



**E**UROPEAN  
**T**ELECOMMUNICATION  
**S**TANDARD

**DRAFT**  
pr **ETS 300 950**

March 1999

Fourth Edition

---

Source: SMG

Reference: RE/SMG-030480QR3

ICS: 33.020

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



**Digital cellular telecommunications system (Phase 2+);  
Mobile radio interface layer 3  
supplementary services specification;  
Formats and coding  
(GSM 04.80 version 5.3.0 Release 1996)**

**ETSI**

European Telecommunications Standards Institute

**ETSI Secretariat**

**Postal address:** F-06921 Sophia Antipolis CEDEX - FRANCE

**Office address:** 650 Route des Lucioles - Sophia Antipolis - Valbonne - FRANCE

**Internet:** [secretariat@etsi.fr](mailto:secretariat@etsi.fr) - <http://www.etsi.org>

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

---

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

© European Telecommunications Standards Institute 1999. All rights reserved.



## Contents

Intellectual Property Rights.....	7
Foreword .....	7
1 Scope .....	9
1.1 Normative references .....	9
1.2 Abbreviations .....	10
2 Message functional definitions and contents .....	11
2.1 General .....	11
2.2 Messages for supplementary services control.....	12
2.3 Facility.....	12
2.4 Register.....	13
2.4.1 Register (network to MS direction).....	13
2.4.2 Register (MS to network direction).....	13
2.4.2.1 SS version .....	14
2.5 Release complete .....	14
2.5.1 Cause .....	14
2.5.2 Facility .....	14
3 General message format and information elements coding .....	15
3.1 Overview .....	15
3.2 Protocol discriminator .....	15
3.3 Transaction identifier .....	15
3.4 Message type.....	15
3.5 Other information elements .....	16
3.6 Facility information element .....	16
3.6.1 Component (octet 3 etc.).....	17
3.6.2 Component type tag .....	19
3.6.3 Component ID tag .....	19
3.6.4 Operation Code .....	20
3.6.5 Sequence and Set tags .....	20
3.6.6 Error Code.....	20
3.6.7 Problem Code .....	20
3.7 Version handling for supplementary services .....	22
3.7.1 Supplementary service screening indicator.....	22
3.7.2 Supplementary service version indicator.....	22
4 Supplementary services operation specifications .....	23
4.1 General .....	23
4.2 Operation types.....	24
4.2.1 [spare] .....	26
4.2.2 Operation types description.....	26
4.2.2.1 RegisterSS (MS --> network).....	26
4.2.2.2 EraseSS (MS --> network).....	26
4.2.2.3 ActivateSS (MS --> network) .....	26
4.2.2.4 DeactivateSS (MS --> network) .....	26
4.2.2.5 InterrogateSS (MS --> network).....	26
4.2.2.6 NotifySS (network --> MS) .....	26
4.2.2.7 RegisterPassword (MS --> network).....	26
4.2.2.8 GetPassword (network --> MS) .....	26
4.2.2.9 ProcessUnstructuredSS-Data (MS --> network) .....	26
4.2.2.10 ProcessUnstructuredSS-Request (MS --> network).....	27
4.2.2.11 UnstructuredSS-Request (network --> MS).....	27
4.2.2.12 UnstructuredSS-Notify (network --> MS) .....	27
4.2.2.13 ForwardCheckSSIndication (network --> MS) .....	27
4.2.2.14 ForwardChargeAdvice (network --> MS).....	27
4.2.2.15 BuildMPTY (MS --> network).....	27
4.2.2.16 HoldMPTY (MS --> network) .....	27

	4.2.2.17	RetrieveMPTY (MS --> network).....	27
	4.2.2.18	SplitMPTY (MS --> network) .....	27
	4.2.2.19	ForwardCUG-Info (MS --> network).....	27
	4.2.2.20	ExplicitCT (MS --> Network) .....	27
4.3	Error types.....		28
4.3.1	Error types ASN.1 specification .....		28
4.3.2	Error types description .....		28
	4.3.2.1	UnknownSubscriber .....	28
	4.3.2.2	BearerServiceNotProvisioned .....	28
	4.3.2.3	TeleServiceNotProvisioned .....	28
	4.3.2.4	IllegalSS-Operation .....	29
	4.3.2.5	SS-ErrorStatus .....	29
	4.3.2.6	SS-NotAvailable .....	29
	4.3.2.7	SS-SubscriptionViolation .....	29
	4.3.2.8	SS-Incompatibility .....	29
	4.3.2.9	SystemFailure .....	29
	4.3.2.10	DataMissing .....	29
	4.3.2.11	UnexpectedDataValue .....	29
	4.3.2.12	PasswordRegistrationFailure .....	29
	4.3.2.13	NegativePasswordCheck .....	30
	4.3.2.14	FacilityNotSupported .....	30
	4.3.2.15	ResourcesNotAvailable .....	30
	4.3.2.16	MaxNumberOfMPTY-ParticipantsExceeded .....	30
	4.3.2.17	CallBarred .....	30
	4.3.2.18	NumberOfPW-AttemptsViolation .....	30
	4.3.2.19	AbsentSubscriber .....	30
	4.3.2.20	IllegalSubscriber .....	30
	4.3.2.21	IllegalEquipment .....	30
	4.3.2.22	USSD-Busy .....	30
	4.3.2.23	UnknownAlphabet .....	30
4.4	Data types and identifiers.....		31
4.4.1	General .....		31
4.4.2	ASN.1 data types .....		31
4.4.3	Identifiers definition .....		33
	4.4.3.1	chargingInformation .....	33
	4.4.3.2	e1 .....	33
	4.4.3.3	e2 .....	33
	4.4.3.4	e3 .....	33
	4.4.3.5	e4 .....	33
	4.4.3.6	e5 .....	33
	4.4.3.7	e6 .....	33
	4.4.3.8	e7 .....	33
	4.4.3.9	ss-Code .....	33
	4.4.3.10	ss-Notification .....	33
	4.4.3.11	ss-Status .....	33
	4.4.3.12	callsWaiting-Indicator .....	34
	4.4.3.13	callOnhold-Indicator .....	34
	4.4.3.14	mpty-Indicator .....	34
	4.4.3.15	forwardCUG-InfoArg .....	34
	4.4.3.16	cug-Index .....	34
	4.4.3.17	suppressPrefCUG .....	34
	4.4.3.18	suppressOA .....	34
	4.4.3.19	clirSuppressionRejected .....	34
	4.4.3.20	ect-Indicator .....	34
	4.4.3.21	ect-CallState .....	34
	4.4.3.22	rdn .....	34
	4.4.3.23	presentationAllowedAddress .....	34
	4.4.3.24	presentationRestricted .....	34
	4.4.3.25	numberNotAvailableDueToInterworking .....	35
	4.4.3.26	presentationRestrictedAddress .....	35
	4.4.3.27	partyNumber .....	35
	4.4.3.28	partyNumberSubaddress .....	35
4.5	Operations and errors implementation.....		36

Annex A (informative): Expanded ASN.1 Module "SS-Protocol" ..... 38

Annex B (informative): Change Request History ..... 48

History..... 50

Blank page

## 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 SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available **free of charge** from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://www.etsi.org/ipr>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

## Foreword

This draft European Telecommunication Standard (ETS) has been produced by the Special Mobile Group (SMG) Technical Committee of the European Telecommunications Standards Institute (ETSI), and is now submitted for the One-step Approval Procedure phase of the ETSI standards approval procedure.

The present document defines the coding of information necessary for support of supplementary service operation on the mobile radio interface layer 3 within the digital cellular telecommunications system.

The specification from which the present document has been derived was originally based on CEPT documentation, hence the presentation of the present document may not be entirely in accordance with the ETSI/PNE rules.

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

Version 5.x.y

where:

- 5 indicates Release 1997 of GSM Phase 2+;
- x the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.;
- y the third digit is incremented when editorial only changes have been incorporated in the specification.

<b>Proposed transposition dates</b>	
Date of latest announcement of this ETS (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this ETS (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	6 months after doa

Blank page



## 1 Scope

This ETS contains the coding of information necessary for support of supplementary service operation on the mobile radio interface layer 3.

Clause 2 gives the functional definitions and contents of messages for call independent supplementary service operations. Messages necessary for support of call related supplementary service operations are defined in GSM 04.08.

Clause 3 gives the general format and coding for messages used for call independent supplementary service and the format and coding of information elements used for both call related and call independent supplementary service operations.

Clause 4 gives the specification of the call related and call independent supplementary service operations.

### 1.1 Normative references

This ETS incorporates by dated and undated references, 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 (ETR 350): "Digital cellular telecommunications system (Phase 2+); Abbreviations and acronyms".
- [2] GSM 02.24 (ETS 300 923): "Digital cellular telecommunications system (Phase 2+); Description of Charge Advice Information (CAI)".
- [3] GSM 04.06 (ETS 300 938): "Digital cellular telecommunications system (Phase 2+); Mobile Station - Base Station System (MS - BSS) interface Data Link (DL) layer specification".
- [4] GSM 04.07 (ETS 300 939): "Digital cellular telecommunications system (Phase 2+); Mobile radio interface signalling layer 3; General aspects".
- [5] GSM 04.08 (ETS 300 940): "Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 specification".
- [6] GSM 04.10 (ETS 300 941): "Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3; Supplementary services specification; General aspects".
- [7] GSM 04.80 (ETS 300 950): "Digital cellular telecommunications system (Phase 2+); Mobile radio interface layer 3 supplementary services specification; Formats and coding".
- [8] GSM 04.81 (ETS 300 951): "Digital cellular telecommunications system (Phase 2+); Line identification supplementary services - Stage 3".
- [9] GSM 04.82 (ETS 300 952): "Digital cellular telecommunications system (Phase 2+); Call Forwarding (CF) supplementary services - Stage 3".
- [10] GSM 04.83 (ETS 300 953): "Digital cellular telecommunications system (Phase 2+); Call Waiting (CW) and Call Hold (HOLD) supplementary services - Stage 3".
- [11] GSM 04.84 (ETS 300 954): "Digital cellular telecommunications system (Phase 2+); MultiParty (MPTY) supplementary services - Stage 3".
- [12] GSM 04.85: "Digital cellular telecommunications system (Phase 2+); Closed User Group (CUG) supplementary services - Stage 3".

