-ForBS..............................identifier of SS-ForBS-Code
DEFINED in MAP-SupplementaryServi :  110

ss-ForBS..............................identifier of SS-ForBS-Code
DEFINED in MAP-SupplementaryServi :  130

ss-ForBS..............................identifier of SS-ForBS-Code
DEFINED in MAP-SupplementaryServi :  149

SS-ForBS-Code..........................type reference SEQUENCE
DEFINED in MAP-SS-DataTypes :  155
USED in MAP-SupplementaryServi :  55  93  110  130  149
USED in MAP-SS-DataTypes :  18

ss-Incompatibility......................value reference SS-Incompatibility, CHOICE VALUE
DEFINED in MAP-Protocol :  279

SS-Incompatibility......................type reference ERROR
DEFINED in MAP-Errors :  284
USED in MAP-Protocol :  110  279
USED in MAP-SupplementaryServi :  40  89  124
USED in MAP-Errors :  59

ss-IncompatibilityCause.................identifier of SS-IncompatibilityCause
DEFINED in MAP-Errors :  286

SS-IncompatibilityCause.................type reference SEQUENCE
DEFINED in MAP-ER-DataTypes :  107
USED in MAP-Errors :  83  286
USED in MAP-ER-DataTypes :  17

ss-Info.................................identifier of SS-Info
DEFINED in MAP-SupplementaryServi :  78

ss-Info.................................identifier of SS-Info
DEFINED in MAP-SupplementaryServi :  95

ss-Info.................................identifier of SS-Info
DEFINED in MAP-SupplementaryServi :  112

ss-Info.................................identifier of SS-Info
DEFINED in MAP-SupplementaryServi :  132

SS-Info.................................type reference CHOICE
DEFINED in MAP-SS-DataTypes :  62
USED in MAP-SupplementaryServi :  54  78  95  112  132
USED in MAP-SS-DataTypes :  24  210

SS-InfoList..............................type reference SEQUENCE OF
DEFINED in MAP-SS-DataTypes :  209
USED in MAP-SS-DataTypes :  25

ss-List.................................identifier of [3] SS-List
DEFINED in MAP-MS-DataTypes :  475

ss-List.................................identifier of [2] SS-List
DEFINED in MAP-MS-DataTypes :  494

ss-List.................................identifier of [1] SS-List
DEFINED in MAP-CH-DataTypes :  110

SS-List.................................type reference SEQUENCE OF
DEFINED in MAP-SS-DataTypes :  204
USED in MAP-SS-DataTypes :  75  475  494
USED in MAP-CH-DataTypes :  38  110
USED in MAP-SS-DataTypes :  24

ss-NotAvailable.........................value reference SS-NotAvailable, CHOICE VALUE
DEFINED in MAP-Protocol :  277

SS-NotAvailable.........................type reference ERROR
DEFINED in MAP-Errors :  280
USED in MAP-Protocol :  108  277
USED in MAP-SupplementaryServi :  38  160
USED in MAP-Errors :  57
ss-Status......................identifier of SS-Status
DEFINED in MAP-Errors : 277

DEFINED in MAP-MS-DataTypes : 321

DEFINED in MAP-MS-DataTypes : 396

DEFINED in MAP-MS-DataTypes : 458

ss-Status......................identifier of [4] SS-Status
DEFINED in MAP-SS-DataTypes : 78

SS-Status......................type reference OCTET STRING
DEFINED in MAP-SS-DataTypes : 85
USED in MAP-Errors : 78 277
USED in MAP-SS-DataTypes : 16 78 132 137 161 166

ss-Status......................identifier of [4] SS-Status
DEFINED in MAP-SS-DataTypes : 132

ss-Status......................identifier of [4] SS-Status
DEFINED in MAP-SS-DataTypes : 137

ss-Status......................identifier of SS-Status
DEFINED in MAP-SS-DataTypes : 161

ss-Status......................identifier of [0] SS-Status
DEFINED in MAP-SS-DataTypes : 166

ss-Status......................identifier of [4] SS-Status
DEFINED in MAP-ER-DataTypes : 110

ss-SubscriptionOption.............identifier of SS-SubscriptionOption
DEFINED in MAP-MS-DataTypes : 459

ss-SubscriptionOption.............identifier of SS-SubscriptionOption
DEFINED in MAP-SS-DataTypes : 138

SS-SubscriptionOption.............type reference CHOICE
DEFINED in MAP-SS-DataTypes : 142
USED in MAP-MS-DataTypes : 74 459
USED in MAP-SS-DataTypes : 17 138

ss-SubscriptionViolation.........value reference SS-SubscriptionViolation, CHOICE VALUE
DEFINED in MAP-Protocol : 278

SS-SubscriptionViolation.........type reference ERROR
DEFINED in MAP-Errors : 282
USED in MAP-Protocol : 109 278
USED in MAP-SupplementaryServi : 39 123 143 214
USED in MAP-Errors : 58

storedMSISDN.....................identifier of ISDN-AddressString
DEFINED in MAP-SM-DataTypes : 121

storedMSISDN.....................identifier of ISDN-AddressString
DEFINED in MAP-SM-DataTypes : 132

subBusyForMT-SMS-Param.............identifier of SubBusyForMT-SMS-Param
DEFINED in MAP-Errors : 306

SubBusyForMT-SMS-Param.............type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 230
USED in MAP-Errors : 109 306
USED in MAP-ER-DataTypes : 40

subscriberBusyForMT-SMS.........value reference SubscriberBusyForMT-SMS, CHOICE VALUE
DEFINED in MAP-Protocol : 290

SubscriberBusyForMT-SMS.........type reference ERROR
DEFINED in MAP-Errors : 304
USED in MAP-Protocol : 116 290
USED in MAP-ShortMessageServic : 38 107
USED in MAP-Errors : 67

SubscriberData------------------type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 210
USED in MAP-MS-DataTypes : 39 206

SubscriberId------------------type reference CHOICE
DEFINED in MAP-CommonDataTypes : 196
USED in MAP-CommonDataTypes : 26

subscriberIdentity------------identifier of [0] SubscriberIdentity
DEFINED in MAP-MS-DataTypes : 677

SubscriberIdentity------------type reference CHOICE
DEFINED in MAP-MS-DataTypes : 688
USED in MAP-MS-DataTypes : 677

subscriberInfo---------------identifier of SubscriberInfo
DEFINED in MAP-MS-DataTypes : 615

SubscriberInfo---------------type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 619
USED in MAP-MS-DataTypes : 62 615 684
USED in MAP-CH-DataTypes : 27 109

subscriberInfo---------------identifier of SubscriberInfo
DEFINED in MAP-MS-DataTypes : 684

subscriberInfo---------------identifier of [7] SubscriberInfo
DEFINED in MAP-CH-DataTypes : 109

subscriberNotMemberOfCUG-------identifier of Named Number, 1
DEFINED in MAP-ER-DataTypes : 103

subscriberNotSC-Subscriber------identifier of Named Number, 6
DEFINED in MAP-ER-DataTypes : 126

subscriberState--------------identifier of [1] SubscriberState
DEFINED in MAP-MS-DataTypes : 621

subscriberState--------------identifier of [1] NULL
DEFINED in MAP-MS-DataTypes : 627

SubscriberState--------------type reference CHOICE
DEFINED in MAP-MS-DataTypes : 662
USED in MAP-MS-DataTypes : 64 621

subscriberStatus---------------identifier of [3] SubscriberStatus
DEFINED in MAP-MS-DataTypes : 213

SubscriberStatus--------------type reference ENUMERATED
DEFINED in MAP-MS-DataTypes : 232
USED in MAP-MS-DataTypes : 41 213

subsequentHandoverFailure-------value reference SubsequentHandoverFailure, CHOICE
VALUE
DEFINED in MAP-Protocol : 247

SubsequentHandoverFailure-------type reference ERROR
DEFINED in MAP-Errors : 204
USED in MAP-Protocol : 94 247
USED in MAP-MobileServiceOpera : 64 201
USED in MAP-Errors : 35

successfulTransfer-------------identifier of Named Number, 2
DEFINED in MAP-SM-DataTypes : 118

supportedCamelPhases------------identifier of [6] SupportedCamelPhases
DEFINED in MAP-MS-DataTypes : 479

SupportedCamelPhases------------type reference BIT STRING
DEFINED in MAP-MS-DataTypes : 555
USED in MAP-MS-DataTypes : 47 479
USED in MAP-CH-DataTypes : 30 163

supportedCamelPhases------------identifier of SupportedCamelPhases
DEFINED in MAP-CH-DataTypes : 163
suppressionOfAnnouncement.................identifier of [12] SuppressionOfAnnouncement
  DEFINED in MAP-CH-DataTypes : 81

SuppressionOfAnnouncement.................type reference NULL
  DEFINED in MAP-CH-DataTypes : 85
  USED in MAP-CH-DataTypes : 21 81 135

suppressionOfAnnouncement.................identifier of [7] SuppressionOfAnnouncement
  DEFINED in MAP-CH-DataTypes : 135

suppress-T-CSI..........................identifier of NULL
  DEFINED in MAP-CH-DataTypes : 164

systemFailure...........................value reference SystemFailure, CHOICE VALUE
  DEFINED in MAP-Protocol : 217

SystemFailure............................type reference ERROR
  DEFINED in MAP-Protocol : 121
  USED in MAP-Protocol : 79 217
  USED in MAP-MobileServiceOpera : 54 113 160 174 212 225 267
  USED in MAP-OperationAndMainte : 23 57 71
  USED in MAP-CallHandlingOperat : 23 61 83
  USED in MAP-SupplementaryServi : 30 81 98 115 135 153 168 181 195 210
  USED in MAP-ShortMessageServic : 27 72 88 100 128
  USED in MAP-Errors : 14

systemFailureParam.......................identifier of SystemFailureParam
  DEFINED in MAP-Errors : 123

SystemFailureParam........................type reference CHOICE
  DEFINED in MAP-Errors : 142
  USED in MAP-Errors : 86 123
  USED in MAP-ER-DataTypes : 20

targetCellId...........................identifier of GlobalCellId
  DEFINED in MAP-MS-DataTypes : 171

targetCellId...........................identifier of GlobalCellId
  DEFINED in MAP-MS-DataTypes : 182

targetMSC-Number........................identifier of ISDN-AddressString
  DEFINED in MAP-MS-DataTypes : 183

tBCD-STRING............................type reference OCTET STRING
  DEFINED in MAP-CommonDataTypes : 63
  USED in MAP-CommonDataTypes : 191 200

TCAPMessages.............................module reference
  DEFINED in TCAPMessages : 1
  USED in MAP-MobileServiceOpera : 51
  USED in MAP-OperationAndMainte : 20
  USED in MAP-CallHandlingOperat : 20
  USED in MAP-SupplementaryServi : 27
  USED in MAP-ShortMessageServic : 24
  USED in MAP-Errors : 75

telephony...............................value reference TeleserviceCode, '00010001'B
  DEFINED in MAP-TS-Code : 41

  DEFINED in MAP-CommonDataTypes : 268

TeleserviceCode..........................type reference OCTET STRING
  DEFINED in MAP-TS-Code : 11
  USED in MAP-CommonDataTypes : 42 268
  USED in MAP-TS-Code : 38 40 41 42 44 45 46 48 49 50 51 55 58 67 69 70 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87

  DEFINED in MAP-MS-DataTypes : 217

TeleserviceList..........................type reference SEQUENCE OF
  DEFINED in MAP-MS-DataTypes : 241
  USED in MAP-MS-DataTypes : 217 473
teleserviceList..........................identifier of [1] TeleserviceList
DEFINED in MAP-MS-DataTypes : 473

teleserviceNot Provisioned..............value reference TeleserviceNotProvided, CHOICE VALUE
DEFINED in MAP-Protocol : 239

TeleserviceNotProvided....................type reference ERROR
DEFINED in MAP-Errors : 193
USED in MAP-Protocol : 92 239
USED in MAP-CallHandlingOperat : 31 69
USED in MAP-SupplementaryServ : 34 85 102 119 139 157
USED in MAP-ShortMessageServic : 35 77
USED in MAP-Errors : 31

telesevNotProvParam.....................identifier of TeleservNotProvParam
DEFINED in MAP-Errors : 195

TeleservNotProvParam.....................type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 194
USED in MAP-Errors : 97 195
USED in MAP-ER-DataTypes : 31

temporaryDefaultAllowed.................identifier of Named Number, 2
DEFINED in MAP-SS-DataTypes : 149

temporaryDefaultRestricted..............identifier of Named Number, 1
DEFINED in MAP-SS-DataTypes : 148

termAttemptAuthorized...................identifier of Named Number, 12
DEFINED in MAP-CH-DataTypes : 201

tmsi.....................................identifier of TMSI
DEFINED in MAP-MobileServiceOpera : 134

TMSI.....................................type reference OCTET STRING
DEFINED in MAP-CommonDataTypes : 194
USED in MAP-MobileServiceOpera : 97 134
USED in MAP-CommonDataTypes : 25 198
tmsi.....................................identifier of [1] TMSI
DEFINED in MAP-CommonDataTypes : 198
	tooManyZoneCodes........................identifier of Named Number, 1
DEFINED in MAP-MS-DataTypes : 485

traceReference..........................identifier of [1] TraceReference
DEFINED in MAP-OM-DataTypes : 38

TraceReference..........................type reference OCTET STRING
DEFINED in MAP-OM-DataTypes : 44
USED in MAP-OM-DataTypes : 38 56

traceReference..........................identifier of [1] TraceReference
DEFINED in MAP-OM-DataTypes : 56

traceType...............................identifier of [2] TraceType
DEFINED in MAP-OM-DataTypes : 39

TraceType...............................type reference INTEGER
DEFINED in MAP-OM-DataTypes : 46
USED in MAP-OM-DataTypes : 39

tracingBufferFull.......................value reference TracingBufferFull, CHOICE VALUE
DEFINED in MAP-Protocol : 253

TracingBufferFull.......................type reference ERROR
DEFINED in MAP-Errors : 209
USED in MAP-Protocol : 95 253
USED in MAP-OperationAndMainte : 29 62
USED in MAP-Errors : 38

tracingBufferFullParam...................identifier of TracingBufferFullParam
DEFINED in MAP-Errors : 211

TracingBufferFullParam...................type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 198
ETSI TS 100 974 V5.18.0 (2003-03)

UNUSED in MAP-Errors : 98 211
UNUSED in MAP-ER-DataTypes : 32

TransactionID..................type reference OCTET STRING
DEFINED in TCAPMessages : 100
USED in TCAPMessages : 47 97 98

T-BcsmCamelTDPData.................type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 192
USED in MAP-CH-DataTypes : 190

t-BcsmCamelTDPDataList..............identifier of T-BcsmCamelTDPDataList
DEFINED in MAP-CH-DataTypes : 185

T-BcsmCamelTDPDataList...............type reference SEQUENCE OF
DEFINED in MAP-CH-DataTypes : 189
USED in MAP-CH-DataTypes : 185

t-BcsmTriggerDetectionPoint...........identifier of T-BcsmTriggerDetectionPoint
DEFINED in MAP-CH-DataTypes : 193

T-BcsmTriggerDetectionPoint...........type reference ENUMERATED
DEFINED in MAP-CH-DataTypes : 200
USED in MAP-CH-DataTypes : 193

t-CSI................................identifier of [0] T-CSI
DEFINED in MAP-CH-DataTypes : 179

T-CSI...................................type reference SEQUENCE
DEFINED in MAP-CH-DataTypes : 184
USED in MAP-CH-DataTypes : 179

undetermined......................identifier of Named Number, 0
DEFINED in MAP-ER-DataTypes : 114

unexpectedDataParam...............identifier of UnexpectedDataParam
DEFINED in MAP-Errors : 134

UnexpectedDataParam...............type reference SEQUENCE
DEFINED in MAP-ER-DataTypes : 158
USED in MAP-Errors : 88 134
USED in MAP-ER-DataTypes : 22

unexpectedDataValue...............value reference UnexpectedDataValue, CHOICE VALUE
DEFINED in MAP-Protocol : 219

UnexpectedDataValue...............type reference ERROR
DEFINED in MAP-Errors : 132
USED in MAP-Protocol : 81 219
USED in MAP-MobileServiceOpera : 56 115 125 150 163 176 198 214 239 250 269
USED in MAP-OperationAndMainte : 25 59 73 84
USED in MAP-CallHandlingOperat : 25 63 85 99
USED in MAP-SupplementaryServi : 32 83 100 117 137 155 170 183 197 212
USED in MAP-ShortMessageServic : 29 74 89 102 119 130 142
USED in MAP-Errors : 16

unexpectedError..................identifier of Named Number, 3
DEFINED in TCAPMessages : 199

unexpectedLinkedOperation..........identifier of Named Number, 7
DEFINED in TCAPMessages : 190

unidentifiedSubParam...............identifier of UnidentifiedSubParam
DEFINED in MAP-Errors : 162

