GSM Protocol Stack

Test Specification
SMS

Author: Condat AG
Alt-Moabit 90a
D-10559 Berlin
Germany

Date:	10 February 2003
Document No.:	8410.400.02.108
File:	SMS.DOC

Table of Contents

0 DOCUMENT CONTROL 5

0.1 Document History 5

0.2 References 5

0.3 Abbreviations 10

0.4 Terms 12

1 OVERVIEW 13

2 PARAMETERS 15

3 TEST CASES 82

3.1 Routing (internal) and Configuration 82
3.1.1 SMS000: Setup the routing and PCO view for the SMS test 82
3.1.2 SMS001: ME Memory not available 83
3.1.3 SMS002: ME Memory available (3 Records, empty) 83
3.1.4 SMS003: ME Memory available (3 Records, 2 occupied) 83
3.1.5 SMS004: ME Memory available (3 Records, all occupied) 84

3.2 Initialisation Phase 84
3.2.1 SMS005: Start with CPHS and IMSI Check 84
3.2.2 SMS006: Start with CPHS and IMSI Check (with ME Memory) 86
3.2.3 SMS007: No Short Messages available (ME Memory) 87
3.2.4 SMS008: MO and MT-SM available (ME Memory) 87
3.2.5 SMS009: MO and MT-SM available (ME Memory full) 88
3.2.6 SMS010: Read and check additonal SIM Properties 89
3.2.7 SMS012: No Short Messages available (SIM Memory) 89
3.2.8 SMS013: MO-SMS message available 91
3.2.9 SMS014: MT-SMS message available 93
3.2.10 SMS015: MO and MT-SMS message available 94
3.2.11 SMS016: 1 MO and 2 MT-SMS messages available, SIM memory full 96
3.2.12 SMS017: 1 MO and 2 MT-SMS messages available, SIM memory full not set 97
3.2.13 SMS018: MO and MT-SMS message available, Memory full is set 99
3.2.14 SMS019: SIM memory full, ME memory existent, MCEF set 101
3.2.15 SMS021: Unexpected Messages stored on SIM 103
3.2.16 SMS022: No SMS messages available, but SMS download possible 104
3.2.17 SMS031: Configuring the SMS Entity 106
3.2.18 SMS032: Configuring the SMS Entity (supporting STATUS REQUEST) 107
3.2.19 SMS033: Configuring the SMS Entity (support MT Acknowledge Response) 108
3.2.20 SMS034: Configuring the SMS Entity (supporting ME Memory) 108

3.3 Mobile Originated Short Message 110
3.3.1 SMS041: Initiation by MMI 110
3.3.2 SMS042: No SMS Connection established 112
3.3.3 SMS043: Confirmation of SMS Connection 113
3.3.4 SMS044: Timeout TR1M 114
3.3.5 SMS045: Multiple Request by MMI 115
3.3.6 SMS046: Confirmation to CP-DATA Message 115
3.3.7 SMS047: Wrong Message received (CP_DATA) 116
3.3.8 SMS048: Wrong Message received (unknown) 117
3.3.9 SMS049: CP Error received 118
3.3.10 SMS050: Lower Layer Failure 119
3.3.11 SMS051: Release of MM-Connection 120
3.3.12 SMS052: Timeout TC1M (first time) 120
3.3.13 SMS053: Timeout TC1M (max retransmissions) 121
3.3.14 SMS061: Acknowledgment of the Network (SUBMIT successful) 121
3.3.15 SMS062: Error signalled by the Network 123
3.3.16 SMS063: Wrong Message signalled by the Network (CP layer, CP-DATA) 124
3.3.17 SMS064: Wrong Message signalled by the Infrastructure (CP layer, unknown)125

3.3.18 SMS065: Wrong Message signalled by the Infrastructure (RL layer) 126
3.3.19 SMS066: Timeout TR1M while waiting for RP-ACK 127
3.3.20 SMS067: Lower Layer Failure 128
3.3.21 SMS068: Release of MM-Connection 129