- [13] GSM 04.86 (ETS 300 955): "Digital cellular telecommunications system (Phase 2+); Advice of Charge (AoC) supplementary services - Stage 3".
- [14] GSM 04.88 (ETS 300 956): "Digital cellular telecommunications system (Phase 2+); Call Barring (CB) supplementary services - Stage 3".
- [15] GSM 04.90 (ETS 300 957): "Digital cellular telecommunications system (Phase 2+); Unstructured supplementary services operation - Stage 3".
- [16] GSM 09.02 (ETS 300 974): "Digital cellular telecommunications system (Phase 2+); Mobile Application Part (MAP) specification".
- [17] GSM 09.11: "Digital cellular telecommunications system (Phase 2+); Signalling interworking for supplementary services".
- [18] CCITT Recommendation X.208: "Specification of basic encoding rules for Abstract Syntax Notation One (ASN.1)".
- [19] CCITT Recommendation X.209: "Specification of Abstract Syntax Notation One (ASN.1)".
- [20] CCITT Recommendation Q.773: "Transaction capabilities formats and encoding".

## 1.2 Abbreviations

Abbreviations used in this specification are listed in GSM 01.04.

## 2 Message functional definitions and contents

### 2.1 General

This clause defines the structure of the messages of the layer 3 protocol defined in GSM 04.80. These messages are standard L3 messages as defined in GSM 04.07.

Each definition includes:

- a) a brief description of the message;
- b) a table listing the information elements in the order of their appearance in the message. In a sequence of consecutive IEs with half octet length, the first IE occupies bits 1 to 4 of octet N, the second bits 5 to 8 of octet N, the third bits 1 to 4 of octet N+1 etc.;

For each IE the table indicates:

- 1) the information element identifier, in hexadecimal notation, if the IE has format T, TV or TLV. If the IEI has half octet length, it is specified by a notation representing the IEI as a hexadecimal digit followed by a "-" (example: B-);
  - 2) the name of the IE (which gives an idea of the semantics of the element), which is used in this and other specifications as a reference to the IE within the message;
  - 3) the name of the type of the IE (which indicates the coding of the value part of the IE), and a reference to a description of the value part of the IE;
  - 4) the presence requirement indication (M, C or O) for the IE, as defined in GSM 04.07;
  - 5) the format of the IE (T, V, TV, LV, TLV) as defined in GSM 04.07;
  - 6) the length of the IE (or permissible range of lengths), in octets, in the message, where "?" means that the maximum length of the IE is only constrained by the link layer protocol, and in the case of the facility IE by possible further considerations specified in GSM 04.10. This indication is non-normative.
- c) subclauses specifying conditions for IEs with presence requirement C or O in the relevant message. Together with other conditions specified in GSM 04.80, GSM 04.10 or GSM 04.8x and 04.9x-series this defines when the IE shall be included or not, what non-presence of such IEs means, and (for IEs with presence requirement C) the static conditions for presence and/or non-presence of the IEs (see GSM 04.07).

## 2.2 Messages for supplementary services control

Table 2.1 summarizes the messages for call independent supplementary services control (see GSM 04.10 for a detailed description of call independent supplementary service messages).

**Table 2.1: Messages for call independent supplementary service control**

Messages for supplementary service control	Reference
FACILITY	2.3
REGISTER	2.4
RELEASE COMPLETE	2.5

## 2.3 Facility

This message is sent by the mobile station or the network to request or acknowledge a supplementary service. It is used when information is to be conveyed and the transaction already exists, but is not to be released. The supplementary service to be invoked, and its associated parameters, are specified in the Facility information element (see table 2.2).

**Table 2.2: FACILITY message content**

IEI	Information element	Type / Reference	Presence	Format	Length
	Supplementary service protocol discriminator	Protocol discriminator 3.2	M	V	1/2
	Transaction identifier	Transaction identifier 3.3	M	V	1/2
	Facility message type	Message type 3.4	M	V	1
	Facility	Facility 3.5	M	LV	2-?

## 2.4 Register

### 2.4.1 Register (network to MS direction)

This message is sent by the network to the mobile station to assign a new transaction identifier for call independent supplementary service control and to request or acknowledge a supplementary service (see table 2.3).

**Table 2.3: REGISTER message content (network to MS direction)**

IEI	Information element	Type / Reference	Presence	Format	Length
	Supplementary service protocol discriminator	Protocol discriminator 3.2	M	V	1/2
	Transaction identifier	Transaction identifier 3.3	M	V	1/2
	Register message type	Message type 3.4	M	V	1
1C	Facility	Facility 3.5	M	TLV	2-?

### 2.4.2 Register (MS to network direction)

This message is sent by the mobile station to the network to assign a new transaction identifier for call independent supplementary service control and to request or acknowledge a supplementary service (see table 2.4).

**Table 2.4: REGISTER message content (MS to network direction)**

IEI	Information element	Type / Reference	Presence	Format	Length
	Supplementary service protocol discriminator	Protocol discriminator 3.2	M	V	1/2
	Transaction identifier	Transaction identifier 3.3	M	V	1/2
	Register message type	Message type 3.4	M	V	1
1C	Facility	Facility 3.5	M	TLV	2-?
7F	SS version	SS version indicator 3.8.2	O	TLV	3

**2.4.2.1 SS version**

This information element shall be included if the supplementary service operation being invoked is implemented according to the phase 2 GSM standards.

**2.5 Release complete**

This message is sent by the mobile station or the network to release a transaction used for call independent supplementary service control. It may also request or acknowledge a supplementary service (see table 2.5).

**Table 2.5: RELEASE COMPLETE message content**

IEI	Information element	Type / Reference	Presence	Format	Length
	Supplementary service protocol discriminator	Protocol discriminator 3.2	M	V	1/2
	Transaction identifier	Transaction identifier 3.3	M	V	1/2
	Release Complete message type	Message type 3.4	M	V	1
08	Cause	Cause GSM 04.08	O	TLV	4-32
1C	Facility	Facility 3.5	O	TLV	2-?

**2.5.1 Cause**

This information element shall be included when the functional handling of the Cause IE is specified in the service description or GSM 09.11. If the functional handling of the Cause IE is not specified, the receiving entity may ignore the IE.

**2.5.2 Facility**

This information element shall be included as required by the service description and the procedures defined in GSM 04.10.

### 3 General message format and information elements coding

The figures and text in this clause describe message contents. Within each octet, the bit designated "bit 1" is transmitted first, followed by bits 2, 3, 4, etc. Similarly, the octet shown at the top of each figure is sent first.

#### 3.1 Overview

Within the layer 3 protocol defined in GSM 04.80, every message is a standard L3 message as defined in GSM 04.07. This means that the message consists of the following parts:

- a) protocol discriminator;
- b) transaction identifier;
- c) message type;
- d) other information elements, as required.

Unless specified otherwise, a particular information element may be present only once in a given message.

When a field extends over more than one octet, the order of bit values progressively decreases as the octet number increases. The least significant bit of the field is represented by the lowest numbered bit of the highest numbered octet of the field.

#### 3.2 Protocol discriminator

The Protocol Discriminator (PD) and its use are defined in GSM 04.07. GSM 04.80 defines the protocols relating to the PD values:

1 0 1 1      supplementary services (call independent).

#### 3.3 Transaction identifier

For general rules, format and coding of transaction identifier values, see GSM 04.08.

#### 3.4 Message type

The message type IE and its use are defined in GSM 04.07. Table 3.1 defines the value part of the message type IE used in the supplementary service protocol.

**Table 2.5: RELEASE COMPLETE message content**

8	7	6	5	4	3	2	1	Message types
0	x	1	0	.	.	.	.	Clearing messages: - RELEASE COMPLETE
				1	0	1	0	
0	x	1	1	.	.	.	.	Miscellaneous message group: - FACILITY - REGISTER
				1	0	1	0	
				1	0	1	1	
NOTE 1: Bit 8 is reserved for possible future use as an extension bit, see GSM 04.07.								
NOTE 2: Bit 7 is reserved for the send sequence number in messages sent from the mobile station. In messages sent from the network, bit 7 is coded with a "0", see GSM 04.07.								

**3.5 Other information elements**

These information elements are coded according to the general coding rules as defined in GSM 04.08.

Table 3.2 contains the code-points allocated to the information elements used in messages defined in this specification. All IEs are defined in GSM 04.08, but the content of the Facility and SS version indicator IEs are defined within this specification.

**Table 3.2: Information elements specific to call independent supplementary service control**

8 7 6 5 4 3 2 1		<b>Reference (IE content)</b>
0 . . . . .	Type 3 and 4 information elements	
0 0 0 1 0 0 0	Cause	GSM 04.08
0 0 1 1 1 0 0	Facility	3.6
1 1 1 1 1 1 1	SS version indicator	3.8.2

**3.6 Facility information element**

The purpose of the Facility information element is to indicate the invocation and operation of supplementary services, identified by the corresponding operation code within the Facility information element.