UnidentifiedSubParam...............type reference UnidentifiedSubParam
DEFINED in MAP-ER-DataTypes : 178
USED in MAP-Errors : 92 162
USED in MAP-ER-DataTypes : 27

unidentifiedSubscriber...............value reference UnidentifiedSubscriber, CHOICE VALUE
DEFINED in MAP-Protocol : 228

UnidentifiedSubscriber...............type reference ERROR
DEFINED in MAP-Errors : 160
USED in MAP-Protocol : 86 228
used in MAP-MobileServiceOpera: 59 139 240 251
used in MAP-OperationAndMainte: 28 61 75
used in MAP-ShortMessageServic: 32 104
used in MAP-Errors: 23

unidirectional..........................identifier of [APPLICATION 1] IMPLICIT Unidirectional
DEFINED in TCAPMessages: 52

Unidirectional..........................type reference SEQUENCE
DEFINED in TCAPMessages: 58
used in TCAPMessages: 52

unknownAlphabet.........................value reference UnknownAlphabet, CHOICE VALUE
DEFINED in MAP-Protocol: 280

UnknownAlphabet.........................type reference ERROR
DEFINED in MAP-Errors: 289
used in MAP-Protocol: 111 280
used in MAP-SupplementaryServi: 44 171 187 201
used in MAP-Errors: 60

unknownEquipment.......................value reference UnknownEquipment, CHOICE VALUE
DEFINED in MAP-Protocol: 229

UnknownEquipment.......................type reference ERROR
DEFINED in MAP-Errors: 166
used in MAP-Protocol: 87 229
used in MAP-MobileServiceOpera: 60 227
used in MAP-Errors: 24

unknownMSC..............................value reference UnknownMSC, CHOICE VALUE
DEFINED in MAP-Protocol: 227

UnknownMSC..............................type reference ERROR
DEFINED in MAP-Errors: 158
used in MAP-Protocol: 85 227
used in MAP-MobileServiceOpera: 58 200
used in MAP-Errors: 22

unknownServiceCentre....................identifier of Named Number, 3
DEFINED in MAP-ER-DataTypes: 123

unknownSubscriber.......................value reference UnknownSubscriber, CHOICE VALUE
DEFINED in MAP-Protocol: 225

UnknownSubscriber.......................type reference ERROR
DEFINED in MAP-Errors: 147
used in MAP-Protocol: 83 225
used in MAP-MobileServiceOpera: 57 116 164 215 270
used in MAP-OperationAndMainte: 27 85
used in MAP-CallHandlingOperat: 28 66
used in MAP-ShortMessageServic: 31 76 120 144
used in MAP-Errors: 20

unknownSubscriberParam..................identifier of UnknownSubscriberParam
DEFINED in MAP-Errors: 149

UnknownSubscriberParam..................type reference SEQUENCE
DEFINED in MAP-ER-DataTypes: 170
used in MAP-Errors: 90 149
used in MAP-ER-DataTypes: 25

unrecognizedComponent..................identifier of Named Number, 0
DEFINED in TCAPMessages: 179

unrecognizedError.......................identifier of Named Number, 2
DEFINED in TCAPMessages: 198

unrecognizedInvokeID....................identifier of Named Number, 0
DEFINED in TCAPMessages: 192

unrecognizedInvokeID....................identifier of Named Number, 0
DEFINED in TCAPMessages: 196

unrecognizedLinkedID....................identifier of Named Number, 5
DEFINED in TCAPMessages: 188

unrecognizedMessageType...............identifier of Named Number, 0
DEFINED in TCAPMessages : 103
unrecognizedOperation..................identifier of Named Number, 1
DEFINED in TCAPMessages : 184
unrecognizedTransactionID................identifier of Named Number, 1
DEFINED in TCAPMessages : 104
unstructuredSS-Notify..................value reference UnstructuredSS-Notify, CHOICE VALUE
DEFINED in MAP-Protocol : 192
UnstructuredSS-Notify..................type reference OPERATION
DEFINED in MAP-SupplementaryServi : 190
USED in MAP-Protocol : 59 192
USED in MAP-SupplementaryServi : 20
unstructuredSS-Request..................value reference UnstructuredSS-Request, CHOICE VALUE
DEFINED in MAP-Protocol : 191
UnstructuredSS-Request..................type reference OPERATION
DEFINED in MAP-SupplementaryServi : 174
USED in MAP-Protocol : 58 191
USED in MAP-SupplementaryServi : 19
updateLocation..........................value reference UpdateLocation, CHOICE VALUE
DEFINED in MAP-Protocol : 128
UpdateLocation..........................type reference OPERATION
DEFINED in MAP-MobileServiceOpera : 107
USED in MAP-Protocol : 12 128
USED in MAP-MobileServiceOpera : 15
updateLocationArg.......................identifier of UpdateLocationArg
DEFINED in MAP-MobileServiceOpera : 109
UpdateLocationArg.......................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 120
USED in MAP-MobileServiceOpera : 69 109
USED in MAP-MS-DataTypes : 16
updateLocationRes.......................identifier of UpdateLocationRes
DEFINED in MAP-MobileServiceOpera : 111
UpdateLocationRes.......................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 128
USED in MAP-MobileServiceOpera : 70 111
USED in MAP-MS-DataTypes : 17
ussd-Arg................................identifier of USSD-Arg
DEFINED in MAP-SupplementaryServi : 164
ussd-Arg................................identifier of USSD-Arg
DEFINED in MAP-SupplementaryServi : 176
ussd-Arg................................identifier of USSD-Arg
DEFINED in MAP-SupplementaryServi : 192
USSD-Arg................................type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 171
USED in MAP-SupplementaryServi : 57 164 176 192
USED in MAP-SS-DataTypes : 20
ussd-Busy...............................value reference USSD-Busy, CHOICE VALUE
DEFINED in MAP-Protocol : 281
USSD-Busy...............................type reference ERROR
DEFINED in MAP-Errors : 291
USED in MAP-Protocol : 112 281
USED in MAP-SupplementaryServi : 45 188 202
USED in MAP-Errors : 61
ussd-DataCodingScheme...................identifier of USSD-DataCodingScheme
DEFINED in MAP-SS-DataTypes : 172
ussd-DataCodingScheme...................identifier of USSD-DataCodingScheme
DEFINED in MAP-SS-DataTypes : 177
USSD-DataCodingScheme...................type reference OCTET STRING
ussd-Res................................identifier of USSD-Res
DEFINED in MAP-SupplementaryServi : 166

ussd-Res................................identifier of USSD-Res
DEFINED in MAP-SupplementaryServi : 178

USSD-Res................................type reference SEQUENCE
DEFINED in MAP-SS-DataTypes : 176
USED in MAP-SupplementaryServi : 58 166 178
USED in MAP-SS-DataTypes : 21

ussd-String.............................identifier of USSD-String
DEFINED in MAP-SS-DataTypes : 173

ussd-String.............................identifier of USSD-String
DEFINED in MAP-SS-DataTypes : 178

USSD-String.............................type reference OCTET STRING
DEFINED in MAP-SS-DataTypes : 186
USED in MAP-SupplementaryServi : 33 173 178

uss.................................value reference SS-Code, '10000001'B
DEFINED in MAP-SS-Code : 97

VBSDataList.............................type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 580
USED in MAP-MS-DataTypes : 224

vbsGroupIndication......................identifier of [7] NULL
DEFINED in MAP-MS-DataTypes : 497

DEFINED in MAP-MS-DataTypes : 224

VGCSDataList.............................type reference SEQUENCE OF
DEFINED in MAP-MS-DataTypes : 583
USED in MAP-MS-DataTypes : 225

vgcsGroupIndication.....................identifier of [8] NULL
DEFINED in MAP-MS-DataTypes : 498

vgcsSubscriptionData....................identifier of [12] VGCSDataList
DEFINED in MAP-MS-DataTypes : 225

vlr.....................................identifier of Named Number, 2
DEFINED in MAP-CommonDataTypes : 231

vlrCamelSubscriptionInfo................identifier of [13] VlrCamelSubscriptionInfo
DEFINED in MAP-MS-DataTypes : 226

VlrCamelSubscriptionInfo................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 514
USED in MAP-MS-DataTypes : 226

vlr-Number..............................identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 123

vlr-Number..............................identifier of ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 139

vlr-number..............................identifier of [1] ISDN-AddressString
DEFINED in MAP-MS-DataTypes : 634

vmsc....................................identifier of Named Number, 5
DEFINED in MAP-CommonDataTypes : 234

vmsc-Address.............................identifier of [2] ISDN-AddressString
DEFINED in MAP-CH-DataTypes : 113

voiceBroadcastCall......................value reference TeleserviceCode, '10010010'B
DEFINED in MAP-TS-Code : 70

VoiceBroadcastData......................type reference SEQUENCE
DEFINED in MAP-MS-DataTypes : 595
USED in MAP-MS-DataTypes : 581
voiceGroupCall..........................value reference TeleserviceCode, '10010001'B
  DEFINED in MAP-TS-Code : 69

VoiceGroupCallData......................type reference SEQUENCE
  DEFINED in MAP-MS-DataTypes : 590
  USED in MAP-MS-DataTypes : 584

whiteListed.............................identifier of Named Number, 0
  DEFINED in MAP-MS-DataTypes : 197

ZoneCode................................type reference OCTET STRING
  DEFINED in MAP-MS-DataTypes : 467
  USED in MAP-MS-DataTypes : 465 496

ZoneCodeList............................type reference SEQUENCE OF
  DEFINED in MAP-MS-DataTypes : 464
  USED in MAP-MS-DataTypes : 42 223

zoneCodesConflict.......................identifier of Named Number, 2
  DEFINED in MAP-MS-DataTypes : 486
Annex B (informative): Fully expanded ASN.1 sources for abstract syntaxes of MAP

Annex B is not part of the standard, it is included for information purposes only.

For every (Value)Assignment in the root ASN.1 module all the used defined types and defined values, which are defined within the ASN.1 module or imported from ASN.1 modules, are replaced by the constructs this type or value is composed of.

The fully expanded ASN.1 root module is itself a correct and equivalent representation of the MAP-Protocol.

It allows to see at all the parameters, including all nested ones for a specific operation code or error code at once.

Note that for those operations which use a result without parameters the keyword RESULT is not shown. Empty results are only defined in the ASN.1 description in clause 14.

B.1 Fully Expanded ASN.1 Source of MAP-Protocol/TCAPMessages

```
MAP-Protocol { 0 identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
map-Protocol (4) version3 (3) }
DEFINITIONS
::=
BEGIN
updateLocation OPERATION
ARGUMENT
  updateLocationArg SEQUENCE {
    imsi       OCTET STRING ( SIZE (3 .. 8 ) ),
    msc-Number [1] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
    vlr-Number OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
    lmsi       [10] IMPLICIT OCTET STRING ( SIZE (4 ) ) OPTIONAL,
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
        SEQUENCE {
          extId      MAP-EXTENSION .extensionId ( { 
            ..., } ) ,
          extType    MAP-EXTENSION .ExtensionType ( [ 
            ..., ] @extId ) OPTIONAL) OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ..., } OPTIONAL,
        } OPTIONAL,
      ... } OPTIONAL,
    }
RESULT
  updateLocationRes SEQUENCE {
    hlr-Number OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
        SEQUENCE {
          extId      MAP-EXTENSION .extensionId ( { 
            ..., } ) ,
          extType    MAP-EXTENSION .ExtensionType ( [ 
            ..., ] @extId ) OPTIONAL) OPTIONAL,
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ..., } OPTIONAL,
        } OPTIONAL,
      ... } OPTIONAL,
  }
```

... }
ERRORS {
  -- systemFailure  -- localValue : 34,
  -- dataMissing   -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- unknownSubscriber -- localValue : 1,
  -- roamingNotAllowed -- localValue : 8
} ::= localValue : 2

cancelLocation OPERATION
ARGUMENT
cancelLocationArg CHOICE {
  imsi       OCTET STRING ( SIZE (3 .. 8 ) ),
  imsi-WithLMSI SEQUENCE {
    imsi       OCTET STRING ( SIZE (3 .. 8 ) ),
    lmsi       OCTET STRING ( SIZE (4 ) ),
    ... } }
ERRORS {
  -- dataMissing   -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36
} ::= localValue : 3

purgeMS OPERATION
ARGUMENT
  purgeMS-Arg SEQUENCE {
    imsi       OCTET STRING ( SIZE (3 .. 8 ) ),
    vlr-Number OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
    ... }
 ::= localValue : 67

sendIdentification OPERATION
ARGUMENT
 ARGUMENT
tmsi       OCTET STRING ( SIZE (1 .. 4 ) )
RESULT
sendIdentificationRes SEQUENCE {
  imsi       OCTET STRING ( SIZE (3 .. 8 ) ),
  authenticationSetList SEQUENCE SIZE (1 .. 5 ) OF
  SEQUENCE {
    rand       OCTET STRING ( SIZE (16 ) ),
    sres       OCTET STRING ( SIZE (4 ) ),
    kc         OCTET STRING ( SIZE (8 ) ),
    ... } OPTIONAL,
    ... }
ERRORS {
  -- dataMissing   -- localValue : 35,
  -- unidentifiedSubscriber -- localValue : 5
} ::= localValue : 55

prepareHandover OPERATION
ARGUMENT
prepareHO-Arg SEQUENCE {
  targetCellId OCTET STRING ( SIZE (5 .. 7 ) ) OPTIONAL,
  ho-NumberNotRequired NULL OPTIONAL,
  bss-APDU   SEQUENCE {
    protocolId ENUMERATED {
      gsm-0408  (1 ),
      gsm-0806  (2 ),
      gsm-BSSMAP  (3 ),
      ets-300102-1  (4 )},
    signalInfo OCTET STRING ( SIZE (1 .. 200 ) ),
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
      SEQUENCE {
        extId      MAP-EXTENSION .&extensionId  ( {,
        ... } ),
        extType    MAP-EXTENSION .&ExtensionType  ( {,
        ... } @extId   ) OPTIONAL,}
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... ) OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
protocolId ENUMERATED {
    gsm-0408 (1 ),
gsm-0806 (2 ),
gsm-BSSMAP (3 ),
et-300102-1 (4 ),
}
signalInfo OCTET STRING ( SIZE (1 .. 200 ) ), extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( { ,
            ... } ) ,
        extType MAP-EXTENSION .&ExtensionType ( { ,
            ... } ) OPTIONAL ) OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
            ... ) OPTIONAL,
            ... } OPTIONAL,
            ... ) OPTIONAL
    } OPTIONAL,
    ... ) OPTIONAL,
    ... ) OPTIONAL,
    ... ) OPTIONAL
}
ERRORS {
    -- systemFailure -- localValue : 34,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- noHandoverNumberAvailable -- localValue : 25
} ::= localValue : 68
sendEndSignal OPERATION
ARGUMENT
    bss-APDU SEQUENCE {
        protocolId ENUMERATED {
            gsm-0408 (1 ),
gsm-0806 (2 ),
gsm-BSSMAP (3 ),
et-300102-1 (4 ),
}
signalInfo OCTET STRING ( SIZE (1 .. 200 ) ), extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( { ,
            ... } ) ,
        extType MAP-EXTENSION .&ExtensionType ( { ,
            ... } ) OPTIONAL ) OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... ) OPTIONAL,
            ... ) OPTIONAL,
            ... } OPTIONAL,
            ... ) OPTIONAL
    } OPTIONAL,
    ... ) OPTIONAL,
    ... ) OPTIONAL
} ::= localValue : 29
processAccessSignalling OPERATION
ARGUMENT
    bss-APDU SEQUENCE {
        protocolId ENUMERATED {
            gsm-0408 (1 ),
gsm-0806 (2 ),
gsm-BSSMAP (3 ),
et-300102-1 (4 ),
}
signalInfo OCTET STRING ( SIZE (1 .. 200 ) ), extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( { ,
            ... } ) ,
        extType MAP-EXTENSION .&ExtensionType ( { ,
            ... } ) OPTIONAL ) OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... ) OPTIONAL,
            ... ) OPTIONAL,
            ... } OPTIONAL,
            ... ) OPTIONAL
    } OPTIONAL,
    ... ) OPTIONAL,
    ... ) OPTIONAL
} ::= localValue : 33
forwardAccessSignalling OPERATION
ARGUMENT
    bss-APDU SEQUENCE {
protocolId ENUMERATED {
gsm-0408 (1),
gsm-0806 (2),
gsm-BSSMAP (3),
ets-300102-1 (4),
}
signalInfo OCTET STRING ( SIZE (1 .. 200 ) ),
extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId      MAP-EXTENSION .&extensionId ( { }
        ..., ) ),
      extType    MAP-EXTENSION .&ExtensionType ( { }
        ..., ) OPTIONAL) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE ( }
        ... ) OPTIONAL,
  } OPTIONAL,
...
}
::= localValue : 34