3.4 Interworking MO-SM with MT-SM 130
3.4.1 SMS071: Sending of MO-SM parallel to Reception of Class 1 Message (stored in SIM-Card) 130
3.4.2 SMS081: Reception of Class 1 Message (stored in SIM-Card) parallel to Writing of MO-SM to SIM 133
3.4.3 SMS082: Reception of a Replaceable Message parallel to Writing of MO-SM to SIM (Case 1) 135
3.4.4 SMS083: Reception of a Replaceable Message parallel to Writing of MO-SM to SIM (Case 2) 138

3.5 Mobile Originated Short Message Command 142
3.5.1 SMS091: Mobile Originated Short Message Command 142

3.6 Mobile Terminated Short Message Service 145
3.6.1 SMS110: Reception of a Short Message (Preamble) 145
3.6.2 SMS111: Reception of a CP-ERROR Message 146
3.6.3 SMS112: Reception of a CP-ACK message 146
3.6.4 SMS113: Reception of an unknown message 147
3.6.5 SMS121: Second Mobile Terminated Connection 148
3.6.6 SMS122: Additional Message for Mobile Terminated Connection 149
3.6.7 SMS131: MT Message, Phase 2+ Handling, Normal Case 150
3.6.8 SMS132: Class 0 Message, Phase 2+ Handling, no Response 152
3.6.9 SMS133: Class 0 Message, Phase 2+ Handling, Failure case 154
3.6.10 SMS151: Class 0 Message 154
3.6.11 SMS152: Class 0 Message (long message) 156
3.6.12 SMS153: Class 0 Message (long message), stored on SIM 158
3.6.13 SMS161: Class 1 - Reception of Message (stored in SIM-Card) 160
3.6.14 SMS162: Class 1 - Reception of Message (stored in ME Memory) 162
3.6.15 SMS163: Class 1 - Reception of Message (routed to TE) 164
3.6.16 SMS164: Class 1 - Storage available on SIM, writing is NOT successfull 166
3.6.17 SMS171: Class 2 - Storage available on SIM, writing is successfull 168
3.6.18 SMS172: Class 2 - Storage available on SIM, writing not successfull, MCEF set 171
3.6.19 SMS173: Class 2 - Storage not available on SIM/ME, MCEF to be set 173
3.6.20 SMS174: Class 2 - Storage not available on SIM/ME, MCEF already set. 175
3.6.21 SMS175: Class 2 - Storage not available on SIM, but on ME, MCEF not to be set 176
3.6.22 SMS181: Class 3 Message 178
3.6.23 SMS182: Class 3 - Storage available on SIM, writing is not successfull 181
3.6.24 SMS191: No Class - Storage available on SIM, writing is successfull 183
3.6.25 SMS192: No Class - Storage available on SIM, writing is not successfull 185

3.7 Replace MT Short Message 187
3.7.1 SMS200: Storage of a Replaceable Short Message 187
3.7.2 SMS201: PID, SC and OA okay, replace message 189
3.7.3 SMS202: PID different, SC and OA okay, don´t replace message 191
3.7.4 SMS203: PID and OA okay, SC is different, replace message 193
3.7.5 SMS204: PID and SC okay, OA is different, don´t replace message 195

3.8 Mobile Terminated Short Message Status 197
3.8.1 SMS261: Mobile Terminated Short Message Status 197
3.8.2 SMS262: Mobile Terminated Short Message Status, Phase 2+ handling 199

3.9 One2One Operator-specific Short Message Handling 200
3.9.1 SMS311: Receiving SIM-specific Message 200
3.9.2 SMS312: Receiving SIM-specific Message, Memory Full Condition 202
3.9.3 SMS313: Receiving SIM-specific Message, Write Failure 204
3.9.4 SMS314: Receiving Empty Message 206