The Facility information element is coded as shown in figure 3.1 and tables 3.3 to 3.17.

The Facility is a type 4 information element with no upper length limit except that given by the maximum number of octets in a L3 message, see GSM 04.06.

8	7	6	5	4	3	2	1	
0	0	0	1	1	1	0	0	octet 1
Facility IEI								
Length of Facility contents								octet 2
Component(s) (note)								octet 3 etc.

NOTE: One or more components may be included depending on specific service requirements.

**Figure 3.1: Facility information element**



### 3.6.1 Component (octet 3 etc.)

This subclause provides the formats and encoding of components in the Facility information element. Formats and encoding methods make use of and is a subset of CCITT Recommendation Q.773 (Transaction Capabilities formats and Encoding) and T/S 43/BB. The used part of CCITT Recommendation Q.773 respectively T/S 43/BB is almost the same as the Component Portion of TC messages. The only difference is that returnResultNotLast is not used.

This subclause is further based on:

- CCITT Recommendation X.208 (Specification of Abstract Syntax Notation One (ASN.1));
- CCITT Recommendation X.209 (Specification of basic encoding rules for Abstract Syntax Notation One);

and is consistent with these CCITT recommendations.

The CCITT Recommendations X.208 and X.209 formal description language is not used.

The parameters in tables 3.3 to 3.6 may be one of the following:

- a Sequence of Parameters;
- a Set of Parameters;
- a specific Parameter with its own tag (i.e. not part of a Sequence or Set);
- nothing at all (i.e. absent).

NOTE: Concerning the general rules for encoding (structure of encoding, identifier octets, length octets, etc.) see CCITT Recommendations X.208 and X.209. For these general rules the same exceptions apply as stated in GSM 09.02. This holds also for tables 3.3 to 3.6.

**Table 3.3: Invoke component**

Invoke component	Reference	Mandatory indication
Component type tag	3.6.2	M
Component length	X.209	
Invoke ID tag	3.6.3	
Invoke ID length	X.209	M
Invoke ID	3.6.3	
Linked ID tag	3.6.3	
Linked ID length	X.209	O
Linked ID	3.6.3	
Operation Code tag	3.6.4	
Operation Code length	X.209	M
Operation Code	3.6.4	
Parameters	4	O

Table 3.4: Return Result component

Return Result component	Reference	Mandatory indication
Component type tag	3.6.2	M
Component length	X.209	
Invoke ID tag	3.6.3	
Invoke ID length	X.209	M
Invoke ID	3.6.3	
Sequence tag	3.6.5	O (note)
Sequence length	X.209	
Operation Code tag	3.6.4	
Operation Code length	X.209	O (note)
Operation Code	3.6.4	
Parameters	4	O (note)
NOTE: Omitted if the Return Result component does not include any parameters.		

Table 3.5: Return Error component

Return Error component	Reference	Mandatory indication
Component type tag	3.6.2	M
Component length	X.209	
Invoke ID tag	3.6.3	
Invoke ID length	X.209	M
Invoke ID	3.6.3	
Error Code tag	3.6.6	
Error Code length	X.209	M
Error Code	3.6.6	
Parameters	4	O

Table 3.6: Reject component

Reject component	Reference	Mandatory indication
Component type tag	3.6.2	M
Component length	X.209	
Invoke ID tag (note)	3.6.3	
Invoke ID length	X.209	M
Invoke ID	3.6.3	
Problem Code tag	3.6.7	
Problem Code length	X.209	M
Problem Code	3.6.7	
NOTE: If the Invoke ID is not available, Universal Null (table 3.9) with length = 0 shall be used.		

### 3.6.2 Component type tag

The Component type tag is coded context-specific, constructor as indicated in table 3.7.

**Table 3.7: Coding of Component type tag**

Component type tag	8	7	6	5	4	3	2	1
Invoke	1	0	1	0	0	0	0	1
Return Result	1	0	1	0	0	0	1	0
Return Error	1	0	1	0	0	0	1	1
Reject	1	0	1	0	0	1	0	0

### 3.6.3 Component ID tag

The term Component ID refers to the Invoke ID or the Linked ID. The Component ID tag is coded as shown in table 3.8.

**Table 3.8: Coding of Component ID tag**

Component ID tag	8	7	6	5	4	3	2	1
Invoke ID	0	0	0	0	0	0	1	0
Linked ID (note)	1	0	0	0	0	0	0	0
NOTE: This tag differs from the Invoke ID tag, which is coded as a Universal INTEGER, in order to distinguish it from the following tag (Operation Code) which is also coded as a Universal INTEGER.								

The length of a Component ID is 1 octet.

An Invoke Component has one or two Component IDs: an Invoke ID, and if it is desired to associate the Invoke with a previous Invoke, then the Linked ID is provided in addition to the Invoke ID.

Return Result and Return Error Components have one Component ID, called an Invoke ID which is the reflection of the Invoke ID of the Invoke Component to which they are responding.

The Reject Component uses as its Invoke ID, the Invoke ID in the Component being rejected. If this ID is unavailable (e.g. due to mutilation of the message not detected by lower layers), then the Invoke ID tag is replaced with a universal NULL tag as shown in table 3.9. Universal NULL has always length = 0

Any kind of component, except a reject component, may be rejected.

**Table 3.9: Coding of NULL tag**

NULL tag	8	7	6	5	4	3	2	1
NULL tag	0	0	0	0	0	1	0	1

If an Invoke containing both Invoke and Linked IDs is being rejected, only the Invoke ID is used in the Reject Component.

### 3.6.4 Operation Code

Each Operation is assigned an Operation Code to identify it. An Operation Code follows an Operation Code tag and Operation Code length. The Operation Code tag is coded as shown in table 3.10.

**Table 3.10: Coding of Operation Code tag**

	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
Operation Code tag	0	0	0	0	0	0	1	0

The Operation Codes for the different Operations are defined in subclause 4.5.

### 3.6.5 Sequence and Set tags

When there is more than one parameter in a Component (applicable to all Component types), they follow the Sequence or Set tag, which are coded universal, constructor as shown in table 3.11.

**Table 3.11: Coding of Sequence and set tags**

<b>Sequence and set tags</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
Sequence tag	0	0	1	1	0	0	0	0
Set tag	0	0	1	1	0	0	0	1

### 3.6.6 Error Code

Each Error is assigned a value (Error Code) to identify it.

An Error Code follows an Error Code tag and Error Code length. The Error Code tag is coded as shown in table 3.12.

**Table 3.12: Coding of Error Code tag**

	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
Error Code tag	0	0	0	0	0	0	1	0

The Error Codes for the different Errors are defined in subclause 4.5.

### 3.6.7 Problem Code

The Problem Code consists of one of the four elements: General Problem, Invoke Problem, Return Result Problem or Return Error Problem. The tags for these elements are coded as shown in table 3.13.

**Table 3.13: Coding of Problem tags**

<b>Problem tags</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
General Problem tag	1	0	0	0	0	0	0	0
Invoke Problem tag	1	0	0	0	0	0	0	1
Return Result Problem tag	1	0	0	0	0	0	1	0
Return Error Problem tag	1	0	0	0	0	0	1	1

The Problem Codes for the different Problems are shown in tables 3.14 to 3.17.

**Table 3.14: Coding of General Problem Codes**

<b>General Problem Codes</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
Unrecognized Component	0	0	0	0	0	0	0	0
Mistyped Component	0	0	0	0	0	0	0	1
Badly Structured Component	0	0	0	0	0	0	1	0

**Table 3.15: Coding of Invoke Problem Codes**

<b>Invoke Problem Codes</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
Duplicate Invoke ID	0	0	0	0	0	0	0	0
Unrecognized Operation	0	0	0	0	0	0	0	1
Mistyped Parameter	0	0	0	0	0	0	1	0
Resource Limitation	0	0	0	0	0	0	1	1
Initiating Release	0	0	0	0	0	1	0	0
Unrecognized Linked ID	0	0	0	0	0	1	0	1
Linked Response Unexpected	0	0	0	0	0	1	1	0
Unexpected Linked Operation	0	0	0	0	0	1	1	1

**Table 3.16: Coding of Return Result Problem Codes**

<b>Return Result Problem Codes</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
Unrecognized Invoke ID	0	0	0	0	0	0	0	0
Return Result Unexpected	0	0	0	0	0	0	0	1
Mistyped Parameter	0	0	0	0	0	0	1	0

**Table 3.17: Coding of Return Error Problem Codes**

<b>Return Error Problem Codes</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
Unrecognized Invoke ID	0	0	0	0	0	0	0	0
Return Error Unexpected	0	0	0	0	0	0	0	1
Unrecognized Error	0	0	0	0	0	0	1	0
Unexpected Error	0	0	0	0	0	0	1	1
Mistyped Parameter	0	0	0	0	0	1	0	0

### 3.7 Version handling for supplementary services

#### 3.7.1 Supplementary service screening indicator

The purpose of the supplementary service screening indicator is to allow the network to assess the capabilities of the MS in advance of a network initiated SS activity. The SS screening indicator is sent in the mobile station classmark 2 as defined in GSM 04.08. The handling of the SS screening indicator is described in GSM 04.10.

<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
(note)	(note)	SS screening indicator		(note)			

NOTE: Values not relevant to supplementary services.