prepareSubsequentHandover OPERATION
ARGUMENT
prepareSubsequentHO-Arg SEQUENCE {
targetCellId OCTET STRING ( SIZE (5 .. 7 ) ),
targetMSC-Number OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
bss-APDU SEQUENCE {
  protocolId ENUMERATED {
    gsm-0408 (1),
    gsm-0806 (2),
    gsm-BSSMAP (3),
ets-300102-1 (4),
  } signalInfo OCTET STRING ( SIZE (1 .. 200 ) ),
extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId      MAP-EXTENSION .&extensionId ( { }
        ..., ) ),
      extType    MAP-EXTENSION .&ExtensionType ( { }
        ..., ) ) OPTIONAL) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE ( }
        ... ) OPTIONAL,
  } OPTIONAL,
...
}
RESULT
bss-APDU SEQUENCE {
  protocolId ENUMERATED {
    gsm-0408 (1),
    gsm-0806 (2),
    gsm-BSSMAP (3),
ets-300102-1 (4),
  } signalInfo OCTET STRING ( SIZE (1 .. 200 ) ),
extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId      MAP-EXTENSION .&extensionId ( { }
        ..., ) ),
      extType    MAP-EXTENSION .&ExtensionType ( { }
        ..., ) ) OPTIONAL) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE ( }
        ... ) OPTIONAL,
  } OPTIONAL,
...
}
ERRORS {
-- unexpectedDataValue -- localValue : 36,
-- dataMissing -- localValue : 35,
-- unknownMSC -- localValue : 3,
-- subsequentHandoverFailure -- localValue : 26
}::= localValue : 69

sendAuthenticationInfo OPERATION
ARGUMENT
sendAuthenticationInfoArg OCTET STRING ( SIZE (3 .. 8 ) )
RESULT
sendAuthenticationInfoRes SEQUENCE SIZE (1 .. 5) OF
  sendAuthenticationInfoRes {rand OCTET STRING (SIZE (16)),sres OCTET STRING (SIZE (4)),kc OCTET STRING (SIZE (8))}
ERRORS {
-- systemFailure -- localValue : 34,
-- dataMissing -- localValue : 35,
-- unexpectedDataValue -- localValue : 36,
-- unknownSubscriber -- localValue : 1}
::= localValue : 56

checkIMEI OPERATION
ARGUMENT
imci OCTET STRING (SIZE (8))
RESULT
equipmentStatus ENUMERATED {
  whiteListed (0),
  blackListed (1),
  greyListed (2) }
ERRORS {
-- systemFailure -- localValue : 34,
-- dataMissing -- localValue : 35,
-- unknownEquipment -- localValue : 7}
::= localValue : 43

insertSubscriberData OPERATION
ARGUMENT
insertSubscriberDataArg SEQUENCE {
  imsi [0] IMPLICIT OCTET STRING (SIZE (3 .. 8)) OPTIONAL,
  msisdn [1] IMPLICIT OCTET STRING (SIZE (1 .. 20)) (SIZE (1 .. 9)) OPTIONAL,
  category [2] IMPLICIT OCTET STRING (SIZE (1)) OPTIONAL,
  subscriberStatus [3] IMPLICIT ENUMERATED {
    serviceGranted (0),
    operatorDeterminedBarring (1) } OPTIONAL,
  bearerServiceList [4] IMPLICIT OCTET STRING (SIZE (1 .. 50)) OF
    OCTET STRING (SIZE (1 .. 5)) OPTIONAL,
  teleserviceList [5] IMPLICIT OCTET STRING (SIZE (1 .. 20)) OF
    OCTET STRING (SIZE (1 .. 5)) OPTIONAL,
  provisionedSS [7] IMPLICIT OCTET STRING (SIZE (1 .. 30)) OF
    CHOICE {
      forwardingInfo [0] IMPLICIT OCTET STRING {
        ss-Code OCTET STRING (SIZE (1)),
        forwardingFeatureList SEQUENCE SIZE (1 .. 32) OF
          SEQUENCE {
            basicService CHOICE {
              ext-BearerService [2] IMPLICIT OCTET STRING (SIZE (1 .. 5))
            },
            ext-TeleService [3] IMPLICIT OCTET STRING (SIZE (1 .. 5)) } OPTIONAL,
        ss-Status [4] IMPLICIT OCTET STRING (SIZE (1 .. 5)),
        forwardedToNumber [5] IMPLICIT OCTET STRING (SIZE (1 .. 20)) (SIZE (1 .. 9)) OPTIONAL,
        forwardedToSubaddress [8] IMPLICIT OCTET STRING (SIZE (1 .. 21)) OPTIONAL,
        forwardingOptions [6] IMPLICIT OCTET STRING (SIZE (1 .. 5)) OPTIONAL,
        noReplyConditionTime [7] IMPLICIT INTEGER (1 .. 100) OPTIONAL,
        extensionContainer [9] IMPLICIT OCTET STRING {
          privateExtensionList [0] IMPLICIT OCTET STRING SIZE (1 .. 10) OF
            SEQUENCE {
              extId MAP-EXTENSION .&extensionId {
                ..., 
              },
              extType MAP-EXTENSION .&ExtensionType {
                ..., 
              } ( @extId ) } OPTIONAL,
            pcs-Extensions [1] IMPLICIT OCTET STRING {
              ..., 
            } OPTIONAL,
            ... },
          extensionContainer [0] IMPLICIT OCTET STRING {
            privateExtensionList [0] IMPLICIT OCTET STRING SIZE (1 .. 10) OF
            SEQUENCE {
              extId MAP-EXTENSION .&extensionId {
                ..., 
              },
              extType MAP-EXTENSION .&ExtensionType {
                ..., 
              } ( @extId ) } OPTIONAL,
            pcs-Extensions [1] IMPLICIT OCTET STRING {
              ..., 
            } OPTIONAL,
            ... },
        },
      extensionContainer [0] IMPLICIT OCTET STRING {
        privateExtensionList [0] IMPLICIT OCTET STRING SIZE (1 .. 10) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId {
            ..., 
          },
          extType MAP-EXTENSION .&ExtensionType {
            ..., 
          } ( @extId ) } OPTIONAL,
        pcs-Extensions [1] IMPLICIT OCTET STRING {
          ..., 
        } OPTIONAL,
        ... },
    } OPTIONAL,
  }
::= localValue : 34
callBarringInfo [1] IMPLICIT SEQUENCE {
  ss-Code OCTET STRING (SIZE (1)),
  callBarringFeatureList SEQUENCE SIZE (1..32) OF
    basicService CHOICE {
      ext-BearerService [2] IMPLICIT OCTET STRING (SIZE (1..5)),
      ext-Teleservice [3] IMPLICIT OCTET STRING (SIZE (1..5))} OPTIONAL,
  ss-Status [4] IMPLICIT OCTET STRING (SIZE (1..5)),
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1..10) OF
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {{ }, ... })
      extType MAP-EXTENSION .&ExtensionType ( {{ }, ... }) { @extId } OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE { ... } OPTIONAL,
    ... } OPTIONAL,
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1..10) OF
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( {{ }, ... })
      extType MAP-EXTENSION .&ExtensionType ( {{ }, ... }) { @extId } OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE { ... } OPTIONAL,
    ... } OPTIONAL,
  ... },

cug-Info [2] IMPLICIT SEQUENCE {
  cug-SubscriptionList SEQUENCE SIZE (0..10) OF
    SEQUENCE {
      cug-Index INTEGER (0..32767),
      cug-Interlock OCTET STRING (SIZE (4)),
      intraCUG-Options ENUMERATED {
        noCUG-Restrictions (0),
        cugIC-CallBarred (1),
        cugOG-CallBarred (2)},
    basicServiceGroupList SEQUENCE SIZE (1..32) OF
      CHOICE {
        ext-BearerService [2] IMPLICIT OCTET STRING (SIZE (1..5)),
        ext-Teleservice [3] IMPLICIT OCTET STRING (SIZE (1..5))} OPTIONAL,
    extensionContainer [0] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1..10) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( {{ }, ... })
        extType MAP-EXTENSION .&ExtensionType ( {{ }, ... }) { @extId } OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE { ... } OPTIONAL,
      ... } OPTIONAL,
    ... },
  cug-FeatureList SEQUENCE SIZE (1..32) OF
    basicService CHOICE {
      ext-BearerService [2] IMPLICIT OCTET STRING (SIZE (1..5)),
      ext-Teleservice [3] IMPLICIT OCTET STRING (SIZE (1..5))} OPTIONAL,
... ) ),
  extType MAP-EXTENSION .&ExtensionType ( {
... ) ( @extId ) ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... ) OPTIONAL,
extensionContainer [0] IMPLICIT SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
  SEQUENCE {
    extId MAP-EXTENSION .&extensionId ( {
... ) ),
  extType MAP-EXTENSION .&ExtensionType ( {
... ) ( @extId ) ) OPTIONAL} OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... ) OPTIONAL,
... } OPTIONAL,
ss-Data [3] IMPLICIT SEQUENCE {
  ss-Code OCTET STRING ( SIZE (1 ) ),
  ss-Status [4] IMPLICIT OCTET STRING ( SIZE (1 .. 5 ) ),
  ss-SubscriptionOption CHOICE {
    cliRestrictionOption [2] IMPLICIT ENUMERATED {
      permanent (0 ),
      temporaryDefaultRestricted (1 ),
      temporaryDefaultAllowed (2 )},
    overrideCategory [1] IMPLICIT ENUMERATED {
      overrideEnabled (0 ),
      overrideDisabled (1 )}) OPTIONAL,
  basicServiceGroupList SEQUENCE SIZE (1 .. 32 ) OF
  CHOICE {
    ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1 .. 5 ) ),
    ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 .. 5 ) ) ) OPTIONAL,
  extensionContainer [5] IMPLICIT SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
  SEQUENCE {
    extId MAP-EXTENSION .&extensionId ( {
... ) ),
  extType MAP-EXTENSION .&ExtensionType ( {
... ) ( @extId ) ) OPTIONAL} OPTIONAL,
  pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... } OPTIONAL,
... ),
emlpp-Info [4] IMPLICIT SEQUENCE {
  maxmumentitledPriority INTEGER ( 0 .. 15 ),
  defaultPriority INTEGER ( 0 .. 15 )},
extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
  SEQUENCE {
    extId MAP-EXTENSION .&extensionId ( {
... ) ),
  extType MAP-EXTENSION .&ExtensionType ( {
... ) ( @extId ) ) OPTIONAL} OPTIONAL,
  pcs-Extensions [1] IMPLICIT SEQUENCE {
... } OPTIONAL,
... ) OPTIONAL,
odb-Data [8] IMPLICIT SEQUENCE {
  odb-GenerData BIT STRING {
    allLOG-CallsBarred (0 ),
    internationalLOGCallsBarred (1 ),
    internationalLOGCallsNotToHPLMN-CountryBarred (2 ),
    interzonalLOGCallsBarred (6 ),
    interzonalLOGCallsNotToHPLMN-CountryBarred (7 ),
    interzonalLOGCallsAndInternationalLOGCallsNotToHPLMN-CountryBarre (8 ),
    premiumRateInformationLOGCallsBarred (3 ),
    premiumRateEntertainementLOGCallsBarred (4 ),
    ss-AccessBarred (5 ),
    allECT-Barred (9 ),
    chargeableECT-Barred (10 ),
internationalECT-Barred (11),
interzonalECT-Barred (12),
doublyChargeableECT-Barred (13),
multipleECT-Barred (14) ( SIZE (15 .. 32 ) ),
onb-HPLMN-Data BIT STRING {
  plmn-SpecificBarringType1 (0 ),
  plmn-SpecificBarringType2 (1 ),
  plmn-SpecificBarringType3 (2 ),
  plmn-SpecificBarringType4 (3 ) ( SIZE (4 .. 32 ) ) OPTIONAL,
}

extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId MAP-EXTENSION .&extensionId {
        ... },
      extType MAP-EXTENSION .&ExtensionType {
        ... } ( @extId ) OPTIONAL; OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... ) OPTIONAL,
        roamingRestrictionDueToUnsupportedFeature [9] IMPLICIT NULL OPTIONAL,
      }
    ) OPTIONAL,
  } OPTIONAL,
  odb-HPLMN-Data BIT STRING {
    plmn-SpecificBarringType1  (0 ),
    plmn-SpecificBarringType2  (1 ),
    plmn-SpecificBarringType3  (2 ),
    plmn-SpecificBarringType4  (3 ) ( SIZE (4 .. 32 ) ) OPTIONAL,
  }


extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId MAP-EXTENSION .&extensionId {
        ... ),
      extType MAP-EXTENSION .&ExtensionType {
        ... } ( @extId ) OPTIONAL; OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... ) OPTIONAL,
      } OPTIONAL,
    ) OPTIONAL,
  } OPTIONAL,
  odb-HPLMN-Data BIT STRING {
    plmn-SpecificBarringType1  (0 ),
    plmn-SpecificBarringType2  (1 ),
    plmn-SpecificBarringType3  (2 ),
    plmn-SpecificBarringType4  (3 ) ( SIZE (4 .. 32 ) ) OPTIONAL,
  }


extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId MAP-EXTENSION .&extensionId {
        ... ),
      extType MAP-EXTENSION .&ExtensionType {
        ... } ( @extId ) OPTIONAL; OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... ) OPTIONAL,
      } OPTIONAL,
    ) OPTIONAL,
  } OPTIONAL,
  odb-HPLMN-Data BIT STRING {
    plmn-SpecificBarringType1  (0 ),
    plmn-SpecificBarringType2  (1 ),
    plmn-SpecificBarringType3  (2 ),
    plmn-SpecificBarringType4  (3 ) ( SIZE (4 .. 32 ) ) OPTIONAL,
  }


extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId MAP-EXTENSION .&extensionId {
        ... ),
      extType MAP-EXTENSION .&ExtensionType {
        ... } ( @extId ) OPTIONAL; OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... ) OPTIONAL,
      } OPTIONAL,
    ) OPTIONAL,
  } OPTIONAL,
  odb-HPLMN-Data BIT STRING {
    plmn-SpecificBarringType1  (0 ),
    plmn-SpecificBarringType2  (1 ),
    plmn-SpecificBarringType3  (2 ),
    plmn-SpecificBarringType4  (3 ) ( SIZE (4 .. 32 ) ) OPTIONAL,
  }


extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId MAP-EXTENSION .&extensionId {
        ... ),
      extType MAP-EXTENSION .&ExtensionType {
        ... } ( @extId ) OPTIONAL; OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... ) OPTIONAL,
      } OPTIONAL,
    ) OPTIONAL,
  } OPTIONAL,
RESULT
insertSubscriberDataRes SEQUENCE {
  teleserviceList [1] IMPLICIT SEQUENCE SIZE (1 .. 20 ) OF OCTET STRING ( SIZE (1 .. 5 ) ) OPTIONAL,
  bearerServiceList [2] IMPLICIT SEQUENCE SIZE (1 .. 50 ) OF OCTET STRING ( SIZE (1 .. 5 ) ) OPTIONAL,
  ss-List    [3] IMPLICIT SEQUENCE SIZE (1 .. 30 ) OF OCTET STRING ( SIZE (1 ) ) OPTIONAL,
  odb-GeneralData [4] IMPLICIT BIT STRING {
    allOC-CallsBarred  (0 ),
    internationalOCCallsBarred (1 ),
    interzonalOCCallsBarred (6 ),
    interzonalOCCallsNotToHPLMN-CountryBarred (7 ),
    premiumRateInformationOCallsBarred (3 ),
    premiumRateEntertainmentOCallsBarred (4 ),
    ss-AccessBarred (5 ),
    allECT-Barred (9 ),
    chargeableECT-Barred (10 ),
    internationalECT-Barred (11 ),
    interzonalECT-Barred (12 ),
    doublyChargeableECT-Barred (13 ),
    multipleECT-Barred (14 )} ( SIZE (15 .. 32 ) ) OPTIONAL,
  regionalSubscriptionResponse [5] IMPLICIT ENumerated {
    msc-AreaRestricted (0 ),
    tooManyZoneCodes (1 ),
    zoneCodesConflict (2 ),
    regionalSubscNotSupported (3 )} OPTIONAL,
  supportedCamelPhases [6] IMPLICIT BIT STRING {
    phase1 (0 )} ( SIZE (1 .. 16 ) ) OPTIONAL,
extensionContainer [7] IMPLICIT SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF SEQUENCE {
    extId MAP-EXTENSION.&extensionId ( { , ...) },
    extType MAP-EXTENSION.&ExtensionType ( { , ...) \{ @extId \} ) OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE ( { , ...) } OPTIONAL,
    ... }) OPTIONAL,
    ... } OPTIONAL,
    ... } OPTIONAL,
    ... }
ERRORS {
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- unidentifiedSubscriber -- localValue : 5}
::= localValue : 7