3.10 Short Message Service Procedures 207
3.10.1 SMS401: Storing of Short Messages (any record) 207

3.10.2 SMS402: Storing of Short Messages (given record) ... 209
3.10.3 SMS403: Storing of Short Messages, failure cases ... 211
3.10.4 SMS411: Changing of Short Messages ... 213
3.10.5 SMS421: Reading of Short Messages, Mobile originated, not read ... 215
3.10.6 SMS422: Reading of Short Messages, Mobile originated, already read ... 217
3.10.7 SMS423: Reading of Short Messages, Mobile terminated, not read ... 218
3.10.8 SMS424: Reading of Short Messages, Mobile terminated, already read ... 219
3.10.9 SMS425: Reading of Short Messages, failure cases ... 220
3.10.10 SMS426: Searching for Short Messages, any Message ... 222
3.10.11 SMS427: Searching for Short Messages, Mobile Originated, not Sent ... 224
3.10.12 SMS429: Searching for Short Messages, Mobile Terminated, not Read ... 226
3.10.13 SMS431: Preview Reading of Short Messages, Mobile originated, not read ... 227
3.10.14 SMS432: Preview-Reading of Short Messages, Mobile originated, already read228
3.10.15 SMS435: Preview-Reading of Short Messages, failure cases ... 229
3.10.16 SMS441: Status Change Reading of Short Messages, Mobile Originated, not Sent ... 230
3.10.17 SMS442: Status Change-Reading of SMS, Mobile Originated, already Sent ... 231
3.10.18 SMS443: Status Change Reading of Short Messages, Mobile terminated, not read ... 231
3.10.19 SMS444: Reread after Status Change Reading of Short Message, Mobile terminated, previously not read ... 233
3.10.20 SMS445: Status Change Reading of SMS, Mobile Terminated, already Read ... 233
3.10.21 SMS446: Reread after Status Change Reading of Short Message, Mobile terminated, previously read ... 234
3.10.22 SMS447: Status Change-Reading of Short Messages, failure cases ... 235
3.10.23 SMS471: Deletion of a Short Messages from SIM, successful case ... 237
3.10.24 SMS472: Deletion of a Short Messages from ME, successful case ... 238
3.10.25 SMS473: Deleting of Short Messages, failure cases ... 239
3.10.26 SMS474: Deleting of Mobile Originated Short Message from full SIM memory 240
3.10.27 SMS475: Deleting of Mobile Originated Short Message from full ME memory . 242
3.10.28 SMS476: Deleting of Short Message from SIM, no RP-SMMA ... 244
3.10.29 SMS477: Deleting of Short Message from ME Memory, no RP-SMMA ... 245
3.10.30 SMS480: Memory Available Notification ... 246
3.10.31 SMS481: SMS Ready Indication after Memory Available Notification ... 247
3.10.32 SMS482: Repeat RP-SMMA after Timeout TR1M (Timer TRAM) ... 248
3.10.33 SMS491: Modifying and Sending a Stored Message (MO) ... 250
3.10.34 SMS492: Update SIM with Sent Message (MO) ... 252
3.10.35 SMS493: Modifying and Sending a Stored Message (MT) ... 254
3.10.36 SMS494: Update SIM with Forwarded Message (MT) ... 256
3.10.37 SMS495: Unsuccessful Transmission of a Forwarded Message ... 258
3.10.38 SMS496: Unsuccessful Storing of a Forwarded Message ... 259
3.10.39 SMS497: Start Sent Stored Message while an Income MT-SM is stored ... 261
3.10.40 SMS498: Income MT-SM while Message to be sent is read from SIM ... 265
3.10.41 SMS499: Income MT-SM while Sent Message is stored on SIM ... 269