**Figure 3.2: Coding of SS screening indicator in mobile station classmark 2**

**Table 3.18: Coding of SS screening indicator in mobile station classmark 2**

<b>SS screening indicator in mobile station classmark 2</b>	<b>6</b>	<b>5</b>
default value of phase 1	0	0
capability of handling of ellipsis notation and phase 2 error handling (note 1)	0	1
for future use (note 2)	1	0
for future use (note 2)	1	1
NOTE 1: Ellipsis notation is described in GSM 04.10 and GSM 09.02. SS Error handling is described in GSM 04.10.		
NOTE 2: The network shall interpret these values the same as "01".		

#### 3.7.2 Supplementary service version indicator

The purpose of the supplementary service version indicator is to allow the network to select the correct version of a protocol for a specific supplementary service. The SS version indicator is included in messages as defined in GSM 04.08 and GSM 04.80. The coding described in table 3.19 refers to the first octet received in the SS version indicator. Any other octets received shall be ignored. The handling of the SS version indicator is described in GSM 04.10.

**Table 3.19: Coding of SS version indicator**

<b>SS version indicator</b>	<b>8</b>	<b>7</b>	<b>6</b>	<b>5</b>	<b>4</b>	<b>3</b>	<b>2</b>	<b>1</b>
phase 2 service, ellipsis notation, and phase 2 error handling is supported (note 1) all other values are for future use (note 2)	0	0	0	0	0	0	0	0
NOTE 1: Ellipsis notation is described in GSM 04.10 and GSM 09.02. SS Error handling is described in GSM 04.10.								
NOTE 2: The network shall interpret all possible values of the SS version indicator the same as "00000000".								

## 4 Supplementary services operation specifications

### 4.1 General

This clause specifies the abstract syntax for the Supplementary Service protocol using the Abstract Syntax Notation One (ASN.1), defined in CCITT Recommendation X.208 (1998).

The mapping of OPERATION and ERROR to components is defined in clause 3 of this specification.

The encoding rules which are applicable to the defined abstract syntax are the Basic Encoding Rules for Abstract Syntax Notation One, defined in CCITT Recommendation X.209 (1998) with the same exceptions as stated in GSM 09.02. For each Supplementary Service parameter which has to be transferred by a Supplementary Service 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, the first letter of the first word is lower-case and the first letter of the following words are capitalized (e.g. "bearer service" is mapped to "bearerService"). In addition some words may be abbreviated as follows:

- ms mobile subscriber;
- ss supplementary services;
- cug closed user group.

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 specific mention, it has to be considered as mandatory from a semantic point of view. When in an invoke component, a mandatory element is missing in any component or inner data structure, a reject component is returned with the problem code "Mistyped Parameter". 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; the associated type of error is "DataMissing".

In case an element is defined as mandatory in the protocol description (GSM 04.80 including imports from GSM 09.02), but is not present according to the service description (stage 1 to stage 3), the ASN.1 protocol description takes precedence over the diagrams in the GSM 04.8x and 04.9x-series of technical specifications.

When possible operations and errors are imported from GSM 09.02 thereby making the MSC transparent to most of the messages sent to or from the MS.

Timer values for operations which require timers are shown as ASN.1 comments.

Ellipsis Notation shall be used in the same way as described in GSM 09.02 and shall be supported on the radio interface by the MS and the network for all operations defined in this specification including those imported from GSM 09.02.

## 4.2 Operation types

Table 4.1 summarizes the operations defined for supplementary services in this specification and shows which of these operations are call related and call independent. The terms "call related" and "call independent" are defined in GSM 04.10.

**Table 4.1: Relevance of supplementary service operations**

Operation name	Call related SS	Call independent SS
RegisterSS	-	+
EraseSS	-	+
ActivateSS	-	+
DeactivateSS	-	+
InterrogateSS	-	+
RegisterPassword	-	+
GetPassword	-	+
ProcessUnstructuredSS-Data	+	+
ForwardCheckSS-Indication	-	+
ProcessUnstructuredSS-Request	-	+
UnstructuredSS-Request	-	+
UnstructuredSS-Notify	-	+
ForwardChargeAdvice	+	-
NotifySS	+	-
ForwardCUG-Info	+	-
BuildMPTY	+	-
HoldMPTY	+	-
RetrieveMPTY	+	-
SplitMPTY	+	-
ExplicitCT	+	-
NOTE: The ProcessUnstructuredSS-Data operation may be used call related by a GSM Phase 1 MS.		

```

SS-Operations {
    ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
    ss-Operations (0) version3 (3)}

DEFINITIONS ::=

BEGIN

EXPORTS

-- exports operation types

-- operations defined in this specification
ProcessUnstructuredSS-Data, NotifySS, ForwardChargeAdvice, ForwardCUG-Info, BuildMPTY, HoldMPTY,
RetrieveMPTY, SplitMPTY, ExplicitCT;

IMPORTS

OPERATION FROM
TCAPMessages {
    ccitt recommendation q 773 modules (2) messages (1) version2 (2)}

-- The MAP operations:
-- RegisterSS, EraseSS, ActivateSS, DeactivateSS, InterrogateSS, RegisterPassword,
-- GetPassword, ProcessUnstructuredSS-Request, UnstructuredSS-Request, UnstructuredSS-Notify
-- ForwardCheckSS-Indication
-- are imported from MAP-Operations in SS-Protocol module.

-- imports SS-data types
NotifySS-Arg,
ForwardChargeAdviceArg,
ForwardCUG-InfoArg,
SS-UserData
FROM SS-DataTypes {
    ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
    ss-Datatypes (2) version3 (3)}

-- imports MAP-errors
IllegalSS-Operation, SS-ErrorStatus, SS-NotAvailable,
SS-Incompatibility, SystemFailure, FacilityNotSupported, CallBarred, UnexpectedDataValue
FROM MAP-Errors {

```



```

ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
map-Errors (10) version3 (3)}

-- imports SS-Errors
ResourcesNotAvailable, MaxNumberOfMPTY-ParticipantsExceeded
FROM SS-Errors {
  ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
  ss-Errors (1) version3 (3)}
;

-- operation types definition

ProcessUnstructuredSS-Data ::= OPERATION -- Timer T(PUSSD)= 15s to 30s
  ARGUMENT
  ss-UserData      SS-UserData
  RESULT
  ss-UserData      SS-UserData
-- optional
  ERRORS{
  SystemFailure,
  UnexpectedDataValue}

NotifySS ::= OPERATION
  ARGUMENT
  notifySS-Arg     NotifySS-Arg

ForwardChargeAdvice ::= OPERATION -- Timer T(AoC)= 1s to 40s
  ARGUMENT
  forwardChargeAdviceArg ForwardChargeAdviceArg
  RESULT

ForwardCUG-Info ::= OPERATION
  ARGUMENT
  forwardCUG-InfoArg ForwardCUG-InfoArg

BuildMPTY ::= OPERATION -- Timer T(BuildMPTY)= 5s to 30s
  RESULT
  ERRORS{
  IllegalSS-Operation,
  SS-ErrorStatus,
  SS-NotAvailable,
  SS-Incompatibility,
  SystemFailure,
  ResourcesNotAvailable,
  MaxNumberOfMPTY-ParticipantsExceeded}

HoldMPTY ::= OPERATION -- Timer T(HoldMPTY)= 5s to 30s
  RESULT
  ERRORS{
  IllegalSS-Operation,
  SS-ErrorStatus,
  SS-Incompatibility,
  FacilityNotSupported,
  SystemFailure}

RetrieveMPTY ::= OPERATION -- Timer T(RetrieveMPTY)= 5s to 30s
  RESULT
  ERRORS{
  IllegalSS-Operation,
  SS-ErrorStatus,
  SS-Incompatibility,
  FacilityNotSupported,
  SystemFailure}

SplitMPTY ::= OPERATION -- Timer T(SplitMPTY)= 5s to 30s
  RESULT
  ERRORS{
  IllegalSS-Operation,
  SS-ErrorStatus,
  SS-Incompatibility,
  FacilityNotSupported,
  SystemFailure}

ExplicitCT ::= OPERATION -- Timer T(ECT)= 5s to 15s
  RESULT
  ERRORS{
  IllegalSS-Operation,
  SS-ErrorStatus,
  SS-NotAvailable,
  SS-Incompatibility,
  FacilityNotSupported,
  SystemFailure,
  ResourcesNotAvailable,
  CallBarred}

```

END

**4.2.1 [spare]****4.2.2 Operation types description**

For each operation type this subclause provides a brief prose description.

**4.2.2.1 RegisterSS (MS --> network)**

This operation type is invoked by an MS to register data related to a supplementary service in the network. When no BasicService parameter is provided, the registration applies to all provisioned and applicable basic services.

**4.2.2.2 EraseSS (MS --> network)**

This operation type is invoked by an MS to erase data related to a supplementary service in the network. When no BasicService parameter is provided, the erasure applies to all provisioned and applicable basic services.

**4.2.2.3 ActivateSS (MS --> network)**

This operation type is invoked by an MS to request the network for a supplementary service activation. When no BasicService parameter is provided, the activation applies to all provisioned and applicable basic services.

**4.2.2.4 DeactivateSS (MS --> network)**

This operation type is invoked by an MS to request the network for a supplementary service deactivation. When no BasicService parameter is provided, the deactivation applies to all provisioned and applicable basic services.

**4.2.2.5 InterrogateSS (MS --> network)**

This operation type is invoked by an MS to request the network for a supplementary service interrogation. When no BasicService parameter is provided, the interrogation applies to all provisioned and applicable basic services.

**4.2.2.6 NotifySS (network --> MS)**

This operation type is invoked by the network to forward a supplementary service notification towards a mobile subscriber.