deleteSubscriberData OPERATION
ARGUMENT
deleteSubscriberDataArg SEQUENCE {
  imsi [0] IMPLICIT OCTET STRING ( SIZE (3 .. 8) ),
  basicServiceList [1] IMPLICIT SEQUENCE SIZE (1 .. 70) OF CHOICE {
    ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1 .. 5) ),
    ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 .. 5) )},
  ss-List [2] IMPLICIT SEQUENCE SIZE (1 .. 30) OF OCTET STRING ( SIZE (1) ) OPTIONAL,
  roamingRestrictionDueToUnsupportedFeature [4] IMPLICIT NULL OPTIONAL,
  regionalSubscriptionIdentifier [5] IMPLICIT OCTET STRING ( SIZE (2) ) OPTIONAL,
  vbsGroupInformation [7] IMPLICIT NULL OPTIONAL,
  vpgcsGroupInformation [8] IMPLICIT NULL OPTIONAL,
  camelSubscriptionInfoWithDraw [9] IMPLICIT NULL OPTIONAL,
  extensionContainer [6] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF SEQUENCE {
      extId MAP-EXTENSION.&extensionId ( { , ...) },
      extType MAP-EXTENSION.&ExtensionType ( { , ...) \{ @extId \} ) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE ( { , ...) } OPTIONAL,
      ... }) OPTIONAL,
      ... } OPTIONAL,
      ... }
RESULT
deleteSubscriberDataRes SEQUENCE {
  regionalSubscriptionResponse [0] IMPLICIT ENUMERATED {
    msc-AreaRestricted (0),
    tooManyZoneCodes (1),
    zoneCodesConflict (2),
    regionalSubscNotSupported (3) \} OPTIONAL,
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF SEQUENCE {
      extId MAP-EXTENSION.&extensionId ( { , ...) },
      extType MAP-EXTENSION.&ExtensionType ( { , ...) \{ @extId \} ) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE ( { , ...) } OPTIONAL,
      ... }) OPTIONAL,
      ... } OPTIONAL,
      ... }
ERRORS {
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- unidentifiedSubscriber -- localValue : 5}
::= localValue : 8
reset OPERATION
ARGUMENT
resetArg SEQUENCE {
  hlr-Number OCTET STRING ( SIZE (1 .. 20) ) ( SIZE (1 .. 9) ),
  hlr-List SEQUENCE SIZE (1 .. 50) OF
OCTET STRING ( SIZE (3 .. 8 ) ) OPTIONAL,
... }
::= localValue : 37

forwardCheckSS-Indication OPERATION
::= localValue : 38

restoreData OPERATION
ARGUMENT
restoreDataArg SEQUENCE {
  imsi OCTET STRING ( SIZE (3 .. 8 ) ),
  lmsi OCTET STRING ( SIZE (4 ) ) OPTIONAL,
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId MAP-EXTENSION .&extensionId ( {
        ... } ),
      extType MAP-EXTENSION .&ExtensionType ( {
        ... } [extId ] ) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... ] OPTIONAL,
        ... }) OPTIONAL,
        ... }) OPTIONAL,
        ... }) OPTIONAL,
      ...
    } OPTIONAL,
    ...
  } OPTIONAL,
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...
  ...)
deactivateTraceMode OPERATION
ARGUMENT
disableTraceModeArg SEQUENCE {
  imsi [0] IMPLICIT OCTET STRING ( SIZE (3 .. 8 ) ) OPTIONAL,
  traceReference [1] IMPLICIT OCTET STRING ( SIZE (1 .. 2 ) ),
  extensionContainer [2] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( { ...
          } ) ,
        extType MAP-EXTENSION .&ExtensionType ( { ...
          } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE ( { ...
      } ) OPTIONAL,
    ... } OPTIONAL,
    ... } OPTIONAL,
  } OPTIONAL,
  ... } OPTIONAL,
}
RESULT
disableTraceModeRes SEQUENCE {
  extensionContainer [0] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( { ...
          } ) ,
        extType MAP-EXTENSION .&ExtensionType ( { ...
          } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE ( { ...
      } ) OPTIONAL,
    ... } OPTIONAL,
  } OPTIONAL,
  ... } OPTIONAL,
}
ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- facilityNotSupported -- localValue : 21,
  -- unidentifiedSubscriber -- localValue : 5,
  -- tracingBufferFull -- localValue : 40}
 ::= localValue : 50

sendIMSI OPERATION
ARGUMENT
msisdn OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) )
RESULT
imsi OCTET STRING ( SIZE (3 .. 8 ) )
ERRORS {
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- unknownSubscriber -- localValue : 1}
 ::= localValue : 58

sendRoutingInfo OPERATION
ARGUMENT
msisdn [0] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
cug-CheckInfo [1] IMPLICIT SEQUENCE {
  cug-Interlock OCTET STRING ( SIZE (4 ) ),
  cug-OutgoingAccess NULL OPTIONAL,
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( { ...
          } ) ,
        extType MAP-EXTENSION .&ExtensionType ( { ...
          } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE ( { ...
      } ) OPTIONAL,
    ... } OPTIONAL,
  } OPTIONAL,
  ... } OPTIONAL,
}
RESULT
disableTraceModeRes SEQUENCE {
  extensionContainer [0] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( { ...
          } ) ,
        extType MAP-EXTENSION .&ExtensionType ( { ...
          } ) OPTIONAL} OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE ( { ...
      } ) OPTIONAL,
    ... } OPTIONAL,
  } OPTIONAL,
  ... } OPTIONAL,
}
ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- facilityNotSupported -- localValue : 21,
  -- unidentifiedSubscriber -- localValue : 5,
  -- tracingBufferFull -- localValue : 40}
 ::= localValue : 51

::= localValue : 50

RESULT

sendRoutingInfoRes [3] IMPLICIT SEQUENCE {
  imsi [9] IMPLICIT OCTET STRING ( SIZE (3 .. 8 ) ) OPTIONAL,
extendedRoutingInfo CHOICE {
  routingInfo CHOICE {
    roamingNumber OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
    forwardingData SEQUENCE { 
      forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ) 
    }
  }
  OPTIONAL,
  forwardedToSubaddress [4] IMPLICIT OCTET STRING ( SIZE (1 .. 21 ) ) OPTIONAL, 
  forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL, 
  extensionContainer [7] IMPLICIT SEQUENCE { 
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF SEQUENCE { 
      extId MAP-EXTENSION .&extensionId ({ }, 
        ... ), 
      extType MAP-EXTENSION .&ExtensionType ({ }, 
        ... ) @extId ) ( @extId ) ) OPTIONAL, 
      pcs-Extensions [1] IMPLICIT SEQUENCE { 
        ... } OPTIONAL, 
    } 
    camelRoutingInfo [8] IMPLICIT SEQUENCE { 
      forwardingData SEQUENCE { 
        forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ) 
      }
      OPTIONAL,
      forwardedToSubaddress [4] IMPLICIT OCTET STRING ( SIZE (1 .. 21 ) ) OPTIONAL, 
      forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL, 
      extensionContainer [7] IMPLICIT SEQUENCE { 
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF SEQUENCE { 
          extId MAP-EXTENSION .&extensionId ({ }, 
            ... ), 
          extType MAP-EXTENSION .&ExtensionType ({ }, 
            ... ) @extId ) ( @extId ) ) OPTIONAL, 
          pcs-Extensions [1] IMPLICIT SEQUENCE { 
            ... } OPTIONAL, 
        } 
        gmscCamelSubscriptionInfo [0] IMPLICIT SEQUENCE { 
          t-CSI [0] IMPLICIT SEQUENCE { 
            t-BcsmTriggerDetectionPoint ENUMERATED { 
              termAttemptAuthorized (12 ), 
              ... }, 
            serviceKey INTEGER ( 0 .. 2147483647 ), 
            gsmSCF-Address [0] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ), 
            defaultCallHandling [1] IMPLICIT ENUMERATED { 
              continueCall (0 ), 
              releaseCall (1 ), 
              ... }, 
            extensionContainer [2] IMPLICIT SEQUENCE { 
              privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF SEQUENCE { 
                extId MAP-EXTENSION .&extensionId ({ }, 
                  ... ), 
                extType MAP-EXTENSION .&ExtensionType ({ }, 
                  ... ) @extId ) ( @extId ) ) OPTIONAL, 
                  pcs-Extensions [1] IMPLICIT SEQUENCE { 
                    ... } OPTIONAL, 
                } 
                extensionContainer SEQUENCE { 
                  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF SEQUENCE { 
                    extId MAP-EXTENSION .&extensionId ({ }, 
                      ... ), 
                    extType MAP-EXTENSION .&ExtensionType ({ }, 
                      ... ) @extId ) ( @extId ) ) OPTIONAL, 
                      pcs-Extensions [1] IMPLICIT SEQUENCE { 
                        ... } OPTIONAL, 
                    }, 
          }, 
        }, 
      }
    }
... ) OPTIONAL,
... ) OPTIONAL,
  o-Csi     [1] IMPLICIT SEQUENCE {
    o-BcsmCamelTdpDataList SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      o-BcsmTriggerDetectionPoint ENUMERATED {
        collectedInfo (2 ),
        ...
      },
      serviceKey INTEGER ( 0 .. 2147483647 ),
      gsmSccf-Address [0] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) )
      ( SIZE (1 .. 9 ) ),
      defaultCallHandling [1] IMPLICIT ENUMERATED {
        continueCall (0 ),
        releaseCall (1 ),
        ...
      },
      extensionContainer [2] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( { '...
          extType MAP-EXTENSION .&extensionType ( { '...
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... ) OPTIONAL,
          } OPTIONAL,
        } OPTIONAL,
      } extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( { '...
          extType MAP-EXTENSION .&extensionType ( { '...
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... ) OPTIONAL,
          } OPTIONAL,
        } OPTIONAL,
      } extensionContainer [2] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( { '...
          extType MAP-EXTENSION .&extensionType ( { '...
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... ) OPTIONAL,
          } OPTIONAL,
        } OPTIONAL,
      } extensionContainer [1] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
        SEQUENCE {
          extId MAP-EXTENSION .&extensionId ( { '...
          extType MAP-EXTENSION .&extensionType ( { '...
          pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... ) OPTIONAL,
          } OPTIONAL,
        } OPTIONAL,
      } cug-CheckInfo [3] IMPLICIT SEQUENCE {
      cug-Interlock OCTET STRING ( SIZE (4 ) ),
      cug-OutgoingAccess NULL OPTIONAL,
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
      SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( { '...
        extType MAP-EXTENSION .&extensionType ( { '...
      } extensionContainer [1] IMPLICIT SEQUENCE {
      cug-CheckInfo [3] IMPLICIT SEQUENCE {
      cug-Interlock OCTET STRING ( SIZE (4 ) ),
      cug-OutgoingAccess NULL OPTIONAL,
... ( @extId ) ) OPTIONAL) OPTIONAL,

pcs-Extensions [1] IMPLICIT SEQUENCE { ... ) OPTIONAL,

... ) OPTIONAL,

cugSubscriptionFlag [6] IMPLICIT NULL OPTIONAL,

subscriberInfo [7] IMPLICIT SEQUENCE {

locationInformation [0] IMPLICIT SEQUENCE { agpLocationInformation INTEGER ( 0 .. 32767 ) OPTIONAL,

gEOgraphicalInformation [0] IMPLICIT OCTET STRING ( SIZE (8 ) ) OPTIONAL,

vIr-number [1] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ) OPTIONAL,

locationNumber [2] IMPLICIT OCTET STRING ( SIZE (2 .. 10 ) ) OPTIONAL,

cellIdOrLAI [3] CHOICE { cellIdFixedLength [0] IMPLICIT OCTET STRING ( SIZE (7 ) ),

laIFixedLength [1] IMPLICIT OCTET STRING ( SIZE (5 ) )) OPTIONAL,

extensionContainer [4] IMPLICIT SEQUENCE { privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF

SEQUENCE {

extId MAP-EXTENSION .&extensionId ( 

' ...') ),

extType MAP-EXTENSION .&ExtensionType ( {

' ...') ( @extId ) ) OPTIONAL) OPTIONAL,

pcs-Extensions [1] IMPLICIT SEQUENCE { ... ) OPTIONAL,

... ) OPTIONAL,

subscriberState [1] CHOICE { assumedIdle [0] IMPLICIT NULL,

camelBusy [1] IMPLICIT NULL,

netDetNotReachable ENUMERATED { msPurged (0 ),

imsiDetached (1 ),

restrictedArea (2 ),

notRegistered (3 )},

notProvidedFromVLR [2] IMPLICIT NULL) OPTIONAL,

extensionContainer [2] IMPLICIT SEQUENCE { privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF

SEQUENCE {

extId MAP-EXTENSION .&extensionId ( 

' ...') ),

extType MAP-EXTENSION .&ExtensionType ( {

' ...') ( @extId ) ) OPTIONAL) OPTIONAL,

pcs-Extensions [1] IMPLICIT SEQUENCE { ... ) OPTIONAL,

-- or-NotAllowed -- localValue : 48,

-- unknownSubscriber -- localValue : 48,

-- numberChanged -- localValue : 44,

ERRORS {

-- systemFailure -- localValue : 34,

-- dataMissing -- localValue : 35,

-- unexpectedDataValue -- localValue : 36,

-- facilityNotSupported -- localValue : 21,

-- or-NotAllowed -- localValue : 48,

-- unknownSubscriber -- localValue : 1,

-- numberChanged -- localValue : 44,
-- bearerServiceNotProvisioned -- localValue : 10,
-- teleserviceNotProvisioned -- localValue : 11,
-- absentSubscriber -- localValue : 27,
-- busySubscriber -- localValue : 45,
-- noSubscriberReply -- localValue : 46,
-- callBarred -- localValue : 13,
-- cug-Reject -- localValue : 15,
-- forwardingViolation -- localValue : 14
::= localValue : 22

provideRoamingNumber OPERATION
ARGUMENT
provideRoamingNumberArg SEQUENCE {
    imsi       [0] IMPLICIT OCTET STRING ( SIZE (3 .. 8 ) ),
    msc-Number [1] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
    msisdn     [2] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ) OPTIONAL,
    lmsi       [4] IMPLICIT OCTET STRING ( SIZE (4 ) ) OPTIONAL,
    gsm-BearerCapability [5] IMPLICIT SEQUENCE {
        protocolId ENUMERATED {
            gsm-0408  (1 ),
            gsm-0806  (2 ),
            gsm-BSSMAP (3 ),
            ets-300102-1 (4 )},
        signalInfo OCTET STRING ( SIZE (1 .. 200 ) ),
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
            SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( { ,
                    ...) ,
                extType MAP-EXTENSION .&ExtensionType ( { ,
                    ...) ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE { ,
                ... } OPTIONAL,
            networkSignalInfo [6] IMPLICIT SEQUENCE {
                protocolId ENUMERATED {
                    gsm-0408 (1 ),
                    gsm-0806 (2 ),
                    gsm-BSSMAP (3 ),
                    ets-300102-1 (4 )},
                signalInfo OCTET STRING ( SIZE (1 .. 200 ) ),
                extensionContainer SEQUENCE {
                    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId MAP-EXTENSION .&extensionId ( { ,
                            ...) ,
                        extType MAP-EXTENSION .&ExtensionType ( { ,
                            ...) ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                suppressionOfAnnouncement [7] IMPLICIT NULL OPTIONAL,
                gsmc-Address [8] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ) OPTIONAL,
                callReferenceNumber [9] IMPLICIT OCTET STRING ( SIZE (1 .. 8 ) ) OPTIONAL,
                or-Interrogation [10] IMPLICIT NULL OPTIONAL,
                    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
                    SEQUENCE {
                        extId MAP-EXTENSION .&extensionId ( { ,
                            ...) ,
                        extType MAP-EXTENSION .&ExtensionType ( { ,
                            ...) ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                RESULT
provideRoamingNumberRes SEQUENCE {
    roamingNumber OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
    extensionContainer SEQUENCE {


privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1..10) OF
  SEQUENCE {
    extId MAP-EXTENSION .&extensionId ( { 
      ..., } ),
    extType MAP-EXTENSION .&ExtensionType ( { 
      ..., } [ @extId ] ) OPTIONAL, 
    pcs-Extensions [1] IMPLICIT SEQUENCE ( 
      ..., ) OPTIONAL, 
    ... ) OPTIONAL, 
  } OPTIONAL, 
ERRORS {
  -- systemFailure -- localValue : 34, 
  -- dataMissing -- localValue : 35, 
  -- unexpectedDataValue -- localValue : 36, 
  -- facilityNotSupported -- localValue : 21, 
  -- or-NotAllowed -- localValue : 48, 
  -- absentSubscriber -- localValue : 27, 
  -- noRoamingNumberAvailable -- localValue : 39) 
::= localValue : 4
resumeCallHandling OPERATION
ARGUMENT
  resumeCallHandlingArg SEQUENCE {
    callReferenceNumber [0] IMPLICIT OCTET STRING ( SIZE (1..8) ),
    basicServiceGroup [1] CHOICE {
      ext-BearerService [2] IMPLICIT OCTET STRING ( SIZE (1..5) ),
      ext-Teleservice [3] IMPLICIT OCTET STRING ( SIZE (1..5) )},
    forwardingData [2] IMPLICIT SEQUENCE {
      forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ) 
      OPTIONAL,
      forwardedToSubaddress [4] IMPLICIT OCTET STRING ( SIZE (1..21) ) OPTIONAL,
      forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE (1) ) OPTIONAL,
      extensionContainer [7] IMPLICIT SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1..10) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( { 
              ..., } ),
            extType MAP-EXTENSION .&ExtensionType ( { 
              ..., } [ @extId ] ) OPTIONAL, 
            pcs-Extensions [1] IMPLICIT SEQUENCE ( 
              ..., ) OPTIONAL, 
            ... ) OPTIONAL, 
          } OPTIONAL,
      } OPTIONAL,
      imsi [3] IMPLICIT OCTET STRING ( SIZE (3..8) ),
      cug-CheckInfo [4] IMPLICIT SEQUENCE {
        cug-Interlock OCTET STRING ( SIZE (4 ) ),
        cug-OutgoingAccess NULL OPTIONAL,
      } OPTIONAL,
      extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1..10) OF
          SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( { 
              ..., } ),
            extType MAP-EXTENSION .&ExtensionType ( { 
              ..., } [ @extId ] ) OPTIONAL, 
            pcs-Extensions [1] IMPLICIT SEQUENCE ( 
              ..., ) OPTIONAL, 
            ... ) OPTIONAL, 
          } OPTIONAL,
      } OPTIONAL,
    } OPTIONAL,
    o-CSI [5] IMPLICIT SEQUENCE {
      o-BCsmCamel1TDDataList SEQUENCE SIZE (1..10) OF
        SEQUENCE {
          o-BCsmTriggerDetectionPoint ENUMERATED {
            collectedInfo (2),
            ..., 
          },
          serviceKey INTEGER (0 .. 2147483647),
          gsMSC-Address [0] IMPLICIT OCTET STRING ( SIZE (1..20) ) ( SIZE (1..9) ) 
          ),
          defaultCallHandling [1] IMPLICIT ENUMERATED {
            continueCall (0),
            releaseCall (1),
            ..., 
          },
          extensionContainer [2] IMPLICIT SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1..10) OF
            ... ) OPTIONAL,
          } OPTIONAL,
        } OPTIONAL,
      } OPTIONAL,
    } OPTIONAL,
  } OPTIONAL,
SEQUENCE {
  extId      MAP-EXTENSION .&extensionId  ( { 
      ... } ),
  extType    MAP-EXTENSION .&ExtensionType  ( { 
      ... } { @extId   }  )  OPTIONAL,
  pcs-Extensions [1] IMPLICIT SEQUENCE { 
      ... ) OPTIONAL,
      ... } OPTIONAL,
    ... ),
extensionContainer  SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
    SEQUENCE { 
      extId      MAP-EXTENSION .&extensionId  ( { 
      ... } ),
      extType    MAP-EXTENSION .&ExtensionType  ( { 
      ... } { @extId   }  )  OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE { 
      ... ) OPTIONAL,
      ... } OPTIONAL,
    ... } OPTIONAL,
extensionContainer [7] IMPLICIT SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
    SEQUENCE { 
      extId      MAP-EXTENSION .&extensionId  ( { 
      ... } ),
      extType    MAP-EXTENSION .&ExtensionType  ( { 
      ... } { @extId   }  )  OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE { 
      ... ) OPTIONAL,
      ... ) OPTIONAL,
    ... } OPTIONAL,
resultCallHandlingRes  SEQUENCE {
  extensionContainer  SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
    SEQUENCE { 
      extId      MAP-EXTENSION .&extensionId  ( { 
      ... } ),
      extType    MAP-EXTENSION .&ExtensionType  ( { 
      ... } { @extId   }  )  OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE { 
      ... ) OPTIONAL,
      ... } OPTIONAL,
    ... ) OPTIONAL,
resultCallHandlingRes  SEQUENCE {
  extensionContainer  SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
    SEQUENCE { 
      extId      MAP-EXTENSION .&extensionId  ( { 
      ... } ),
      extType    MAP-EXTENSION .&ExtensionType  ( { 
      ... } { @extId   }  )  OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE { 
      ... ) OPTIONAL,
      ... ) OPTIONAL,
    ... ) OPTIONAL,
resultCallHandlingRes  SEQUENCE {
  extensionContainer  SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
    SEQUENCE { 
      extId      MAP-EXTENSION .&extensionId  ( { 
      ... } ),
      extType    MAP-EXTENSION .&ExtensionType  ( { 
      ... } { @extId   }  )  OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE { 
      ... ) OPTIONAL,
      ... ) OPTIONAL,
    ... ) OPTIONAL,
resultCallHandlingRes  SEQUENCE {
  extensionContainer  SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
    SEQUENCE { 
      extId      MAP-EXTENSION .&extensionId  ( { 
      ... } ),
      extType    MAP-EXTENSION .&ExtensionType  ( { 
      ... } { @extId   }  )  OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE { 
      ... ) OPTIONAL,
      ... ) OPTIONAL,
    ... ) OPTIONAL,
resultCallHandlingRes  SEQUENCE {
  extensionContainer  SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
    SEQUENCE { 
      extId      MAP-EXTENSION .&extensionId  ( { 
      ... } ),
      extType    MAP-EXTENSION .&ExtensionType  ( { 
      ... } { @extId   }  )  OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE { 
      ... ) OPTIONAL,
      ... ) OPTIONAL,
    ... ) OPTIONAL,
ss-Status [4] IMPLICIT OCTET STRING ( SIZE (1) ) OPTIONAL,
forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE (1 .. 20) ) ( SIZE (1 .. 9) )
OPTIONAL,
forwardedToSubaddress [8] IMPLICIT OCTET STRING ( SIZE (1 .. 21) ) OPTIONAL,
forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE (1) ) OPTIONAL,
noReplyConditionTime [7] IMPLICIT INTEGER ( 5 .. 30 ) OPTIONAL,
... ),
callBarringInfo [1] IMPLICIT SEQUENCE {
  ss-Code OCTET STRING ( SIZE (1) ) OPTIONAL,
callBarringFeatureList SEQUENCE SIZE (1 .. 13) OF
  SEQUENCE {
    basicService CHOICE {
      bearerService [2] IMPLICIT OCTET STRING ( SIZE (1) ),
teleService [3] IMPLICIT OCTET STRING ( SIZE (1) )} OPTIONAL,
  ss-Status [4] IMPLICIT OCTET STRING ( SIZE (1) ) OPTIONAL,
  ... },
ss-Data [3] IMPLICIT SEQUENCE {
  ss-Code OCTET STRING ( SIZE (1) ) OPTIONAL,
  ss-Status [4] IMPLICIT OCTET STRING ( SIZE (1) ) OPTIONAL,
  ss-SubscriptionOption CHOICE {
    cliRestrictionOption [2] IMPLICIT ENUMERATED {
      permanent (0),
temporaryDefaultRestricted (1),
temporaryDefaultAllowed (2)},
    overrideCategory [1] IMPLICIT ENUMERATED {
      overrideEnabled (0),
      overrideDisabled (1)} OPTIONAL,
  basicServiceGroupList SEQUENCE SIZE (1 .. 13) OF
    CHOICE {
      bearerService [2] IMPLICIT OCTET STRING ( SIZE (1) ),
teleService [3] IMPLICIT OCTET STRING ( SIZE (1) )} OPTIONAL,
  ... }
}
ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- bearerServiceNotProvisioned -- localValue : 10,
  -- teleserviceNotProvisioned -- localValue : 11,
  -- callBarred -- localValue : 13,
  -- illegalSS-Operation -- localValue : 16,
  -- ss-ErrorStatus -- localValue : 17,
  -- ss-Incompatibility -- localValue : 20
 ::= localValue : 10
eraseSS OPERATION
ARGUMENT
  ss-ForBS SEQUENCE {
    ss-Code OCTET STRING ( SIZE (1) ),
    basicService CHOICE {
      bearerService [2] IMPLICIT OCTET STRING ( SIZE (1) ),
teleService [3] IMPLICIT OCTET STRING ( SIZE (1) )} OPTIONAL,
  ... }
RESULT
  ss-Info CHOICE {
    forwardingInfo [0] IMPLICIT SEQUENCE {
      ss-Code OCTET STRING ( SIZE (1) ) OPTIONAL,
      forwardingFeatureList SEQUENCE SIZE (1 .. 13) OF
        SEQUENCE {
          basicService CHOICE {
            bearerService [2] IMPLICIT OCTET STRING ( SIZE (1) ),
teleService [3] IMPLICIT OCTET STRING ( SIZE (1) )} OPTIONAL,
          ss-Status [4] IMPLICIT OCTET STRING ( SIZE (1) ) OPTIONAL,
          forwardedToNumber [5] IMPLICIT OCTET STRING ( SIZE (1 .. 20) ) ( SIZE (1 .. 9) )
        OPTIONAL,
      forwardedToSubaddress [8] IMPLICIT OCTET STRING ( SIZE (1 .. 21) ) OPTIONAL,
      forwardingOptions [6] IMPLICIT OCTET STRING ( SIZE (1) ) OPTIONAL,
      noReplyConditionTime [7] IMPLICIT INTEGER ( 5 .. 30 ) OPTIONAL,
      ... },
callBarringInfo [1] IMPLICIT SEQUENCE {
  ss-Code OCTET STRING ( SIZE (1) ) OPTIONAL,
callBarringFeatureList SEQUENCE SIZE (1 .. 13) OF
  SEQUENCE {
    basicService CHOICE {
      bearerService [2] IMPLICIT OCTET STRING ( SIZE (1) ),
teleService [3] IMPLICIT OCTET STRING ( SIZE (1) )} OPTIONAL,
activateSS OPERATION
ARGUMENT
ss-ForBS SEQUENCE {
  ss-Code OCTET STRING (SIZE 1 ))
, basicService CHOICE {
  bearerService [2] IMPLICIT OCTET STRING (SIZE 1 ))
, teleservice [3] IMPLICIT OCTET STRING (SIZE 1 )) } OPTIONAL,
... }
RESULT
ss-Info CHOICE {
  forwardingInfo [0] IMPLICIT SEQUENCE {
    ss-Code OCTET STRING (SIZE 1 )) OPTIONAL,
  forwardingFeatureList SEQUENCE SIZE (1 .. 13 ) OF
  SEQUENCE {
    basicService CHOICE {
      bearerService [2] IMPLICIT OCTET STRING (SIZE 1 ))
, teleservice [3] IMPLICIT OCTET STRING (SIZE 1 )) } OPTIONAL,
    ... }
OPTIONAL,
forwardedToNumber [5] IMPLICIT OCTET STRING (SIZE 1 .. 20 ) (SIZE 1 .. 9 )
, forwardedToSubaddress [8] IMPLICIT OCTET STRING (SIZE 1 .. 21 ) OPTIONAL,
forwardingOptions [6] IMPLICIT OCTET STRING (SIZE 1 ) OPTIONAL,
noReplyConditionTime [7] IMPLICIT INTEGER ( 5 .. 30 ) OPTIONAL,
... },
, call1BarringInfo [1] IMPLICIT SEQUENCE {
    ss-Code OCTET STRING (SIZE 1 )) OPTIONAL,
  call1BarringFeatureList SEQUENCE SIZE (1 .. 13 ) OF
  SEQUENCE {
    basicService CHOICE {
      bearerService [2] IMPLICIT OCTET STRING (SIZE 1 ))
, teleservice [3] IMPLICIT OCTET STRING (SIZE 1 )) } OPTIONAL,
      ss-Status [4] IMPLICIT OCTET STRING (SIZE 1 )) ) OPTIONAL,
      forwardedToNumber [5] IMPLICIT OCTET STRING (SIZE 1 .. 20 ) (SIZE 1 .. 9 )
    } SIZE (1 .. 9 )
, ... },
, ss-Data [3] IMPLICIT SEQUENCE {
    ss-Code OCTET STRING (SIZE 1 )) OPTIONAL,
  ss-Status [4] IMPLICIT OCTET STRING (SIZE 1 )) ) OPTIONAL,
  ss-SubscriptionOption CHOICE {
    cliRestrictionOption [2] IMPLICIT ENUMERATED {
      permanent (0 ),
      temporaryDefaultRestricted (1 ),
      temporaryDefaultAllowed (2 )
    }
, overrideCategory [1] IMPLICIT ENUMERATED {
    overrideEnabled (0 ),
    overrideDisabled (1 )
    } OPTIONAL,
  basicServiceGroupList SEQUENCE SIZE (1 .. 13 ) OF
  CHOICE {
interrogateSS OPERATION
ARGUMENT
ss-ForBS SEQUENCE {
  ss-Code OCTET STRING (SIZE (1 )),
  basicService CHOICE {
    bearerService [2] IMPLICIT OCTET STRING (SIZE (1 )),
    teleservice [3] IMPLICIT OCTET STRING (SIZE (1 ))) OPTIONAL,
  ...
}
RESULT
interrogateSS-Res CHOICE {
  ss-Status [0] IMPLICIT OCTET STRING (SIZE (1 )) ,
  basicServiceGroupList [2] IMPLICIT SEQUENCE SIZE (1 .. 13 ) OF
    CHOICE {
      bearerService [2] IMPLICIT OCTET STRING (SIZE (1 )),
      teleservice [3] IMPLICIT OCTET STRING (SIZE (1 ))) ,
    forwardingFeatureList [3] IMPLICIT SEQUENCE SIZE (1 .. 13 ) OF
      SEQUENCE {
        basicService CHOICE {
          bearerService [2] IMPLICIT OCTET STRING (SIZE (1 )) ,
          teleservice [3] IMPLICIT OCTET STRING (SIZE (1 ))) OPTIONAL, ss-Status [4] IMPLICIT OCTET STRING (SIZE (1 )) OPTIONAL,
        forwardingToNumber [5] IMPLICIT OCTET STRING (SIZE (1 .. 20 )) (SIZE (1 .. 9 ))
      OPTIONAL,
      cli-RestrictionInfo [4] IMPLICIT SEQUENCE {
        ss-Status OCTET STRING (SIZE (1 )),
        cliRestrictionOption ENUMERATED {
          permanent (0 ), temporaryDefaultRestricted (1 ), temporaryDefaultAllowed (2 ) OPTIONAL,
        ...
      }
    }
  }
ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- bearerServiceNotProvisioned -- localValue : 10,
  -- teleserviceNotProvisioned -- localValue : 11,
  -- callBarred -- localValue : 13,
  -- illegalSS-Operation -- localValue : 16,
  -- ss-NotAvailable -- localValue : 18
}
ussd-String OCTET STRING ( SIZE (1 .. 160 ) ),
... }

ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- absentSubscriber -- localValue : 27,
  -- illegalSubscriber -- localValue : 9,
  -- illegalEquipment -- localValue : 12,
  -- unknownAlphabet -- localValue : 71,
  -- ussd-Busy -- localValue : 72}
::= localValue : 60

unstructuredSS-Notify OPERATION
ARGUMENT
ussd-Arg SEQUENCE {
  ussd-DataCodingScheme OCTET STRING ( SIZE (1 ) ),
  ussd-String OCTET STRING ( SIZE (1 .. 160 ) ),
  ... }

ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- absentSubscriber -- localValue : 27,
  -- illegalSubscriber -- localValue : 9,
  -- illegalEquipment -- localValue : 12,
  -- unknownAlphabet -- localValue : 71,
  -- ussd-Busy -- localValue : 72}
::= localValue : 61

registerPassword OPERATION
ARGUMENT
ss-Code OCTET STRING ( SIZE (1 ) )
RESULT
newPassword NumericString ( FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9" )|SIZE (4 ) )

ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- callBarred -- localValue : 13,
  -- ss-SubscriptionViolation -- localValue : 19,
  -- pw-RegistrationFailure -- localValue : 37,
  -- negativePW-Check -- localValue : 38,
  -- numberOfPW-AttemptsViolation -- localValue : 43}
LINKED {
  -- getPassword -- localValue : 18}
::= localValue : 17

getPassword OPERATION
ARGUMENT
  guidanceInfo ENUMERATED {
    enterPW (0 ),
    enterNewPW (1 ),
    enterNewPW-Again (2 )}
RESULT
  currentPassword NumericString ( FROM ("0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9" )|SIZE (4 ) )
::= localValue : 18

sendRoutingInfoForSM OPERATION
ARGUMENT
routingInfoForSM-Arg SEQUENCE {
  msisdn [0] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) )

  | ( SIZE (1 .. 9 ) ),
  ss-RP-PRI [1] IMPLICIT BOOLEAN,
  serviceCentreAddress [2] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ),
  extensionContainer [6] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 )