3.11 SIM Toolkit - SMS Download ... 274
3.11.1 SMS511: Class 2 - PID SMS Download, no support by SIM ... 274
3.11.2 SMS512: Class 2 - PID SMS Download, wrong DCS ... 276
3.11.3 SMS521: Class 2 - PID SMS Download, support by SIM ... 278
3.11.4 SMS522: SMS Download, response 90 00 ... 279
3.11.5 SMS523: SMS Download, response 93 00 ... 280
3.11.6 SMS524: SMS Download, response 9F 05 ... 281
3.11.7 SMS525: SMS Download, response 9E 05 ... 282

3.12 Engineering Mode ... 283
3.12.1 SMS551: Acknowledge of the Infrastructure ... 283

0 Document Control

Every effort has been made to ensure that the information contained in this document is accurate at the time of printing. However, the software described in this document is subject to continuous development and improvement. Condat GmbH reserves the right to change the specification of the software. Information in this document is subject to change without notice and does not represent a commitment on the part of Condat GmbH. Condat GmbH accepts no liability for any loss or damage arising from the use of any information contained in this document.

TI Berlin AG

Alt Moabit 90a

10559 Berlin

Germany

Telephone: +49.30.3983-0

Fax: +49.30.3983-1300

Internet: http://www.ti.com

0.1 Document History

Document Id.	Date	Author	Remarks
6147.400.97.100	25-Apr-97	Stefan Lemke et al	Initial
8410.400.98.105	21-Nov-2001	Frank Kaiser	Major rework due to change of SAP MNSMS
8410.400.02.106	26-Apr-2002	Frank Kaiser	Test One2One Special Message Indication
8410.400.02.107	21-Jun-2002	Frank Kaiser	Support of ME memory
8410.400.02.108	10-Feb-2003	Frank Kaiser	bit length for SMS SDUs received from memory adjusted

0.2 References

[1] GSM 2.81, Line Identification Supplementary Services - Stage 1
ETS 300 514, ETSI, September 1994

[2] GSM 2.82, Call Forwarding Supplementary Services - Stage 1
ETS 300 515, ETSI, September 1994

[3] GSM 2.83, Call Waiting and Call Hold Supplementary Services - Stage 1
ETS 300 516, ETSI, September 1994

[4] GSM 2.84, Multi Party Supplementary Services - Stage 1
ETS 300 517, ETSI, September 1994

[5] GSM 2.85, Closed User Group Supplementary Services - Stage 1
ETS 300 518, ETSI, September 1994

[6] GSM 2.86, Advice of Charge Supplementary Services - Stage 1
ETS 300 519, ETSI, September 1994

[7] GSM 2.88, Call Barring Supplementary Services - Stage 1
ETS 300 520, ETSI, September 1994

[8] GSM 3.14, Support of Dual Tone Multi Frequency Signalling via the GSM System
ETS 300 532, ETSI, April 1994

[9] GSM 3.40, Technical Realization of the Short Message Service Point-to-Point
ETS 300 536, ETSI, January 1996

[10] GSM 3.41, Technical Realization of Short Message Service Cell Broadcast
ETS 300 537, ETSI, June 1995

[11] GSM 3.81, Line Identification Supplementary Services - Stage 2
ETS 300 542, ETSI, February 1995

[12] GSM 3.82, Call Forwarding Supplementary Services - Stage 2
ETS 300 543, ETSI, February 1995

[13] GSM 3.83, Call Waiting and Call Hold Supplementary Services - Stage 2
ETS 300 544, ETSI, November 1994

[14] GSM 3.84, Multi Party Supplementary Services - Stage 2
ETS 300 545, ETSI, November 1994

[15] GSM 3.85, Closed User Group Supplementary Services - Stage 2
ETS 300 546, ETSI, January 1996

[16] GSM 3.86, Advice of Charge Supplementary Services - Stage 2
ETS 300 547, ETSI, March 1995

[17] GSM 3.88, Call Barring Supplementary Services - Stage 2
ETS 300 548, ETSI, November 1994

[18] GSM 4.01, MS-BSS Interface General Aspects and Principles
ETS 300 550, ETSI, September 1994

[18a] GSM 4.03, MS-BSS Interface Channel Structures and Access Capabilities
ETS 300 552, ETSI, September 1994