**4.2.2.7 RegisterPassword (MS --> network)**

This operation type is invoked by an MS to register a new password related to the management by the subscriber himself of subscription data in the HLR. The operation "Register password" will be successful if the subscriber can provide the old password, the new password and the new password again as results of 3 subsequent operations "Get password".

**4.2.2.8 GetPassword (network --> MS)**

This operation type is invoked by the network to request a password from the mobile subscriber. It may be used to allow the registration of a new password or the management of subscription data by the subscriber himself (e.g. modification of call barring activation status).

**4.2.2.9 ProcessUnstructuredSS-Data (MS --> network)**

This operation type is invoked by an MS to relay unstructured information in order to allow end to end SS operation between the MS and the network following specific rules (e.g. embedding of keypad commands). The operation is used in order to provide backward compatibility (see GSM 04.90).

**4.2.2.10 ProcessUnstructuredSS-Request (MS --> network)**

This operation type is invoked by an MS to start an unstructured supplementary service data application in the network.

**4.2.2.11 UnstructuredSS-Request (network --> MS)**

This operation type is invoked by the network to request unstructured information from the MS in order to perform an unstructured supplementary service data application.

**4.2.2.12 UnstructuredSS-Notify (network --> MS)**

This operation type is invoked by the network to give an unstructured supplementary service notification to the mobile user.

**4.2.2.13 ForwardCheckSSIndication (network --> MS)**

This operation type is invoked by the network to indicate to the mobile subscriber that the status of supplementary services may not be correct in the network. The procedures for initiating ForwardCheckSSIndication are specified in GSM 09.02.

**4.2.2.14 ForwardChargeAdvice (network --> MS)**

This operation type is invoked by the network to forward Advice of Charge information to the mobile subscriber.

**4.2.2.15 BuildMPTY (MS --> network)**

This operation type is invoked by an MS to request the network to connect calls in a multi party call.

**4.2.2.16 HoldMPTY (MS --> network)**

This operation type is invoked by an MS to put the MS-connection to a multi party call (invoked by that MS) on hold.

**4.2.2.17 RetrieveMPTY (MS --> network)**

This operation type is invoked by an MS to request retrieval of a multi party call held by that MS.

**4.2.2.18 SplitMPTY (MS --> network)**

This operation type is invoked by an MS to request a private communication with one of the remote parties in a multi party call invoked by that MS.

**4.2.2.19 ForwardCUG-Info (MS --> network)**

This operation type is used by an MS to explicitly invoke a CUG call.

**4.2.2.20 ExplicitCT (MS --> Network)**

This operation type is invoked by an MS to request the network to connect the two calls of the subscriber.

### 4.3 Error types

#### 4.3.1 Error types ASN.1 specification

The following ASN.1 module provides an ASN.1 specification of errors. Errors from MAP are imported in the SS-Protocol module in subclause 4.5.

```

SS-Errors {
  ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
  ss-Errors (1) version3 (3)}

DEFINITIONS ::=

BEGIN

IMPORTS

ERROR FROM
TCAPMessages {
  ccitt recommendation q 773 modules (2) messages (1) version2 (2)};

-- The MAP errors
-- UnknownSubscriber, BearerServiceNotProvisioned, TeleserviceNotProvisioned,
-- IllegalSS-Operation, SS-ErrorStatus, SS-NotAvailable, SS-SubscriptionViolation,
-- SS-Incompatibility, SystemFailure, DataMissing, UnexpectedDataValue, FacilityNotSupported,
-- PW-RegistrationFailure, NegativePW-Check, CallBarred, NumberOfPW-AttemptsViolation,
-- AbsentSubscriber, IllegalSubscriber, IllegalEquipment, USSD-Busy, UnknownAlphabet
-- are imported from MAP-Errors in SS-Protocol module.

-- error types definition
ResourcesNotAvailable ::= ERROR
MaxNumberOfMPTY-ParticipantsExceeded ::= ERROR

END

```

#### 4.3.2 Error types description

For each error type this subclause provides a brief prose description.

##### 4.3.2.1 UnknownSubscriber

This error is returned by the network when it is requested to perform an operation concerning an unknown subscriber.

##### 4.3.2.2 BearerServiceNotProvisioned

This error is returned by the network when it is requested to perform an operation on a supplementary service and not even a subset of the requested bearer service group has been subscribed to.

##### 4.3.2.3 TeleServiceNotProvisioned

This error is returned by the network when it is requested to perform an operation on a supplementary service and not even a subset of the requested teleservice group has been subscribed to.

#### **4.3.2.4 IllegalSS-Operation**

This error is returned by the network when it is requested to perform an illegal operation which is defined as not applicable for the relevant supplementary service(s) (e.g. registration request for a service which must be registered by the administration). For the definition of the allowed operations for the individual supplementary services, see GSM 04.8x and 04.9x-series of technical specifications.

#### **4.3.2.5 SS-ErrorStatus**

This error is returned by the network when it is requested to perform an operation which is not compatible with the current status of the relevant supplementary service. The current status may be given as additional information by use of the SS-parameter.

#### **4.3.2.6 SS-NotAvailable**

This error is returned by the network when it is requested to perform an operation on a supplementary service which is not available in the current location area.

#### **4.3.2.7 SS-SubscriptionViolation**

This error is returned by the network when it is requested to perform an operation on a supplementary service, transgressing the subscription restrictions. The nature of the restriction or the transgressed options may be sent as parameters.

#### **4.3.2.8 SS-Incompatibility**

This error is returned by the network when it is requested for a supplementary service operation incompatible with the status of an other supplementary service or with the teleservice or bearer service for which the operation is requested. This error shall only be used if the operation is not compatible for even a subset of the teleservice group or bearer service group specified in the request. The identity and status of the conflicting service may also be indicated. The additional information may contain the SS-code parameter, the Basic Service Group parameter and the SS-status parameter.

#### **4.3.2.9 SystemFailure**

This error is returned by the network, when it cannot perform an operation because of a failure in the network.

#### **4.3.2.10 DataMissing**

This error is returned by the network when an optional parameter is missing in an invoke component or an inner data structure, while it is required by the context of the request.

#### **4.3.2.11 UnexpectedDataValue**

This error is returned by the network when it receives a parameter with an unexpected value, without type violation.

#### **4.3.2.12 PasswordRegistrationFailure**

This error is returned when a password registration procedure fails because of abnormal subscriber inputs. A more specific diagnostic may be passed as error parameter and indicates situations such as:

- invalid password format;
- new passwords mismatch.

**4.3.2.13 NegativePasswordCheck**

This error is returned to indicate the negative result of a password check because the subscriber has not provided the required password or has provided a password which does not match the valid one.

**4.3.2.14 FacilityNotSupported**

This error is returned by the network receiving a request about a facility which is not supported in the PLMN.

**4.3.2.15 ResourcesNotAvailable**

This error is returned by the network to the MS if temporarily there are no resources to support e.g. a multi party call available in the network.

**4.3.2.16 MaxNumberOfMPTY-ParticipantsExceeded**

This error is returned by the network to the MS if the request must be rejected because the number of subscribers to join a multi party call would exceed the maximum value.

**4.3.2.17 CallBarred**

This error is returned by the network to the MS when call independent subscriber control procedures are barred by the operator. The parameter "operator barring" shall be included.

**4.3.2.18 NumberOfPW-AttemptsViolation**

This error is returned by the network to the MS when the maximum number of wrong password attempts is exceeded.

**4.3.2.19 AbsentSubscriber**

This error is returned when the subscriber has activated the detach service or the system detects the absence condition. This error is not used on the radio interface but only between network entities.

**4.3.2.20 IllegalSubscriber**

This error is returned when illegality of the access has been established by use of authentication procedure. This error is not used on the radio interface but only between network entities.

**4.3.2.21 IllegalEquipment**

This error is returned when the IMEI check procedure has shown that the IMEI is blacklisted or not white-listed. This error is not used on the radio interface but only between network entities.

**4.3.2.22 USSD-Busy**

This error is returned by the MS to the network when the MS is not able to process the unstructured supplementary service data operation due to an on-going MMI input of the user or an already existing call independent supplementary service transaction.

**4.3.2.23 UnknownAlphabet**

This error is returned by the MS or the network when the alphabet/language used for the unstructured supplementary service data operation is not known by the network or the MS.

## 4.4 Data types and identifiers

### 4.4.1 General

The data types used in the SS protocol specifications are described in the ASN.1 module provided in subclause 4.4.2, while subclause 4.4.3 provides an overview of the identifiers used in SS ASN.1 specifications.

Since size constraints are subject to modifications named values have been defined in the following module for the upper boundaries of the value ranges associated to several sub-type specifications.