    OF SEQUENCE {
      extId MAP-EXTENSION .&extensionId ( {
        ... } ),
      extType MAP-EXTENSION .&ExtensionType ( {
        ... } ) OPTIONAL ] OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE ( {
        ... } ) OPTIONAL,
    ... } OPTIONAL,
    ... ] OPTIONAL,
    ... } OPTIONAL,

    ... ]

RESULT
routingInfoForSM-Res SEQUENCE {
  imsi   OCTET STRING ( SIZE (3 .. 8 ) ),
  locationInfoWithLMSI [0] IMPLICIT SEQUENCE {
    msc-Number [1] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
    imsi   [2] OCTET STRING ( SIZE (4 ) ) OPTIONAL,
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId    MAP-EXTENSION .&extensionId ( { }
        , ...
      ) ,
      extType  MAP-EXTENSION .&ExtensionType ( { ...
        , @extId } ) OPTIONAL,)
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
  } OPTIONAL,
  ... },
  extensionContainer [4] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId    MAP-EXTENSION .&extensionId ( { }
        , ...
      ) ,
      extType  MAP-EXTENSION .&ExtensionType ( { ...
        , @extId } ) OPTIONAL,)
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
  } OPTIONAL,
  ... },
ERRORS {
  -- systemFailure             localValue : 34,
  -- dataMissing               localValue : 35,
  -- unexpectedDataValue       localValue : 36,
  -- facilityNotSupported      localValue : 21,
  -- unknownSubscriber         localValue : 1,
  -- teleserviceNotProvisioned localValue : 11,
  -- callBarred                localValue : 13,
  -- absentsubscriberSM        localValue : 6}
::= localValue : 45

mo-forwardSM OPERATION
ARGUMENT
  mo-forwardSM-Arg SEQUENCE {
    sm-RP-DA   CHOICE {
      imsi      [0] IMPLICIT OCTET STRING ( SIZE (3 .. 8 ) ),
      msc-Number [1] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
      locationInfoWithLMSI [4] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ),
    sm-RP-DA   CHOICE {
      msisdn    [2] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
      serviceCentreAddressDA [4] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ),
      serviceCentreAddressOA [4] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ),
    sm-RP-UI   OCTET STRING ( SIZE (1 .. 200 ) ),
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId    MAP-EXTENSION .&extensionId ( { }
        , ...
      ) ,
      extType  MAP-EXTENSION .&ExtensionType ( { ...
        , @extId } ) OPTIONAL,)
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
  } OPTIONAL,
  ...
  },
RESULT
  mo-forwardSM-Res SEQUENCE {
    sm-RP-UI   OCTET STRING ( SIZE (1 .. 200 ) ),
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId    MAP-EXTENSION .&extensionId ( { }
        , ...
      ) ,
      extType  MAP-EXTENSION .&ExtensionType ( { ...
        , @extId } ) OPTIONAL,)
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
  } OPTIONAL,
  ...
  }
```
ERRORS {
  -- systemFailure -- localValue : 34,
  -- unexpectedDataValue -- localValue : 36,
  -- facilityNotSupported -- localValue : 21,
  -- sm-DeliveryFailure -- localValue : 32)
::= localValue : 46

mt-forwardSM OPERATION
ARGUMENT
mt-forwardSM-Arg SEQUENCE {
  sm-RP-DA CHOICE {
    imsi [0] IMPLICIT OCTET STRING ( SIZE (3 .. 8 ) ),
    lmsi [1] IMPLICIT OCTET STRING ( SIZE (4 ) ),
    serviceCentreAddressDA [4] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ),
    noSM-RP-DA [5] IMPLICIT NULL,
  }
  sm-RP-OA CHOICE {
    msisdn [2] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
    serviceCentreAddressOA [4] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ),
    noSM-RP-OA [5] IMPLICIT NULL,
  }
  sm-RP-UI OCTET STRING ( SIZE (1 .. 200 ) ),
  moreMessagesToSend NULL OPTIONAL,
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId MAP-EXTENSION .&extensionId ( { 
        ,...
      } ),
      extType MAP-EXTENSION .&ExtensionType ( { 
        ,...
      } { @extId } ) OPTIONAL,}
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ,...
    } OPTIONAL,
    ...
  } OPTIONAL,
  ...
} OPTIONAL,

RESULT
mt-forwardSM-Res SEQUENCE {
  sm-RP-UI OCTET STRING ( SIZE (1 .. 200 ) ) OPTIONAL,
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
      extId MAP-EXTENSION .&extensionId ( { 
        ,...
      } ),
      extType MAP-EXTENSION .&ExtensionType ( { 
        ,...
      } { @extId } ) OPTIONAL,}
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ,...
    } OPTIONAL,
    ...
  } OPTIONAL,
  ...
} OPTIONAL,

ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- facilityNotSupported -- localValue : 21,
  -- unidentifiedSubscriber -- localValue : 5,
  -- illegalSubscriber -- localValue : 9,
  -- illegalEquipment -- localValue : 12,
  -- subscriberBusyForMT-SMS -- localValue : 31,
  -- sm-DeliveryFailure -- localValue : 32,
  -- absentsubscriberSM -- localValue : 6}
::= localValue : 44

reportSM-DeliveryStatus OPERATION
ARGUMENT
reportSM-DeliveryStatusArg SEQUENCE {
  msisdn OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
  serviceCentreAddress OCTET STRING ( SIZE (1 .. 20 ) ),
  sm-DeliveryOutcome ENUMERATED {
    memoryCapacityExceeded (0 ),
    absentSubscriber (1 ),
    successfulTransfer (2 )},
```
absentSubscriberDiagnosticSM [0] IMPLICIT INTEGER (0 .. 255) OPTIONAL,
extensionContainer [1] IMPLICIT SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF
  SEQUENCE {
    extId MAP-EXTENSION &extensionId ( { }
    ...
    ) ,
    extType MAP-EXTENSION &ExtensionType ( { }
    ...
    ) OPTIONAL, pcs-Extensions [1] IMPLICIT SEQUENCE ( ...
    ...
    ) OPTIONAL,
  ...
  }
}

RESULT

reportSM-DeliveryStatusRes SEQUENCE {
  storedMSISDN OCTET STRING (SIZE (1 .. 20)) (SIZE (1 .. 9)) OPTIONAL,
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF
    SEQUENCE {
      extId MAP-EXTENSION &extensionId ( { }
      ...
      ) ,
      extType MAP-EXTENSION &ExtensionType ( { }
      ...
      ) OPTIONAL, pcs-Extensions [1] IMPLICIT SEQUENCE ( ...
      ...
      ) OPTIONAL,
    ...
  }
}

ERRORS {
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- unknownSubscriber -- localValue : 1,
  -- messageWaitingListFull -- localValue : 33}
::= localValue : 47

informServiceCentre OPERATION
ARGUMENT

informServiceCentreArg SEQUENCE {
  storedMSISDN OCTET STRING (SIZE (1 .. 20)) (SIZE (1 .. 9)) OPTIONAL,
  mw-Status BIT STRING {
    sc-AddressNotIncluded (0),
    mnrf-Set (1),
    mcef-Set (2)} (SIZE (6 .. 16)) OPTIONAL,
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF
    SEQUENCE {
      extId MAP-EXTENSION &extensionId ( { }
      ...
      ) ,
      extType MAP-EXTENSION &ExtensionType ( { }
      ...
      ) OPTIONAL, pcs-Extensions [1] IMPLICIT SEQUENCE ( ...
      ...
      ) OPTIONAL,
    ...
  }
}
::= localValue : 63

alertServiceCentre OPERATION
ARGUMENT

alertServiceCentreArg SEQUENCE {
  msisdn OCTET STRING (SIZE (1 .. 20)) (SIZE (1 .. 9)),
  serviceCentreAddress OCTET STRING (SIZE (1 .. 20)),
  ...
}

ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36}
::= localValue : 64

readyForSM OPERATION
ARGUMENT

readyForSM-Arg SEQUENCE {
  imsi [0] IMPLICIT OCTET STRING (SIZE (3 .. 8)),
  alertReason ENUMERATED {
    ms-Present (0),
  }
}
PROVIDE SUBSCRIBER INFO

ARGUMENT
provideSubscriberInfoArg SEQUENCE {
  imsi [0] IMPLICIT OCTET STRING ( SIZE (3 .. 8 ) ) OPTIONAL,
  lmsi [1] IMPLICIT OCTET STRING ( SIZE (4 ) ) OPTIONAL,
  requestedInfo [2] IMPLICIT SEQUENCE {
    locationInformation [0] IMPLICIT NULL OPTIONAL,
    subscriberState [1] IMPLICIT NULL OPTIONAL,
    extensionContainer [2] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
        SEQUENCE {
          extId   MAP-EXTENSION .&extensionId ( { ,
            , } ),
          extType  MAP-EXTENSION .&ExtensionType ( { ,
            , } ) OPTIONAL, pcs-Extensions [1] IMPLICIT SEQUENCE {
            , , } OPTIONAL,
            } OPTIONAL,
        } OPTIONAL,
        } OPTIONAL,
      } OPTIONAL,
      } OPTIONAL,
    } OPTIONAL,
  } OPTIONAL,
  } OPTIONAL,
  extensionContainer [2] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
      SEQUENCE {
        extId   MAP-EXTENSION .&extensionId ( { ,
          , } ),
        extType  MAP-EXTENSION .&ExtensionType ( { ,
          , } ) OPTIONAL, pcs-Extensions [1] IMPLICIT SEQUENCE {
            , , } OPTIONAL,
            } OPTIONAL,
        } OPTIONAL,
        } OPTIONAL,
      } OPTIONAL,
    } OPTIONAL,
    } OPTIONAL,
    } OPTIONAL,
    } OPTIONAL,
    extensionContainer [3] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
      SEQUENCE {
        extId   MAP-EXTENSION .&extensionId ( { ,
          , } ),
        extType  MAP-EXTENSION .&ExtensionType ( { ,
          , } ) OPTIONAL, pcs-Extensions [1] IMPLICIT SEQUENCE {
            , , } OPTIONAL,
            } OPTIONAL,
        } OPTIONAL,
        } OPTIONAL,
      } OPTIONAL,
      } OPTIONAL,
    } OPTIONAL,
    },
    } OPTIONAL,
    } OPTIONAL,
    },
    } OPTIONAL,
    subscriberState [1] CHOICE {
      assumedIdle [0] IMPLICIT NULL,
      camelBusy [1] IMPLICIT NULL,
      netDetNotReachable ENUMERATED {
        msPurged (0 ),
        imsiDetached (1 ),
        restrictedArea (2 ),
        notRegistered (3 )},
      notProvidedFromVLR [2] IMPLICIT NULL OPTIONAL,
      extensionContainer [2] IMPLICIT SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    }
  }}
}
SEQUENCE {
  extId      MAP-EXTENSION .&extensionId ( {
    
    } ) ,
  extType    MAP-EXTENSION .&ExtensionType ( {
    
    } ) OPTIONAL) OPTIONAL,
  pcs-Extensions [1] IMPLICIT SEQUENCE ( 
    ... ) OPTIONAL,
    ... ) OPTIONAL,
  ... ),
  extensionContainer SEQUENCE { 
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
  SEQUENCE ( 
    extId      MAP-EXTENSION .&extensionId ( { 
    
    } ) ,
    extType    MAP-EXTENSION .&ExtensionType ( { 
    
    } ) OPTIONAL) OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE ( 
    ... ) OPTIONAL,
    ... ) OPTIONAL,
    ... ) OPTIONAL,
  ERRORS { 
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36} 
  ::= localValue : 70
}

anyTimeInterrogation OPERATION
ARGUMENT
anyTimeInterrogationArg SEQUENCE {
  subscriberIdentity [0] CHOICE {
    imsi       [0] IMPLICIT OCTET STRING ( SIZE (3 .. 8 ) ),
    msisdn     [1] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
  },
  requestedInfo [1] IMPLICIT SEQUENCE { 
    locationInformation [0] IMPLICIT NULL OPTIONAL,
    subscriberState [1] IMPLICIT NULL OPTIONAL,
    extensionContainer [2] IMPLICIT SEQUENCE { 
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
    SEQUENCE ( 
      extId      MAP-EXTENSION .&extensionId ( { 
      
      } ) ,
      extType    MAP-EXTENSION .&ExtensionType ( { 
      
      } ) OPTIONAL) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE ( 
      ... ) OPTIONAL,
      ... ) OPTIONAL,
      ... ) OPTIONAL,
    gsmSCF-Address [3] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ),
    extensionContainer [2] IMPLICIT SEQUENCE { 
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
    SEQUENCE ( 
      extId      MAP-EXTENSION .&extensionId ( { 
      
      } ) ,
      extType    MAP-EXTENSION .&ExtensionType ( { 
      
      } ) OPTIONAL) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE ( 
      ... ) OPTIONAL,
      ... ) OPTIONAL,
    ... ) OPTIONAL,
  },
  RESULT
  anyTimeInterrogationRes SEQUENCE {
    subscriberInfo SEQUENCE {
      locationInformation [0] IMPLICIT SEQUENCE ( 
      ageOfLocationInformation INTEGER ( 0 .. 32767 ) OPTIONAL,
      geographicalInformation [0] IMPLICIT OCTET STRING ( SIZE (8 ) ) OPTIONAL,
      vlr-number [1] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) ( SIZE (1 .. 9 ) ) OPTIONAL,
      locationNumber [2] IMPLICIT OCTET STRING ( SIZE (2 .. 10 ) ) OPTIONAL,
      cellIdOrLAI [3] CHOICE { 
      cellIdFixedLength [0] IMPLICIT OCTET STRING ( SIZE (7 ) ),
      laiFixedLength [1] IMPLICIT OCTET STRING ( SIZE (5 ) ) } OPTIONAL,
    extensionContainer [4] IMPLICIT SEQUENCE { 
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
  }

ETSİ
SEQUENCE {
    extId MAP-EXTENSION .&extensionId ( { '...' } ),
    extType MAP-EXTENSION .&ExtensionType ( { '...' } { @extId } ) OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE { ... } OPTIONAL,
    ... } OPTIONAL,
subscriberState [1] CHOICE {
    assumedIdle [0] IMPLICIT NULL,
    camelBusy [1] IMPLICIT NULL,
    netDetNotReachable ENUMERATED {
        msPurged (0),
        imsiDetached (1),
        restrictedArea (2),
        notRegistered (3),
        notProvidedFromVLR [2] IMPLICIT NULL)} OPTIONAL,
extensionContainer [2] IMPLICIT SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( { '...' } ),
        extType MAP-EXTENSION .&ExtensionType ( { '...' } { @extId } ) OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE { ... } OPTIONAL,
        ... } OPTIONAL,
    } OPTIONAL,
extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
    SEQUENCE {
        extId MAP-EXTENSION .&extensionId ( { '...' } ),
        extType MAP-EXTENSION .&ExtensionType ( { '...' } { @extId } ) OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE { ... } OPTIONAL,
        ... } OPTIONAL,
    } OPTIONAL,
    } OPTIONAL,
ERRORS {
    -- systemFailure -- localValue : 34,
    -- ati-NotAllowed -- localValue : 49,
    -- dataMissing -- localValue : 35,
    -- unexpectedDataValue -- localValue : 36,
    -- unknownSubscriber -- localValue : 1
    ::= localValue : 71
}

systemFailure ERROR
PARAMETER
systemFailureParam CHOICE {
    networkResource ENUMERATED {
        plmn (0),
        hlr (1),
        vlr (2),
        pvlr (3),
        controllingMSC (4),
        vmsc (5),
        eir (6),
        rss (7)
    },
    extensibleSystemFailureParam SEQUENCE {
        networkResource ENUMERATED {
            plmn (0),
            hlr (1),
            vlr (2),
            pvlr (3),
            controllingMSC (4),
            vmsc (5),
            eir (6),
            rss (7) } OPTIONAL,
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
            SEQUENCE {
            extId MAP-EXTENSION .&extensionId ( { '...' } ),
            extType MAP-EXTENSION .&ExtensionType ( { '...' } { @extId } ) OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE { ... } OPTIONAL,
            ... } OPTIONAL,
        } OPTIONAL,
        ... } OPTIONAL,
SEQUENCE {
  extId      MAP-EXTENSION .&extensionId ( { 
  