[19] GSM 4.05, Data Link Layer General Aspects
ETS 300 554, ETSI, September 1994

[20] GSM 4.06, MS-BSS Interface Data Link Layer Specification
ETS 300 555, ETSI, September 1994

[21] GSM 4.07, Mobile Radio Interface Signalling Layer 3 General Aspects
ETS 300 556, ETSI, February 1995

[22] GSM 4.08, Mobile Radio Interface Layer 3 Specification
ETS 300 557, ETSI, January 1996

[23] GSM 4.10, Mobile Radio Interface Layer 3 Supplementary Services Specification
General Aspects
ETS 300 558, ETSI, February 1995

[24] GSM 4.11, Point-to-Point Short Message Service Support on Mobile Radio Interface
ETS 300 559, ETSI, October 1995

[25] GSM 4.12, Short Message Service Cell Broadcast Support on Mobile Radio Interface
ETS 300 560, ETSI, January 1996

[26] GSM 4.80, Mobile Radio Interface Supplementary Services Specification Formats and Coding
ETS 300 564, ETSI, February 1995

[27] GSM 4.81, Line Identification Supplementary Services - Stage 3
ETS 300 565, ETSI, February 1995

[28] GSM 4.82, Call Forwarding Supplementary Services - Stage 3
ETS 300 566, ETSI, February 1995

[29] GSM 4.83, Call Waiting and Call Hold Supplementary Services - Stage 3
ETS 300 567, ETSI, February 1995

[30] GSM 4.84, Multi Party Supplementary Services - Stage 3
ETS 300 568, ETSI, February 1995

[31] GSM 4.85, Closed User Group Supplementary Services - Stage 3
ETS 300 569, ETSI, February 1995

[32] GSM 4.86, Advice of Charge Supplementary Services - Stage 3
ETS 300 570, ETSI, February 1995

[33] GSM 4.88, Call Barring Supplementary Services - Stage 3
ETS 300 571, ETSI, February 1995

[34] GSM 5.01, Physical Layer on the Radio Path General Description
ETS 300 573, ETSI, October 1995

[35] GSM 5.02, Multiplexing and Multiple Access on the Radio Path
ETS 300 574, ETSI, January 1996