### 4.4.2 ASN.1 data types

This subclause provides an ASN.1 module defining the abstract data types in operations and errors specification. Only data types which are specific for this specification are defined. All other data types are imported from MAP together with the import of operations and errors.

```

SS-DataTypes {
    ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
    ss-DataTypes (2) version3 (3)}

DEFINITIONS

IMPLICIT TAGS ::=

BEGIN

-- exports all data types defined in this module

IMPORTS

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)}

-- imports MAP-SS-DataTypes
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)}

CUG-Index
FROM MAP-MS-DataTypes {
    ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
    map-MS-DataTypes (11) version3 (3)}

maxSignalInfoLength,
ISDN-AddressString,
ISDN-SubaddressString
FROM MAP-CommonDataTypes {
    ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
    map-CommonDataTypes (18) version3 (3)}
;

-- data types definition

SS-UserData ::= IA5String (SIZE (1.. maxSignalInfoLength))

NotifySS-Arg ::= SEQUENCE{
    ss-Code [1] SS-Code OPTIONAL,
    ss-Status [4] SS-Status OPTIONAL,
    ss-Notification [5] SS-Notification OPTIONAL,
    callIsWaiting-Indicator [14] NULL OPTIONAL,
    callOnHold-Indicator [15] CallOnHold-Indicator OPTIONAL,
    mpty-Indicator [16] NULL OPTIONAL,
    cug-Index [17] CUG-Index OPTIONAL,
    clirSuppressionRejected [18] NULL OPTIONAL,
    ...
    ect-Indicator [19] ECT-Indicator OPTIONAL}

ForwardChargeAdviceArg ::= SEQUENCE{
    ss-Code [0] SS-Code,
    chargingInformation [1] ChargingInformation,
    ...}

SS-Notification ::= OCTET STRING (SIZE (1))

```

```

--      Bit 8 7 6 5 4 00000 (Unused)

--      Bit 3  Call is forwarded indication to A-subscriber
--            (calling subscriber)
--      0  No information content
--      1  Outgoing call has been forwarded to C

--      Bit 2  Call is forwarded indication to B-subscriber
--            (forwarding subscriber)
--      0  No information content
--      1  Incoming call has been forwarded to C

--      Bit 1  Call is forwarded indication to C-subscriber
--            (forwarded-to subscriber)
--      0  No information content
--      1  Incoming call is a forwarded call

ChargingInformation ::= SEQUENCE{
    e1  [1] E1 OPTIONAL,
    e2  [2] E2 OPTIONAL,
    e3  [3] E3 OPTIONAL,
    e4  [4] E4 OPTIONAL,
    e5  [5] E5 OPTIONAL,
    e6  [6] E6 OPTIONAL,
    e7  [7] E7 OPTIONAL,
    ...}

E1 ::= INTEGER (0..max10TimesUnitsPerTime)
max10TimesUnitsPerTime INTEGER ::= 8191

E2 ::= INTEGER (0..max10TimesTimeInterval)
max10TimesTimeInterval INTEGER ::= 8191

E3 ::= INTEGER (0..max100TimesScalingFactor)
max100TimesScalingFactor INTEGER ::= 8191

E4 ::= INTEGER (0..max10TimesIncrement)
max10TimesIncrement INTEGER ::= 8191

E5 ::= INTEGER (0..max10TimesIncrementPerDataInterval)
max10TimesIncrementPerDataInterval INTEGER ::= 8191

E6 ::= INTEGER (0..maxNumberOfSegmentsPerDataInterval)
maxNumberOfSegmentsPerDataInterval INTEGER ::= 8191

E7 ::= INTEGER (0..max10TimesInitialTime)
max10TimesInitialTime INTEGER ::= 8191

CallOnHold-Indicator ::= ENUMERATED {
    callRetrieved (0),
    callOnHold (1)}

ForwardCUG-InfoArg ::= SEQUENCE {
    cug-Index          [0] CUG-Index OPTIONAL,
    suppressPrefCUG    [1] NULL OPTIONAL,
    suppressOA         [2] NULL OPTIONAL,
    ...}

ECT-Indicator ::= SEQUENCE {
    ect-CallState      [0] ECT-CallState,
    rdn [1] RDN OPTIONAL,
    ...}

ECT-CallState ::= ENUMERATED {
    alerting (0),
    active (1)}

RDN ::= CHOICE {
    presentationAllowedAddress          [0] RemotePartyNumber,
    presentationRestricted               [1] NULL,
    numberNotAvailableDueToInterworking [2] NULL,
    presentationRestrictedAddress        [3] RemotePartyNumber}

RemotePartyNumber ::= SEQUENCE {
    partyNumber          [0] ISDN-AddressString,
    partyNumberSubaddress [1] ISDN-SubaddressString OPTIONAL,
    ...}

END

```



#### **4.4.3 Identifiers definition**

The parameters which are described in the following subclauses correspond to the identifiers used in operation and error types description.

##### **4.4.3.1 chargingInformation**

The chargingInformation identifier refers to the necessary information for the Advice of Charge supplementary service (see GSM 02.24).

##### **4.4.3.2 e1**

The e1 identifier refers to 10 times the number of LPLMN units per time interval in connection with the Advice of Charge supplementary service, see GSM 02.24.

##### **4.4.3.3 e2**

The e2 identifier refers to 10 times the length of the time interval in seconds in connection with the Advice of Charge supplementary service, see GSM 02.24.

##### **4.4.3.4 e3**

The e3 identifier refers to 100 times the scaling factor to convert from LPLMN units to HPLMN units in connection with the Advice of Charge supplementary service, see GSM 02.24.

##### **4.4.3.5 e4**

The e4 identifier refers to 10 times the LPLMN increment in connection with the Advice of Charge supplementary service, see GSM 02.24.

##### **4.4.3.6 e5**

The e5 identifier refers to 10 times the number of LPLMN units incremented per data interval in connection with the Advice of Charge supplementary service, see GSM 02.24.

##### **4.4.3.7 e6**

The e6 identifier refers to the number of segments per data interval in connection with the Advice of Charge supplementary service, see GSM 02.24.

##### **4.4.3.8 e7**

The e7 identifier refers to 10 times the length of the initial time interval in seconds in connection with the Advice of Charge supplementary service, see GSM 02.24.

##### **4.4.3.9 ss-Code**

The ss-Code identifier refers to the code which identify a supplementary service or a group of supplementary services.

##### **4.4.3.10 ss-Notification**

The ss-Notification identifier refers to one or several supplementary service notifications which have to be forwarded to a mobile subscriber.

##### **4.4.3.11 ss-Status**

The ss-Status identifier refers to the status of a supplementary service.

**4.4.3.12        callsWaiting-Indicator**

The callsWaiting-Indicator identifier refers to the indication given to the mobile station that the call is waiting.

**4.4.3.13        callOnhold-Indicator**

The callOnHold-Indicator identifier refers to the indication given to the mobile station that the call has been put on hold or has been retrieved.

**4.4.3.14        mpty-Indicator**

The mpty-Indicator identifier refers to the indication given to the mobile station that the multi party call has been invoked.

**4.4.3.15        forwardCUG-InfoArg**

The forwardCUG-InfoArg identifier refers to the indication given from the mobile subscriber to the network in connection with explicit invocation of a CUG call.

**4.4.3.16        cug-Index**

The cug-Index identifier refers to the index of a CUG given in an explicit invocation of a CUG call.

**4.4.3.17        suppressPrefCUG**

The suppressPrefCUG identifier refers to the mobile subscribers request to the network to prohibit the use of the preferential CUG.

**4.4.3.18        suppressOA**

The suppressOA identifier refers to the mobile subscribers request to the network to prohibit the use of the subscriber option "OA allowed".

**4.4.3.19        clirSuppressionRejected**

The clirSuppressionRejected identifier refers to the indication given to the mobile station that the CLIR suppression request has been rejected.

**4.4.3.20        ect-Indicator**

The ect-Indicator identifier refers to the indication given to the mobile station that the call was transferred.

**4.4.3.21        ect-CallState**

The ect-CallState identifier refers to the state of the call to the other remote party in which Explicit Call Transfer was invoked.

**4.4.3.22        rdn**

The Rdn identifier refers to the line identity information of the other remote party.

**4.4.3.23        presentationAllowedAddress**

The presentationAllowedAddress identifier refers to the line identity of the other remote party that is allowed to be presented.

**4.4.3.24        presentationRestricted**

The presentationRestricted identifier refers to the restriction of presentation of the line identity of the other remote party .

**4.4.3.25      numberNotAvailableDueToInterworking**

The numberNotAvailableDueToInterworking identifier refers to the unavailability of the line identity of the other remote party.

**4.4.3.26      presentationRestrictedAddress**

The presentationRestrictedAddress identifier refers to the line identity of the other remote party which presentation restriction is overridden.

**4.4.3.27      partyNumber**

The partyNumber identifier refers to the remote party number.

**4.4.3.28      partyNumberSubaddress**

The partyNumberSubaddress identifier refers to remote party number subaddress.