  } ),
  extType    MAP-EXTENSION .&ExtensionType ( { 
  
  } ) OPTIONAL, 
  pcs-Extensions [1] IMPLICIT SEQUENCE ( 
  
  ) OPTIONAL, 
  ... ) OPTIONAL, 
... })}
::= localValue : 34

dataMissing ERROR
PARAMETER
dataMissingParam SEQUENCE {
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF 
    SEQUENCE {
      extId      MAP-EXTENSION .&extensionId ( { 
      
      } ),
      extType    MAP-EXTENSION .&ExtensionType ( { 
      
      } ) OPTIONAL, 
      pcs-Extensions [1] IMPLICIT SEQUENCE ( 
      
      ) OPTIONAL, 
      ... ) OPTIONAL, 
      ... }) OPTIONAL, 
  }
::= localValue : 35

unexpectedDataValue ERROR
PARAMETER
unexpectedDataParam SEQUENCE {
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF 
    SEQUENCE {
      extId      MAP-EXTENSION .&extensionId ( { 
      
      } ),
      extType    MAP-EXTENSION .&ExtensionType ( { 
      
      } ) OPTIONAL, 
      pcs-Extensions [1] IMPLICIT SEQUENCE ( 
      
      ) OPTIONAL, 
      ... ) OPTIONAL, 
      ... }) OPTIONAL, 
  }
::= localValue : 36

facilityNotSupported ERROR
PARAMETER
facilityNotSupParam SEQUENCE {
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF 
    SEQUENCE {
      extId      MAP-EXTENSION .&extensionId ( { 
      
      } ),
      extType    MAP-EXTENSION .&ExtensionType ( { 
      
      } ) OPTIONAL, 
      pcs-Extensions [1] IMPLICIT SEQUENCE ( 
      
      ) OPTIONAL, 
      ... ) OPTIONAL, 
      ... }) OPTIONAL, 
  }
::= localValue : 21

unknownSubscriber ERROR
PARAMETER
unknownSubscriberParam SEQUENCE {
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF 
    SEQUENCE {
      extId      MAP-EXTENSION .&extensionId ( { 
      
      } ),
      extType    MAP-EXTENSION .&ExtensionType ( { 
      
      } )
numberChanged ERROR
PARAMETER
numberChangedParam SEQUENCE {
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF SEQUENCE {
      extId     MAP-EXTENSION .&extensionId ( {', ...} ),
      extType   MAP-EXTENSION .&ExtensionType ( {', ...} {@extId} ) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
    }...
  }...
}::= localValue : 1

unknownMSC ERROR
::= localValue : 3

unidentifiedSubscriber ERROR
PARAMETER
unidentifiedSubParam SEQUENCE {
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF SEQUENCE {
      extId     MAP-EXTENSION .&extensionId ( {', ...} ),
      extType   MAP-EXTENSION .&ExtensionType ( {', ...} {@extId} ) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
    }...
  }...
}::= localValue : 5

unknownEquipment ERROR
::= localValue : 7

roamingNotAllowed ERROR
PARAMETER
roamingNotAllowedParam SEQUENCE {
  roamingNotAllowedCause ENUMERATED {
    plmnRoamingNotAllowed (0),
    operatorDeterminedBarring (3)
  },
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF SEQUENCE {
      extId     MAP-EXTENSION .&extensionId ( {', ...} ),
      extType   MAP-EXTENSION .&ExtensionType ( {', ...} {@extId} ) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... } OPTIONAL,
    }...
  }...
}::= localValue : 8

illegalSubscriber ERROR
PARAMETER
illegalSubscriberParam SEQUENCE {
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF SEQUENCE {
      extId     MAP-EXTENSION .&extensionId ( {', ...} ),
    }...
  }...
}::= localValue : 44
illegalEquipmentParam SEQUENCE {
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10) OF SEQUENCE {
      extId MAP-EXTENSION .&extensionId ( {
        ...} ),
      extType MAP-EXTENSION .&ExtensionType ( {
        ...} { @extId } ) OPTIONAL, 
      pcs-Extensions [1] IMPLICIT SEQUENCE { 
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
        ... } OPTIONAL,
absentSubscriber ERROR
PARAMETER
absentSubscriberParam SEQUENCE {
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
    SEQUENCE {
      extId      MAP-EXTENSION .&extensionId ( { 
        ..., } ) ,
      extType    MAP-EXTENSION .&ExtensionType ( { 
        ..., } { @extId   }  )  OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE { 
        ... } OPTIONAL,
      ... } OPTIONAL,
    ... }
  ::= localValue : 27

busySubscriber ERROR
PARAMETER
busySubscriberParam SEQUENCE {
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
    SEQUENCE {
      extId      MAP-EXTENSION .&extensionId ( { 
        ..., } ) ,
      extType    MAP-EXTENSION .&ExtensionType ( { 
        ..., } { @extId   }  )  OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE { 
        ... } OPTIONAL,
      ... } OPTIONAL,
    ... }
  ::= localValue : 45

noSubscriberReply ERROR
PARAMETER
noSubscriberReplyParam SEQUENCE {
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF 
    SEQUENCE {
      extId      MAP-EXTENSION .&extensionId ( { 
        ..., } ) ,
      extType    MAP-EXTENSION .&ExtensionType ( { 
        ..., } { @extId   }  )  OPTIONAL} OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE { 
        ... } OPTIONAL,
callBarred ERROR
  PARAMETER
    callBarredParam CHOICE
      callBarringCause ENUMERATED {
        barringServiceActive (0 ),
        operatorBarring (1 )},
    extensibleCallBarredParam SEQUENCE {
      callBarringCause ENUMERATED {
        barringServiceActive (0 ),
        operatorBarring (1 )} OPTIONAL,
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
        SEQUENCE {
          extId    MAP-EXTENSION .&extensionId ( {
            ,
          },
          extType  MAP-EXTENSION .&ExtensionType ( {
            ,
          } ) OPTIONAL,  
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ,
        } OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,
        ... ) OPTIONAL,  
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,  
        ... ) OPTIONAL,
        ... ) OPTIONAL,  
        ... ) OPTIONAL,
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTIONAL,  
        ... ) OPTION
cug-Reject ERROR
PARAMETER
cug-RejectParam SEQUENCE {
cug-RejectCause ENUMERATED {
incomingCallsBarredWithinCUG (0),
subscriberNotMemberOfCUG (1),
requestedBasicServiceViolatesCUG-Constraints (5),
calledPartySS-InteractionViolation (7) } OPTIONAL,
extensionContainer SEQUENCE { 
privateExtensionList [0] IMPLICIT OCTET STRING ( SIZE (1 .. 10 ) )
SEQUENCE { 
  extId    MAP-EXTENSION .&extensionId  ( { ,
  extType   MAP-EXTENSION .&ExtensionType  ( { ,
  pcs-Extensions [1] IMPLICIT OCTET STRING ( SIZE (1 .. 10 ) ) } OPTIONAL, 
... } OPTIONAL, 
... } OPTIONAL, 
... } OPTIONAL,
... 

} OPTIONAL,
... 

} ::= localValue : 14

ati-NotAllowed ERROR
PARAMETER
ati-NotAllowedParam SEQUENCE {
extensionContainer SEQUENCE { 
privateExtensionList [0] IMPLICIT OCTET STRING ( SIZE (1 .. 10 ) )
SEQUENCE { 
  extId    MAP-EXTENSION .&extensionId  ( { ,
  extType   MAP-EXTENSION .&ExtensionType  ( { ,
  pcs-Extensions [1] IMPLICIT OCTET STRING ( SIZE (1 .. 10 ) ) } OPTIONAL, 
... } OPTIONAL, 
... } OPTIONAL, 
... } OPTIONAL, 
... } OPTIONAL,
... 

} ::= localValue : 15

illegalSS-Operation ERROR
 ::= localValue : 16

ss-ErrorStatus ERROR
PARAMETER
ss-Status OCTET STRING ( SIZE (1 ) )
 ::= localValue : 17

ss-NotAvailable ERROR
 ::= localValue : 18

ss-SubscriptionViolation ERROR
 ::= localValue : 19

ss-Incompatibility ERROR
PARAMETER
ss-IncompatibilityCause SEQUENCE {
  ss-Code    [1] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
basicService CHOICE {
  bearerService [2] IMPLICIT OCTET STRING ( SIZE (1 ) ),
teleservice [3] IMPLICIT OCTET STRING ( SIZE (1 ) ) } OPTIONAL,
ss-Status  [4] IMPLICIT OCTET STRING ( SIZE (1 ) ) } OPTIONAL,
... 

} ::= localValue : 20

unknownAlphabet ERROR
 ::= localValue : 71

ussd-Busy ERROR
 ::= localValue : 72

pw-RegistrationFailure ERROR

ETSI
PARAMETER
  pw-RegistrationFailureCause ENUMERATED {
    undetermined (0 ),
    invalidFormat (1 ),
    newPasswordsMismatch (2 )
  ::= localValue : 37

negativePW-Check ERROR
  ::= localValue : 38

numberOfPW-AttemptsViolation ERROR
  ::= localValue : 43

subscriberBusyForMT-SMS ERROR
PARAMETER
  subBusyForMT-SMS-Param SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
        SEQUENCE {
          extId      MAP-EXTENSION .&extensionId ( { '...
          extType    MAP-EXTENSION .&ExtensionType ( { '...
        ( @extId ) ) OPTIONAL, pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... ) OPTIONAL,
          ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
      ... ) OPTIONAL,
    } ::= localValue : 31

sm-DeliveryFailure ERROR
PARAMETER
  sm-DeliveryFailureCause SEQUENCE {
    sm-EnumeratedDeliveryFailureCause ENUMERATED {
      memoryCapacityExceeded (0 ),
      equipmentProtocolError (1 ),
      equipmentNotSM-Equipped (2 ),
      unknownServiceCentre (3 ),
      subscriberNotSC-Subscriber (6 ),
    diagnosticInfo OCTET STRING ( SIZE (1 .. 200 ) ) OPTIONAL,
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
        SEQUENCE {
          extId      MAP-EXTENSION .&extensionId ( { '...
          extType    MAP-EXTENSION .&ExtensionType ( { '...
        ( @extId ) ) OPTIONAL, pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... ) OPTIONAL,
          ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
      ... ) OPTIONAL,
    } ::= localValue : 32

messageWaitingListFull ERROR
PARAMETER
  messageWaitListFullParam SEQUENCE {
    extensionContainer SEQUENCE {
      privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
        SEQUENCE {
          extId      MAP-EXTENSION .&extensionId ( { '...
          extType    MAP-EXTENSION .&ExtensionType ( { '...
        ( @extId ) ) OPTIONAL, pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... ) OPTIONAL,
          ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE {
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
        ... ) OPTIONAL,
      ... ) OPTIONAL,
    } ::= localValue : 33

absentSubscriberSM ERROR
PARAMETER
absentSubscriberSM-Param SEQUENCE {
    absentSubscriberDiagnosticSM INTEGER ( 0 .. 255 ) OPTIONAL,
    extensionContainer SEQUENCE {
        privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
            SEQUENCE {
                extId MAP-EXTENSION .&extensionId ( { ' , ... } ) ,
                extType MAP-EXTENSION .&ExtensionType ( { ' , ... } ( @extId | ) OPTIONAL) OPTIONAL,
            }
        pcs-Extensions [1] IMPLICIT SEQUENCE {
            ... } OPTIONAL,
        ... } OPTIONAL,
        ... }
::= localValue : 6
END
B.2 Fully Expanded ASN.1 Source of MAP-DialogueInformation

-- Expanded ASN1 Module 'MAP-DialogueInformation'
--SIEMENS ASN.1 Compiler      P3.90M (04-07-00-00-64-00)
--    Date: 98-07-23    Time: 09:09:07

MAP-DialogueInformation ( 0 identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3) map-DialogueInformation (3) version3 (3) )

DEFINITIONS :
BEGIN
EXPORTS
    map-DialogueAS,
    MAP-DialoguePDU ;

map-DialogueAS OBJECT IDENTIFIER ::= { ccitt (0) identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) 1 map-DialoguePDU (1) version1 (1) }

MAP-DialoguePDU ::= CHOICE {
    map-open [0] IMPLICIT SEQUENCE {
        destinationReference [0] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) OPTIONAL,
        originationReference [1] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) OPTIONAL,
        ...,
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF SEQUENCE {
                extId    MAP-EXTENSION .&extensionId   ( {
                    '...'} ) ,
                extType  MAP-EXTENSION .&ExtensionType ( {
                    '...'} @extId ) OPTIONAL) OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL},
    map-accept [1] IMPLICIT SEQUENCE {
        ...,
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF SEQUENCE {
                extId    MAP-EXTENSION .&extensionId   ( { 
                    '...'} ) ,
                extType  MAP-EXTENSION .&ExtensionType ( {
                    '...'} @extId ) OPTIONAL) OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL},
    map-close  [2] IMPLICIT SEQUENCE {
        ...,
        extensionContainer SEQUENCE {
            privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF SEQUENCE {
                extId    MAP-EXTENSION .&extensionId   ( { 
                    '...'} ) ,
                extType  MAP-EXTENSION .&ExtensionType ( {
                    '...'} @extId ) OPTIONAL) OPTIONAL,
                pcs-Extensions [1] IMPLICIT SEQUENCE {
                    ... } OPTIONAL,
                    ... } OPTIONAL},
    map-refuse [3] IMPLICIT SEQUENCE {
        reason  ENUMERATED {
            noReasonGiven (0 ),
            invalidDestinationReference (1 ),
            ... } OPTIONAL,
        ... } OPTIONAL}
invalidOriginatingReference (2 )},
...
extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
  SEQUENCE {
    extId      MAP-EXTENSION .&extensionId ( { ' ...
    extType    MAP-EXTENSION .&ExtensionType ( { ' ...
      ( @extId ) ) OPTIONAL } OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
    ... ) OPTIONAL},
map-userAbort [4] IMPLICIT SEQUENCE {
  map-UserAbortChoice CHOICE {
    userSpecificReason [0] IMPLICIT NULL,
    userResourceLimitation [1] IMPLICIT NULL,
    shortTermResourceLimitation ( 0 ),
    longTermResourceLimitation ( 1 )},
  applicationProcedureCancellation [3] IMPLICIT ENUMERATED {
    handoverCancellation ( 0 ),
    radioChannelRelease ( 1 ),
    networkPathRelease ( 2 ),
    callRelease ( 3 ),
    associatedProcedureFailure ( 4 ),
    tandemDialogueRelease ( 5 ),
    remoteOperationsFailure ( 6 )},
...
extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
  SEQUENCE {
    extId      MAP-EXTENSION .&extensionId ( { ' ...
    extType    MAP-EXTENSION .&ExtensionType ( { ' ...
      ( @extId ) ) OPTIONAL } OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
    ... ) OPTIONAL},
map-providerAbort [5] IMPLICIT SEQUENCE {
  map-ProviderAbortReason ENUMERATED {
    abnormalDialogue ( 0 ),
    invalidPDU ( 1 )},
...
extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
  SEQUENCE {
    extId      MAP-EXTENSION .&extensionId ( { ' ...
    extType    MAP-EXTENSION .&ExtensionType ( { ' ...
      ( @extId ) ) OPTIONAL } OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
    ... ) OPTIONAL},
MAP-OpenInfo ::= SEQUENCE {
  destinationReference [0] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) OPTIONAL,
  originationReference [1] IMPLICIT OCTET STRING ( SIZE (1 .. 20 ) ) OPTIONAL,
...
extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF
  SEQUENCE {
    extId      MAP-EXTENSION .&extensionId ( { ' ...
    extType    MAP-EXTENSION .&ExtensionType ( { ' ...
      ( @extId ) ) OPTIONAL } OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
    ... ) OPTIONAL}
extensionContainer SEQUENCE {
  privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF SEQUENCE {
    extId MAP-EXTENSION .&extensionId ( { ,
      .... } ) ,
    extType MAP-EXTENSION .&ExtensionType ( { ,
      .... } { @extId } ) OPTIONAL) OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE ( .... ) OPTIONAL,
    ... ) OPTIONAL}

MAP-CloseInfo ::= SEQUENCE {
  ... ,
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF SEQUENCE {
      extId MAP-EXTENSION .&extensionId ( { ,
        .... } ) ,
      extType MAP-EXTENSION .&ExtensionType ( { ,
        .... } { @extId } ) OPTIONAL) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE ( .... ) OPTIONAL,
      ... ) OPTIONAL}

MAP-RefuseInfo ::= SEQUENCE {
  reason ENUMERATED {
    noReasonGiven (0 ),
    invalidDestinationReference (1 ),
    invalidOriginatingReference (2 )},
  ... ,
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF SEQUENCE {
      extId MAP-EXTENSION .&extensionId ( { ,
        .... } ) ,
      extType MAP-EXTENSION .&ExtensionType ( { ,
        .... } { @extId } ) OPTIONAL) OPTIONAL,
      pcs-Extensions [1] IMPLICIT SEQUENCE ( .... ) OPTIONAL,
      ... ) OPTIONAL}