[36] GSM 5.08, Radio Sub-system Link Control
ETS 300 578, ETSI, January 1996

[37] GSM 5.10, Radio Sub-system Synchronisation
ETS 300 579, ETSI, October 1995

[38] Service Access Point MMREG
6147.100.96.100; Condat GmbH

[39] Service Access Point MNCC
6147.101.96.100; Condat GmbH

[40] Service Access Point MNSS
6147.102.96.100; Condat GmbH

[41] Service Access Point MNSMS
6147.103.96.100; Condat GmbH

[42] Service Access Point MMCC
6147.104.97.100; Condat GmbH

[43] Service Access Point MMSS
6147.105.97.100; Condat GmbH

[44] Service Access Point MMSMS
6147.106.97.100; Condat GmbH

[45] Service Access Point RR
6147.107.97.100; Condat GmbH

[46] Service Access Point SIM
6147.108.97.100; Condat GmbH

[47] Service Access Point MPH
6147.109.96.100; Condat GmbH

[48] Service Access Point DL
6147.110.96.100; Condat GmbH

[49] Service Access Point MDL
6147.111.96.100; Condat GmbH

[50] Service Access Point PH
6147.112.97.100; Condat GmbH

[51] Service Access Point MMI
6147.113.96.100; Condat GmbH

[52] Message Sequence Charts CC
6147.200.97.100; Condat GmbH

[53] Message Sequence Charts SS
6147.201.97.100; Condat GmbH

[54] Message Sequence Charts SMS
6147.202.97.100; Condat GmbH

[55] Message Sequence Charts MM
6147.203.97.100; Condat GmbH

[56] Message Sequence Charts RR
6147.204.96.100; Condat GmbH

[57] Message Sequence Charts DL
6147.205.96.100; Condat GmbH

[58] Users Guide
6147.300.96.100; Condat GmbH

[59] Test Specification CC
6147.400.97.100; Condat GmbH

[60] Test Specification SS
6147.401.97.100; Condat GmbH

[61] Test Specification SMS
6147.402.97.100; Condat GmbH

[62] Test Specification MM
6147.403.97.100; Condat GmbH

[63] Test Specification RR
6147.404.97.100; Condat GmbH

[64] Test Specification DL
6147.405.97.100; Condat GmbH

[65] Test Specification CCD
6147.406.97.100; Condat GmbH

[66] SDL Specification CC
6147.500.97.100; Condat GmbH

[67] SDL Specification SS
6147.501.97.100; Condat GmbH

[68] SDL Specification SMS
6147.502.97.100; Condat GmbH

[69] SDL Specification MM
6147.503.97.100; Condat GmbH

[70] SDL Specification RR
6147.504.97.100; Condat GmbH

[71] SDL Specification DL
6147.505.97.100; Condat GmbH

[72] Message Specification CC
6147.600.97.100; Condat GmbH

[73] Message Specification SS
6147.601.97.100; Condat GmbH

[74] Message Specification SMS
6147.602.97.100; Condat GmbH

[75] Message Specification MM
6147.603.97.100; Condat GmbH

[76] Message Specification RR
6147.604.97.100; Condat GmbH

[77] Message Specification DL
6147.605.97.100; Condat GmbH

[78] Technical Documentation CC
6147.700.97.100; Condat GmbH

[79] Technical Documentation SS
6147.701.97.100; Condat GmbH

[80] Technical Documentation SMS
6147.702.97.100; Condat GmbH

[81] Technical Documentation MM
6147.703.97.100; Condat GmbH

[82] Technical Documentation RR
6147.704.97.100; Condat GmbH

[83] Technical Documentation DL
6147.705.97.100; Condat GmbH

[84] Technical Documentation CCD
6147.706.97.100; Condat GmbH

0.3 Abbreviations

AGCH	Access Grant Channel
BCCH	Broadcast Control Channel
BS	Base Station
BSIC	Base Station Identification Code
CBCH	Cell Broadcast Channel
CBQ	Cell Bar Qualify
CC	Call Control
CCCH	Common Control Channel
CCD	Condat Coder Decoder
CKSN	Ciphering Key Sequence Number
C/R	Command / Response
C1	Path Loss Criterion
C2	Reselection Criterion
DCCH	Dedicated Control Channel
DISC	Disconnect Frame
DL	Data Link Layer
DM	Disconnected Mode Frame
EA	Extension Bit Address Field
EL	Extension Bit Length Field
EMMI	Electrical Man Machine Interface
F	Final Bit
FACCH	Fast Associated Control Channel
FHO	Forced Handover
GP	Guard Period
GSM	Global System for Mobile Communication
HPLMN	Home Public Land Mobile Network
I	Information Frame
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
Kc	Authentication Key

L	Length Indicator
LAI	Location Area Information
LPD	Link Protocol Discriminator
M	More Data Bit
MCC	Mobile Country Code
MM	Mobility Management
MMI	Man Machine Interface
MNC	Mobile Network Code
MS	Mobile Station
NCC	National Colour Code
NECI	New Establishment Causes included
N(R)	Receive Number
N(S)	Send Number
OTD	Observed Time Difference
P	Poll Bit
PCH	Paging Channel
PDU	Protocol Description Unit
P/F	Poll / Final Bit
PL	Physical Layer
PLMN	Public Land Mobile Network
RACH	Random Access Channel
REJ	Reject Frame
RNR	Receive Not Ready Frame
RR	Radio Resource Management
RR	Receive Ready Frame
RTD	Real Time Difference
SABM	Set Asynchronous Balanced Mode
SACCH	Slow Associated Control Channel
SAP	Service Access Point
SAPI	Service Access Point Identifier
SDCCH	Slow Dedicated Control Channel
SIM	Subscriber Identity Module
SMS	Short Message Service
SMSCB	Short Message Service Cell Broadcast
SS	Supplementary Services
TCH	Traffic Channel
TCH/F	Traffic Channel Full Rate
TCH/H	Traffic Channel Half Rate
TDMA	Time Division Multiple Access
TMSI	Temporary Mobile Subscriber Identity
UA	Unnumbered Acknowledgement Frame
UI	Unnumbered Information Frame
VPLMN	Visiting Public Land Mobile Network
V(A)	Acknowledgement State Variable