## 4.5 Operations and errors implementation

For the actual implementation of supplementary services, operations and errors have to be defined by value. The following ASN.1 module, imports operation types from the ASN.1 module described in subclause 4.2 and operation and error types from MAP. It defines operations by allocating operations and errors a local value. For the involved operations and errors the same local values as in MAP are allocated.

```

SS-Protocol {
    ccitt identified-organization (4) etsi (0) mobileDomain (0)
    gsm-Access (2) modules (3) ss-Protocol (3) version3 (3)}

DEFINITIONS ::=

BEGIN

IMPORTS

-- imports operation types

-- imports operation type from MAP-MobileServiceOperations
ForwardCheckSS-Indication
FROM MAP-MobileServiceOperations {
    ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
    map-MobileServiceOperations (5) version3 (3)}

-- imports operation types from MAP-SupplementaryServiceOperations
RegisterSS, EraseSS, ActivateSS, DeactivateSS, InterrogateSS, RegisterPassword, GetPassword,
ProcessUnstructuredSS-Request, UnstructuredSS-Request, UnstructuredSS-Notify
FROM MAP-SupplementaryServiceOperations {
    ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
    map-SupplementaryServiceOperations (8) version3 (3)}

-- imports operation types from SS-Operations
ProcessUnstructuredSS-Data, NotifySS, ForwardChargeAdvice, BuildMPTY, HoldMPTY, RetrieveMPTY,
SplitMPTY, ExplicitCT, ForwardCUG-Info
FROM SS-Operations {
    ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
    ss-Operations (0) version3 (3)}

-- imports error types

-- imports error types from MAP-Errors
UnknownSubscriber, BearerServiceNotProvisioned, TeleserviceNotProvisioned,
IllegalSS-Operation, SS-ErrorStatus, SS-NotAvailable, SS-SubscriptionViolation,
SS-Incompatibility, SystemFailure, DataMissing, UnexpectedDataValue, PW-RegistrationFailure,
NegativePW-Check, FacilityNotSupported, CallBarred, NumberOfPW-AttemptsViolation,
AbsentSubscriber, IllegalSubscriber, IllegalEquipment, USSD-Busy, UnknownAlphabet

FROM MAP-Errors {
    ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Network (1) modules (3)
    map-Errors (10) version3 (3)}

-- imports error types from SS-Errors
ResourcesNotAvailable, MaxNumberOfMPTY-ParticipantsExceeded
FROM SS-Errors {
    ccitt identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
    ss-Errors (1) version3 (3)}
;

-- allocation of local values to operations

registerSS RegisterSS ::= localValue 10
eraseSS EraseSS ::= localValue 11
activateSS ActivateSS ::= localValue 12
deactivateSS DeactivateSS ::= localValue 13
interrogateSS InterrogateSS ::= localValue 14
notifySS NotifySS ::= localValue 16
registerPassword RegisterPassword ::= localValue 17
getPassword GetPassword ::= localValue 18
processUnstructuredSS-Data ProcessUnstructuredSS-Data ::= localValue 19
forwardCheckSS-Indication ForwardCheckSS-Indication ::= localValue 38
processUnstructuredSS-Request ProcessUnstructuredSS-Request ::= localValue 59
unstructuredSS-Request UnstructuredSS-Request ::= localValue 60
unstructuredSS-Notify UnstructuredSS-Notify ::= localValue 61
forwardCUG-Info ForwardCUG-Info ::= localValue 120
splitMPTY SplitMPTY ::= localValue 121
retrieveMPTY RetrieveMPTY ::= localValue 122
holdMPTY HoldMPTY ::= localValue 123
buildMPTY BuildMPTY ::= localValue 124
forwardChargeAdvice ForwardChargeAdvice ::= localValue 125
explicitCT ExplicitCT ::= localValue 126

```

-- allocation of local values to errors

```
unknownSubscriber    UnknownSubscriber ::= localValue 1
illegalSubscriber    IllegalSubscriber ::= localValue 9
bearerServiceNotProvisioned BearerServiceNotProvisioned ::= localValue 10
teleserviceNotProvisioned TeleserviceNotProvisioned ::= localValue 11
illegalEquipment     IllegalEquipment ::= localValue 12
callBarred           CallBarred ::= localValue 13
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
facilityNotSupported FacilityNotSupported ::= localValue 21
absentSubscriber     AbsentSubscriber ::= localValue 27
systemFailure        SystemFailure ::= localValue 34
dataMissing          DataMissing ::= localValue 35
unexpectedDataValue  UnexpectedDataValue ::= localValue 36
pw-RegistrationFailure PW-RegistrationFailure ::= localValue 37
negativePW-Check     NegativePW-Check ::= localValue 38
numberOfPW-AttemptsViolation NumberOfPW-AttemptsViolation ::= localValue 43
unknownAlphabet      UnknownAlphabet ::= localValue 71
ussd-Busy            USSD-Busy ::= localValue 72
maxNumberOfEMPTY-ParticipantsExceeded
  MaxNumberOfEMPTY-ParticipantsExceeded ::= localValue 126
resourcesNotAvailable ResourcesNotAvailable ::= localValue 127
```

END

## Annex A (informative): Expanded ASN.1 Module "SS-Protocol"

```

--          Expanded ASN1 Module 'SS-Protocol'
--SIEMENS ASN.1 Compiler      R4.11 (41-00-05)
--          Date: 99-02-10 Time: 07:34:09

SS-Protocol { 0 identified-organization (4) etsi (0) mobileDomain (0) gsm-Access (2) modules (3)
ss-Protocol (3) version3 (3) }

DEFINITIONS

 ::=

BEGIN

registerSS OPERATION
  ARGUMENT
    registerSS-Arg SEQUENCE {
      ss-Code      OCTET STRING ( SIZE ( 1 ) ),
      basicService CHOICE {
        bearerService [2] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),
        teleservice [3] IMPLICIT OCTET STRING ( SIZE ( 1 ) ) } OPTIONAL,
      forwardedToNumber [4] IMPLICIT OCTET STRING ( SIZE ( 1..20 ) ) OPTIONAL,
      forwardedToSubaddress [6] IMPLICIT OCTET STRING ( SIZE ( 1..21 ) ) OPTIONAL,
      noReplyConditionTime [5] IMPLICIT INTEGER ( 5..30 ) 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,
            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,
                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}
 ::= localValue : 11

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,
                    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-SubscriptionViolation -- localValue : 19,
-- ss-Incompatibility -- localValue : 20,
-- negativePW-Check -- localValue : 38,
-- numberOfPW-AttemptsViolation -- localValue : 43}
::= localValue : 12

deactivateSS 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,
            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-SubscriptionViolation -- localValue : 19,
    -- negativePW-Check -- localValue : 38,
    -- numberOfPW-AttemptsViolation -- localValue : 43}
 ::= localValue : 13

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,
            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,
            ... },
        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}
 ::= localValue : 14

notifySS OPERATION
ARGUMENT
    notifySS-Arg SEQUENCE {
        ss-Code [1] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
        ss-Status [4] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
        ss-Notification [5] IMPLICIT OCTET STRING ( SIZE (1 ) ) OPTIONAL,
        callIsWaiting-Indicator [14] IMPLICIT NULL OPTIONAL,
        callOnHold-Indicator [15] IMPLICIT ENUMERATED {
            callRetrieved (0 ),
            callOnHold (1 )} OPTIONAL,
        mpty-Indicator [16] IMPLICIT NULL OPTIONAL,
        cug-Index [17] IMPLICIT INTEGER ( 0..32767 ) OPTIONAL,
        cliSuppressionRejected [18] IMPLICIT NULL OPTIONAL,
        ... ,
        ect-Indicator [19] IMPLICIT SEQUENCE {
            ect-CallState [0] IMPLICIT ENUMERATED {
                alerting (0 ),
                active (1 )},
            rdn [1] CHOICE {
                presentationAllowedAddress [0] IMPLICIT SEQUENCE {

```

```

        partyNumber [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ),
        partyNumberSubaddress [1] IMPLICIT OCTET STRING ( SIZE (1..21 ) ) OPTIONAL,
        ... },
    presentationRestricted [1] IMPLICIT NULL,
    numberNotAvailableDueToInterworking [2] IMPLICIT NULL,
    presentationRestrictedAddress [3] IMPLICIT SEQUENCE {
        partyNumber [0] IMPLICIT OCTET STRING ( SIZE (1..20 ) ) ( SIZE (1..9 ) ),
        partyNumberSubaddress [1] IMPLICIT OCTET STRING ( SIZE (1..21 ) ) OPTIONAL,
        ... } OPTIONAL,
    ... } OPTIONAL}
 ::= localValue : 16

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

processUnstructuredSS-Data OPERATION
 ARGUMENT
  ss-UserData IA5String ( SIZE (1..200 ) )
 RESULT
  ss-UserData IA5String ( SIZE (1..200 ) )
 ERRORS {
  -- systemFailure -- localValue : 34,
  -- unexpectedDataValue -- localValue : 36}
 ::= localValue : 19

forwardCheckSS-Indication OPERATION
 ::= localValue : 38

processUnstructuredSS-Request OPERATION
 ARGUMENT
  ussd-Arg SEQUENCE {
    ussd-DataCodingScheme OCTET STRING ( SIZE (1 ) ),
    ussd-String OCTET STRING ( SIZE (1..160 ) ),
    ... }
 RESULT
  ussd-Res SEQUENCE {
    ussd-DataCodingScheme OCTET STRING ( SIZE (1 ) ),
    ussd-String OCTET STRING ( SIZE (1..160 ) ),
    ... }
 ERRORS {
  -- systemFailure -- localValue : 34,
  -- dataMissing -- localValue : 35,
  -- unexpectedDataValue -- localValue : 36,
  -- unknownAlphabet -- localValue : 71,
  -- callBarred -- localValue : 13}
 ::= localValue : 59

unstructuredSS-Request OPERATION
 ARGUMENT
  ussd-Arg SEQUENCE {
    ussd-DataCodingScheme OCTET STRING ( SIZE (1 ) ),
    ussd-String OCTET STRING ( SIZE (1..160 ) ),
    ... }
 RESULT
  ussd-Res 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 : 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

forwardCUG-Info OPERATION
  ARGUMENT
    forwardCUG-InfoArg SEQUENCE {
      cug-Index [0] IMPLICIT INTEGER ( 0..32767 ) OPTIONAL,
      suppressPrefCUG [1] IMPLICIT NULL OPTIONAL,
      suppressOA [2] IMPLICIT NULL OPTIONAL,
      ... }
 ::= localValue : 120

splitEMPTY OPERATION
  RESULT
    zzzz-empty NULL
  ERRORS {
    -- illegalSS-Operation -- localValue : 16,
    -- ss-ErrorStatus -- localValue : 17,
    -- ss-Incompatibility -- localValue : 20,
    -- facilityNotSupported -- localValue : 21,
    -- systemFailure -- localValue : 34}
 ::= localValue : 121

retrieveEMPTY OPERATION
  RESULT
    zzzz-empty NULL
  ERRORS {
    -- illegalSS-Operation -- localValue : 16,
    -- ss-ErrorStatus -- localValue : 17,
    -- ss-Incompatibility -- localValue : 20,
    -- facilityNotSupported -- localValue : 21,
    -- systemFailure -- localValue : 34}
 ::= localValue : 122

holdEMPTY OPERATION
  RESULT
    zzzz-empty NULL
  ERRORS {
    -- illegalSS-Operation -- localValue : 16,
    -- ss-ErrorStatus -- localValue : 17,
    -- ss-Incompatibility -- localValue : 20,
    -- facilityNotSupported -- localValue : 21,
    -- systemFailure -- localValue : 34}
 ::= localValue : 123

buildEMPTY OPERATION
  RESULT
    zzzz-empty NULL
  ERRORS {
    -- illegalSS-Operation -- localValue : 16,
    -- ss-ErrorStatus -- localValue : 17,
    -- ss-NotAvailable -- localValue : 18,
    -- ss-Incompatibility -- localValue : 20,
    -- systemFailure -- localValue : 34,
    -- resourcesNotAvailable -- localValue : 127,
    -- maxNumberOfEMPTY-ParticipantsExceeded -- localValue : 126}
 ::= localValue : 124

forwardChargeAdvice OPERATION
  ARGUMENT
    forwardChargeAdviceArg SEQUENCE {
      ss-Code [0] IMPLICIT OCTET STRING ( SIZE ( 1 ) ),