Reason ::= ENUMERATED {
  noReasonGiven (0 ),
  invalidDestinationReference (1 ),
  invalidOriginatingReference (2 )}

MAP-UserAbortInfo ::= SEQUENCE {
  map-UserAbortChoice CHOICE {
    userSpecificReason [0] IMPLICIT NULL, 
    userResourceLimitation [1] IMPLICIT NULL, 
    resourceUnavailable [2] IMPLICIT ENUMERATED {
      shortTermResourceLimitation (0 ), 
      longTermResourceLimitation (1 )},
    applicationProcedureCancellation [3] IMPLICIT ENUMERATED {
      handoverCancellation (0 ),
      radioChannelRelease (1 ),
      networkPathRelease (2 ),
      callRelease (3 ),
      associatedProcedureFailure (4 ),
      tandemDialogueRelease (5 ),
      remoteOperationsFailure (6 )}},
  ... ,
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF SEQUENCE {
      extId MAP-EXTENSION .&extensionId ( { ,
        .... } ) ,
      extType MAP-EXTENSION .&ExtensionType ( { ,
        .... } { @extId } ) OPTIONAL) OPTIONAL,
pcs-Extensions [1] IMPLICIT SEQUENCE {
  ... } OPTIONAL,
... } OPTIONAL}

MAP-UserAbortChoice ::= CHOICE {
  userSpecificReason [0] IMPLICIT NULL,
  userResourceLimitation [1] IMPLICIT NULL,
  resourceUnavailable [2] IMPLICIT ENUMERATED {
    shortTermResourceLimitation (0 ),
    longTermResourceLimitation (1 )},
  applicationProcedureCancellation [3] IMPLICIT ENUMERATED {
    handoverCancellation (0 ),
    radioChannelRelease (1 ),
    networkPathRelease (2 ),
    callRelease (3 ),
    associatedProcedureFailure (4 ),
    tandemDialogueRelease (5 ),
    remoteOperationsFailure (6 )}}

ResourceUnavailableReason ::= ENUMERATED {
  shortTermResourceLimitation (0 ),
  longTermResourceLimitation (1 )}

ProcedureCancellationReason ::= ENUMERATED {
  handoverCancellation (0 ),
  radioChannelRelease (1 ),
  networkPathRelease (2 ),
  callRelease (3 ),
  associatedProcedureFailure (4 ),
  tandemDialogueRelease (5 ),
  remoteOperationsFailure (6 )}

MAP-ProviderAbortInfo ::= SEQUENCE {
  map-ProviderAbortReason ENUMERATED {
    abnormalDialogue (0 ),
    invalidPDU (1 )},
...,
  extensionContainer SEQUENCE {
    privateExtensionList [0] IMPLICIT SEQUENCE SIZE (1 .. 10 ) OF SEQUENCE {
      extId MAP-EXTENSION .&extensionId ( {
        ...
      }),
      extType MAP-EXTENSION .&ExtensionType ( {
        ...
      } [extId ] } OPTIONAL) OPTIONAL,
    pcs-Extensions [1] IMPLICIT SEQUENCE {
      ... } OPTIONAL,
    ... } OPTIONAL}

MAP-ProviderAbortReason ::= ENUMERATED {
  abnormalDialogue (0 ),
  invalidPDU (1 )}

END
Annex C
Void
Annex D (informative):
Change Request History

<table>
<thead>
<tr>
<th>CR</th>
<th>SMG#</th>
<th>VERS</th>
<th>NEW_VERS</th>
<th>SUBJECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A018</td>
<td>S15</td>
<td>5.0.0</td>
<td></td>
<td>Export of the data type SubscriberData</td>
</tr>
<tr>
<td>A021</td>
<td>S15</td>
<td>5.0.0</td>
<td></td>
<td>Operations with empty result in the expanded ASN.1 source</td>
</tr>
<tr>
<td>A022</td>
<td>S15</td>
<td>5.0.0</td>
<td></td>
<td>Missing invoke Id in the service descriptions</td>
</tr>
<tr>
<td>A029</td>
<td>S16</td>
<td>5.0.0</td>
<td></td>
<td>Correction of MM-Release in SM_Alert_MSC</td>
</tr>
<tr>
<td>A019</td>
<td>S16</td>
<td>5.0.0</td>
<td></td>
<td>MSISDN parameter of the MAP_PROVIDE_ROAMING_NUMBER</td>
</tr>
<tr>
<td>A024</td>
<td>S16</td>
<td>5.0.0</td>
<td></td>
<td>Modification of the upper limit of Private tags numbers</td>
</tr>
<tr>
<td>A016</td>
<td>S16</td>
<td>5.0.0</td>
<td></td>
<td>Use of SCCP class zero</td>
</tr>
<tr>
<td>A025</td>
<td>S16</td>
<td>5.0.0</td>
<td></td>
<td>Editorial corrections to the USSD SDLs</td>
</tr>
<tr>
<td>A017</td>
<td>S16</td>
<td>5.0.0</td>
<td></td>
<td>Use of the SCCO return option</td>
</tr>
<tr>
<td>A035</td>
<td>S17</td>
<td>5.0.0</td>
<td></td>
<td>From &quot;call restrictions&quot; to &quot;call barring&quot;</td>
</tr>
<tr>
<td>A036</td>
<td>S18</td>
<td>5.0.0</td>
<td></td>
<td>Subscriber tracing related corrections and alignment to GSM 12.08</td>
</tr>
<tr>
<td>A034</td>
<td>S17</td>
<td>5.1.0</td>
<td></td>
<td>Not used value for LMSI in the Cancel_Location service</td>
</tr>
<tr>
<td>A038</td>
<td>S17</td>
<td>5.1.0</td>
<td>5.2.0</td>
<td>Introduction of ECT SS code</td>
</tr>
<tr>
<td>A038</td>
<td>S18</td>
<td>5.1.0</td>
<td></td>
<td>Introduction of ECT SS code</td>
</tr>
<tr>
<td>A043</td>
<td>S19</td>
<td>5.2.0</td>
<td></td>
<td>Timer for PrepareHandover operation</td>
</tr>
<tr>
<td>A044</td>
<td>S19</td>
<td>5.2.0</td>
<td>5.3.0</td>
<td>Exporting of data-types used in GSM12 series specifications</td>
</tr>
<tr>
<td>A045</td>
<td>S19</td>
<td>5.2.0</td>
<td>5.3.0</td>
<td>Checking of MSISDN in HLR for MT-SMG procedure</td>
</tr>
<tr>
<td>A046</td>
<td>S20</td>
<td>5.3.0</td>
<td>5.4.0</td>
<td>Timer values for mobile originated USSD</td>
</tr>
<tr>
<td>A049</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>Indication of MAP.IMSI_ATTACH (phase 2+)</td>
</tr>
<tr>
<td>A051</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>VBS, VGCS</td>
</tr>
<tr>
<td>A061</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>Modification of MAP due to CAMEL ph 1</td>
</tr>
<tr>
<td>A054</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>Transfer of SS codes from HLR to GMSC</td>
</tr>
<tr>
<td>A057</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>HSCSD phase 2+</td>
</tr>
<tr>
<td>A052</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>VGS/VGCS Routing</td>
</tr>
<tr>
<td>A051</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>Transfer of new subscription data for the ASCI eMLPP supplementary service</td>
</tr>
<tr>
<td>A056</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>Changes to MAP for Optimal Routing</td>
</tr>
<tr>
<td>A059</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>Setting of the non-confirmed by HLR-flag (phase 2+)</td>
</tr>
<tr>
<td>A055</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>Alignment with GSM 03.18</td>
</tr>
<tr>
<td>A053</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP extensibility (chapter 14 ASN.1)</td>
</tr>
<tr>
<td>A063</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A064</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A051</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A056</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A059</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A055</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A053</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A063</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A064</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A051</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A056</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A059</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A055</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A053</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A063</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A064</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A067</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A066</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A048</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A068</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A062</td>
<td>s21</td>
<td>5.4.0</td>
<td>5.5.0</td>
<td>MAP v3 Fallback</td>
</tr>
<tr>
<td>A085</td>
<td>s22</td>
<td>5.5.0</td>
<td>5.6.0</td>
<td>SCCP addressing for MAP</td>
</tr>
<tr>
<td>A081</td>
<td>s22</td>
<td>5.5.0</td>
<td>5.6.0</td>
<td>SCCP addressing for MAP</td>
</tr>
<tr>
<td>A083</td>
<td>s22</td>
<td>5.5.0</td>
<td>5.6.0</td>
<td>Clarify handling of result of ReportSMDeliveryStatus</td>
</tr>
<tr>
<td>A086</td>
<td>s22</td>
<td>5.5.0</td>
<td>5.6.0</td>
<td>Correction of editing error</td>
</tr>
<tr>
<td>A082</td>
<td>s22</td>
<td>5.5.0</td>
<td>5.6.0</td>
<td>Modifications due to SMS inconsistencies with GSM 03.40</td>
</tr>
<tr>
<td>A080</td>
<td>s22</td>
<td>5.5.0</td>
<td>5.6.0</td>
<td>Using TC-Result-NL to segment the SRI-ACK</td>
</tr>
<tr>
<td>A074</td>
<td>s22</td>
<td>5.5.0</td>
<td>5.6.0</td>
<td>Any Time Interrogation process description in HLR</td>
</tr>
<tr>
<td>A072</td>
<td>s22</td>
<td>5.5.0</td>
<td>5.6.0</td>
<td>Handling of Unknown Operation</td>
</tr>
<tr>
<td>A073</td>
<td>s22</td>
<td>5.5.0</td>
<td>5.6.0</td>
<td>Editorial clarification of handling of parameters for optional features</td>
</tr>
</tbody>
</table>

(continued)
## CR | SMG# | VERS | NEW VERS | SUBJECT
--- | --- | --- | --- | ---
A075 | s22 | 5.5.0 | 5.6.0 | Alignment with CAMEL specs: 03.78 and 09.78
A077 | s22 | 5.5.0 | 5.6.0 | Transmission of gsmSCF-Address to HLR
A078 | s22 | 5.5.0 | 5.6.0 | New Maximum value for Age of Loc. Info
A079 | s22 | 5.5.0 | 5.6.0 | Editorial correction of AC table
A092 | s23 | 5.5.0 | 5.6.0 | ASCl ph1 option pruning (routing or VBS/VGCS calls) [R96]
A104 | s23 | 5.6.0 | 5.7.0 | ASN.1 definition of private extension
A090 | s23 | 5.5.0 | 5.6.0 | Re-introducing of BindingPackage-v1 to GSM 09.02 [R96]
A093 | s23 | 5.6.0 | 5.7.0 | Transmission of gsmSCF-Address to HLR
A078 | s23 | 5.6.0 | 5.7.0 | New Maximum value for Age of Loc. Info
A079 | s23 | 5.6.0 | 5.7.0 | Editorial correction of AC table
A092 | s23 | 5.6.0 | 5.7.0 | ASCl ph1 option pruning (routing or VBS/VGCS calls) [R96]
A104 | s23 | 5.7.0 | 5.8.0 | ASN.1 definition of private extension
A090 | s23 | 5.7.0 | 5.8.0 | Re-introducing of BindingPackage-v1 to GSM 09.02 [R96]
A093 | s23 | 5.7.0 | 5.8.0 | Transmission of gsmSCF-Address to HLR
A078 | s23 | 5.7.0 | 5.8.0 | New Maximum value for Age of Loc. Info
A079 | s23 | 5.7.0 | 5.8.0 | Editorial correction of AC table
A092 | s23 | 5.7.0 | 5.8.0 | ASCl ph1 option pruning (routing or VBS/VGCS calls) [R96]
A104 | s23 | 5.8.0 | 5.9.0 | ASN.1 definition of private extension
A090 | s23 | 5.8.0 | 5.9.0 | Re-introducing of BindingPackage-v1 to GSM 09.02 [R96]
A093 | s23 | 5.8.0 | 5.9.0 | Transmission of gsmSCF-Address to HLR
A078 | s23 | 5.8.0 | 5.9.0 | New Maximum value for Age of Loc. Info
A079 | s23 | 5.8.0 | 5.9.0 | Editorial correction of AC table
A092 | s23 | 5.8.0 | 5.9.0 | ASCl ph1 option pruning (routing or VBS/VGCS calls) [R96]
A104 | s23 | 5.9.0 | 5.10.0 | ASN.1 definition of private extension
A090 | s23 | 5.9.0 | 5.10.0 | Re-introducing of BindingPackage-v1 to GSM 09.02 [R96]
A093 | s23 | 5.9.0 | 5.10.0 | Transmission of gsmSCF-Address to HLR
A078 | s23 | 5.9.0 | 5.10.0 | New Maximum value for Age of Loc. Info
A079 | s23 | 5.9.0 | 5.10.0 | Editorial correction of AC table
A092 | s23 | 5.9.0 | 5.10.0 | ASCl ph1 option pruning (routing or VBS/VGCS calls) [R96]
A104 | s23 | 5.10.0 | 5.11.0 | ASN.1 definition of private extension
A090 | s23 | 5.10.0 | 5.11.0 | Re-introducing of BindingPackage-v1 to GSM 09.02 [R96]
A093 | s23 | 5.10.0 | 5.11.0 | Transmission of gsmSCF-Address to HLR
A078 | s23 | 5.10.0 | 5.11.0 | New Maximum value for Age of Loc. Info
A079 | s23 | 5.10.0 | 5.11.0 | Editorial correction of AC table
A092 | s23 | 5.10.0 | 5.11.0 | ASCl ph1 option pruning (routing or VBS/VGCS calls) [R96]
A104 | s23 | 5.11.0 | 5.12.0 | ASN.1 definition of private extension
A090 | s23 | 5.11.0 | 5.12.0 | Re-introducing of BindingPackage-v1 to GSM 09.02 [R96]
A093 | s23 | 5.11.0 | 5.12.0 | Transmission of gsmSCF-Address to HLR
A078 | s23 | 5.11.0 | 5.12.0 | New Maximum value for Age of Loc. Info
A079 | s23 | 5.11.0 | 5.12.0 | Editorial correction of AC table
A092 | s23 | 5.11.0 | 5.12.0 | ASCl ph1 option pruning (routing or VBS/VGCS calls) [R96]
A104 | s23 | 5.12.0 | 5.13.0 | ASN.1 definition of private extension
A090 | s23 | 5.12.0 | 5.13.0 | Re-introducing of BindingPackage-v1 to GSM 09.02 [R96]
A093 | s23 | 5.12.0 | 5.13.0 | Transmission of gsmSCF-Address to HLR
A078 | s23 | 5.12.0 | 5.13.0 | New Maximum value for Age of Loc. Info
A079 | s23 | 5.12.0 | 5.13.0 | Editorial correction of AC table
A092 | s23 | 5.12.0 | 5.13.0 | ASCl ph1 option pruning (routing or VBS/VGCS calls) [R96]
A104 | s23 | 5.13.0 | 5.14.0 | ASN.1 definition of private extension
A090 | s23 | 5.13.0 | 5.14.0 | Re-introducing of BindingPackage-v1 to GSM 09.02 [R96]
A093 | s23 | 5.13.0 | 5.14.0 | Transmission of gsmSCF-Address to HLR
## History

### Document history

<table>
<thead>
<tr>
<th>Edition</th>
<th>Date</th>
<th>Action</th>
<th>Procedure</th>
<th>Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>V5.3.0</td>
<td>August 1996</td>
<td>Publication as GSM 09.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edition 1</td>
<td>May 1997</td>
<td>Publication as ETS 300 974</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edition 3</td>
<td>August 1997</td>
<td>One-step Approval Procedure</td>
<td>OAP 9752:</td>
<td>1997-08-29 to 1997-12-26</td>
</tr>
<tr>
<td>Edition 2</td>
<td>October 1997</td>
<td>Publication as ETS 300 974</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edition 3</td>
<td>January 1998</td>
<td>Publication as ETS 300 974</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edition 4</td>
<td>July 1998</td>
<td>Publication as ETS 300 974</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edition 5</td>
<td>December 1998</td>
<td>Publication as ETS 300 974</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edition 6</td>
<td>January 1999</td>
<td>Publication as ETS 300 974</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edition 7</td>
<td>March 1999</td>
<td>Publication as ETS 300 974</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edition 8</td>
<td>August 1999</td>
<td>Publication as ETS 300 974</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edition 9</td>
<td>December 1999</td>
<td>Publication as ETS 300 974</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edition 10</td>
<td>April 2000</td>
<td>Publication as ETS 300 974</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V5.16.0</td>
<td>September 2000</td>
<td>Publication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edition 11</td>
<td>December 2000</td>
<td>Publication as ETS 300 974</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V5.17.0</td>
<td>March 2001</td>
<td>Publication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V5.18.0</td>
<td>March 2003</td>
<td>Publication</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>