V(R)	Receive State Variable
V(S)	Send State Variable

0.4 Terms

Entity: Program which executes the functions of a layer

Message: A message is a data unit which is transferred between the entities of the same layer (peer-to-peer) of the mobile and infrastructure side. Message is used as a synonym to protocol data unit (PDU). A message may contain several information elements.

Primitive: A primitive is a data unit which is transferred between layers on one component (mobile station or infrastructure). The primitive has an operation code which identifies the primitive and its parameters.

Service Access Point: A Service Access Point is a data interface between two layers on one component (mobile station or infrastructure).

1 Overview

The Protocol Stacks are used to define the functionality of the GSM protocols for interfaces. The GSM specifications are normative when used to describe the functionality of interfaces, but the stacks and the subdivision of protocol layers does not imply or restrict any implementation.

The base of the Protocol Stack rests on the physical layer.

The Data Link Layer (DL) is used to handle an acknowledged connection between mobile and base station. The LAPDm protocol is used.

Radio Resource (RR) manages the resources of the air-interface. That means configuration of physical layer, cell selection and cell reselection, data transfer, RR-Connection handling.

Mobility Management (MM) handles registration aspects for the mobile station. It detects changes of location areas and updates a mobile station in the new location area.

Call Control (CC) provides the call functionality. This includes call establishment, call maintenance procedures like Hold, Retrieve or Modify, and call disconnection.

Supplementary Services (SS) handles all call independent supplementary services like call forwarding or call barring.

Short Message Services (SMS) is used for sending and receiving point-to-point short messages. Additionally the reception of cell broadcast short messages is included.

The man machine interface (MMI) is the interface to the user. Normally it is connected with a keypad as input device and a display as output device.

Between the several entities data interfaces are defined. These data interfaces are called Service Access Points (SAPs), indicating that an upper layer uses the services of a lower layer.

The GSM specification do not set out any implementation of the Protocol Stack. The following diagrams show the implementation described in all these documents for the mobile station. All entities except the Man Machine Interface and Physical Layer are implemented as part of the Protocol Stack.