```

```

chargingInformation [1] IMPLICIT SEQUENCE {
  e1      [1] IMPLICIT INTEGER ( 0..8191 ) OPTIONAL,
  e2      [2] IMPLICIT INTEGER ( 0..8191 ) OPTIONAL,
  e3      [3] IMPLICIT INTEGER ( 0..8191 ) OPTIONAL,
  e4      [4] IMPLICIT INTEGER ( 0..8191 ) OPTIONAL,
  e5      [5] IMPLICIT INTEGER ( 0..8191 ) OPTIONAL,
  e6      [6] IMPLICIT INTEGER ( 0..8191 ) OPTIONAL,
  e7      [7] IMPLICIT INTEGER ( 0..8191 ) OPTIONAL,
  ... },
... }
RESULT
  zzzz-empty NULL
 ::= localValue : 125

explicitCT OPERATION
  ERRORS {
    -- illegalSS-Operation -- localValue : 16,
    -- ss-ErrorStatus -- localValue : 17,
    -- ss-NotAvailable -- localValue : 18,
    -- ss-Incompatibility -- localValue : 20,
    -- facilityNotSupported -- localValue : 21,
    -- systemFailure -- localValue : 34,
    -- resourcesNotAvailable -- localValue : 127,
    -- callBarred -- localValue : 13}
 ::= localValue : 126

unknownSubscriber ERROR
  PARAMETER
    unknownSubscriberParam 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 : 1

illegalSubscriber ERROR
  PARAMETER
    illegalSubscriberParam 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 : 9

bearerServiceNotProvisioned ERROR
  PARAMETER
    bearerServNotProvParam 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 : 10

teleserviceNotProvisioned ERROR
  PARAMETER
    teleservNotProvParam 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 : 11

illegalEquipment ERROR
PARAMETER
    illegalEquipmentParam 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 : 12

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 ( {
                            '...' { @extId } ) OPTIONAL} OPTIONAL,
                    pcs-Extensions [1] IMPLICIT SEQUENCE {
                        ... } OPTIONAL,
                        ... } OPTIONAL,
                        ... }
            }
        }
    ::= localValue : 13

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

facilityNotSupported ERROR
PARAMETER
    facilityNotSupParam 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 : 21

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

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} 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 ( {
          '...' { @extId } ) OPTIONAL} OPTIONAL,
        pcs-Extensions [1] IMPLICIT SEQUENCE {
          ... } 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 ( {
              '...' { @extId } ) OPTIONAL} OPTIONAL,
            pcs-Extensions [1] IMPLICIT SEQUENCE {
              ... } OPTIONAL,
              ... } OPTIONAL,
              ... }
          }
    }
 ::= localValue : 36

pw-RegistrationFailure ERROR
  PARAMETER
    pw-RegistrationFailureCause ENUMERATED {
      undetermined (0 ),
      invalidFormat (1 ),
      newPasswordsMismatch (2 )}
 ::= localValue : 37

negativePW-Check ERROR
 ::= localValue : 38

numberOfPW-AttemptsViolation ERROR
 ::= localValue : 43

unknownAlphabet ERROR
 ::= localValue : 71

ussd-Busy ERROR
 ::= localValue : 72

maxNumberOfMPY-ParticipantsExceeded ERROR
 ::= localValue : 126

resourcesNotAvailable ERROR
 ::= localValue : 127

END

```

**Annex B (informative): Change Request History**

<b>Status of Technical Specification GSM 04.80 (Release 96)</b>		
<b>Date</b>	<b>Version</b>	<b>Remarks</b>
	3.1.0	Last common Phase 1/Phase 2 version
Release 92	3.2.0	(I-ETS 300 027 = version 3.0.1)
October 1990	4.0.0	CR 04.80-04 rev 1 (category B) approved by GSM#28
January 1991	4.1.0	CR 04.80-10 rev 3 (category A) CR 04.80-16 (category D); all approved by GSM#29
March 1991	4.2.0	CR 04.80-05 rev 3 (category B) approved by GSM#30
June 1991	4.3.0	CR 04.80-10 rev 3 (category D) / CR 04.80-12 rev 3 (category D) CR 04.80-18 rev 1 (category B) / CR 04.80-19 rev 2 (category B) CR 04.80-20 (category B); all approved by GSM#31
April 1992	4.4.0	CR 04.80-21 rev 3 (category D) / CR 04.80-22 rev 2 (category C) CR 04.80-23 rev 2 (category C) / CR 04.80-25 rev 2 (category D) CR 04.80-26 rev 2 (category D) / CR 04.80-27 (category D) CR 04.80-28 rev 1 (category D) / CR 04.80-29 (category D) CR 04.80-30 rev 1 (category C); all approved by SMG#02
September 1992	4.4.1	CR 04.80-33 (category D) approved by SMG#04
January 1993	4.5.0	CR 04.80-35 (category C) approved by SMG#04
April 1993	4.6.0	CR 04.80-36 rev 2 (category D) / CR 04.80-37 rev 1 (category D) CR 04.80-39 rev 1 (category D) CR 04.80-40 rev 1 (category C); all approved by SMG#06
June 1993	4.7.0	CR 04.80-38 (category C) / CR 04.80-42 rev 1 (category D) CR 04.80-43 rev 1 (category D) / CR 04.80-44 (category D) CR 04.80-45 rev 2 (category D) / CR 04.80-47 rev 1 (category C) CR 04.80-49 (category C); all approved by SMG#07 TS conditionally frozen by SMG#07 (except USSD)
October 1993	4.7.1	CR 04.80-50 (category D) approved by SMG#08 TS changed to draft prETS 300 564
January 1994	4.8.0	CR 04.80-46 rev 6 (category F) / CR 04.80-52 (category F) CR 04.80-53 (category F) / CR 04.80-54 (category F) CR 04.80-55 rev 1 (category F) CR 04.80-56 (category F); all approved by SMG#09 TS frozen by SMG#09
April 1994	4.9.0	CR 04.80-55 rev 2 (category F) approved by SMG#10
October 1994	4.9.1	CR 04.80-58 (category D) CR 04.80-59 rev 1 (category D) all approved by SMG#12 TS changed to final draft prETS 300 564
January 1995	4.9.2	TS changed to ETS 300 564 First edition
October 1995	4.10.0	AR 04.80-A003 (category D) approved by SMG#16
February 1996	4.11.0	AR 04.80-A004 (category F) approved by SMG#17



<b>Status of Technical Specification GSM 04.80 (continuation)</b>		
<b>Date</b>	<b>Version</b>	<b>Remarks</b>
April 1996	5.0.0	CR 04.80-A005 (category B) (R96-ECT) approved by SMG#18 (Release 96)
December 1996	5.0.1	TS changed to draft prETS 300 950 (Release 96)
May 1997	5.0.2	TS changed to ETS 300 950 first edition
December 1997	5.1.0	CR 04.80-A007r2 (category B) (Release 97-CNAP) approved by SMG#24 data types for CNAP in section 4.4.2 <i>ASN.1 data types</i> and Identifiers definition for CNAP in section 4.4.3. Also section 4.4.3.24 <i>presentationRestricted</i> has been modified for the support of CNAP \$(CNAP)\$
March 1998	5.2.0	CR 04.80-A006r1 (cat F) (Release 96 - TEI) approved by SMG#25
October 1998	5.2.1	ETS 300 950 fourth edition
February 1999	5.3.0	CR 04.80-A014 (R96-F) (CNAP) approved by SMG#28 removal of CNAP, because CNAP is not part of Release 96
<b>Text and figures:</b> WinWord 6.0 <b>ASN.1:</b> Siemens ASN.1 compiler <b>Stylesheet:</b> etsiw_60.dot <b>Rapporteur:</b>		

**History**

<b>Document history</b>			
December 1996	Unified Approval Procedure	UAP 61:	1996-12-16 to 1997-04-11
May 1997	First Edition		
February 1998	One-step Approval Procedure (Second Edition)	OAP 9826:	1998-02-27 to 1998-06-26
May 1998	One-step Approval Procedure (Third Edition)	OAP 9841:	1998-05-20 to 1998-10-16
July 1998	Second Edition		
October 1998	Third Edition		
March 1999	One-step Approval Procedure (Fourth Edition)	OAP 9929:	1999-03-19 to 1999-07-16