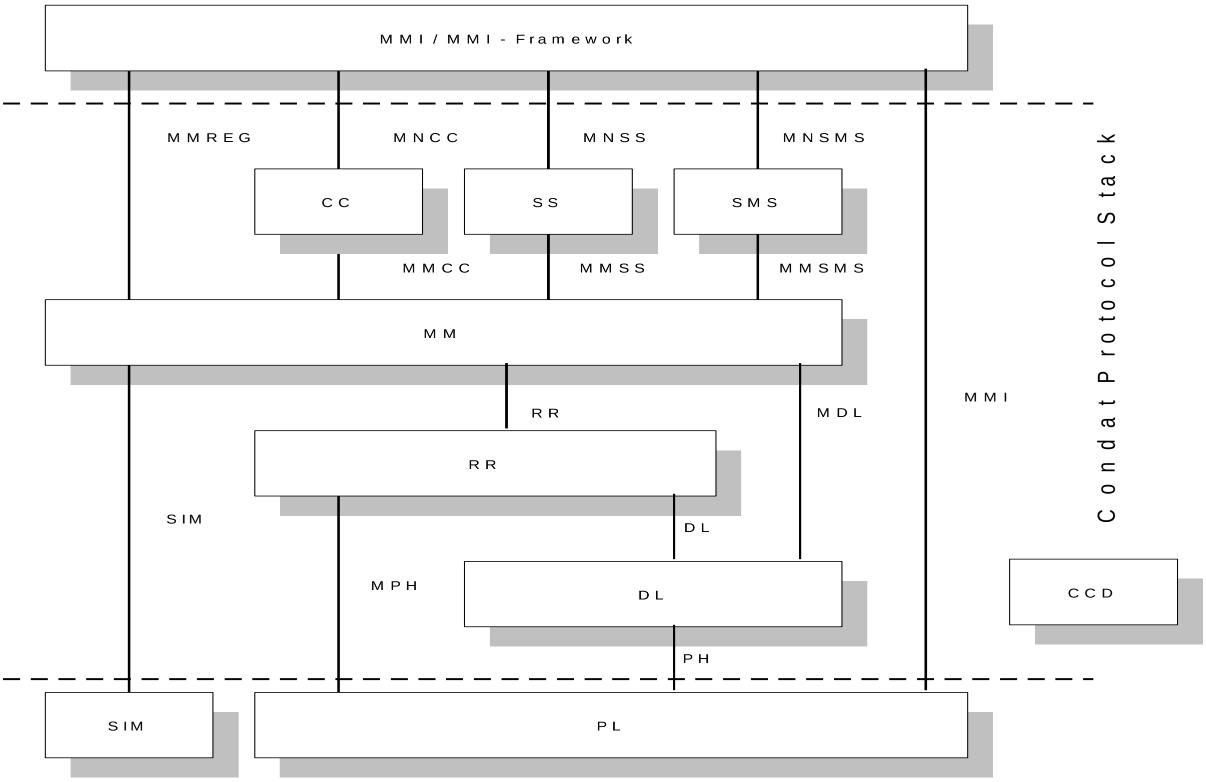

Figure 1: Mobile-station protocol architecture

This document describes the tests for Short Message Service (SMS).

2 Parameters

/*
Note: The following parameters section is also used in the GSMS test document.
*/

/* declaration */

DECLARATION (RP_ACK_DLNK)
DECLARATION (RP_ACK_DLNK_REF_ERR)
DECLARATION (RP_ACK_ULNK)
DECLARATION (RP_ERR_ULNK_RESP)
DECLARATION (RP_SMMA)
DECLARATION (RP_SMMA_REP)
DECLARATION (RP_ACK_SMMA)
DECLARATION (RP_ACK_SMMA_REP)
DECLARATION (RP_ACK_RESP)
DECLARATION (RP_ERR_RESP)
DECLARATION (RP_ACK_DLVR_REP)
DECLARATION (RP_ERR_DLVR_REP)
DECLARATION (RP_CAUSE_CONGESTION)
DECLARATION (RP_CAUSE_MEM_CAP_EXCEEDED)
DECLARATION (RP_CAUSE_PROTOCOL_ERROR)
DECLARATION (RP_CAUSE_TEMP_FAILURE)
DECLARATION (RP_CMD_STAT_REQ)
DECLARATION (RP_CMD_ENQ)
DECLARATION (RP_CMD_CANCEL_REP)
DECLARATION (RP_CMD_DEL)
DECLARATION (RP_ERR_CONGESTION)
DECLARATION (RP_ERR_MEM_CAP_EXC)
DECLARATION (RP_ERR_PROTOCOL)
DECLARATION (RP_ERR_PROTOCOL_SECOND)
DECLARATION (RP_ERR_PROTOCOL_AA)
DECLARATION (RP_ERR_TEMP_FAILURE)
DECLARATION (RP_ERROR_CONGESTION)
DECLARATION (RP_ERROR_MEM_CAP_EXC)
DECLARATION (RP_ERROR_PROTOCOL)
DECLARATION (RP_ERROR_TEMP_FAILURE)
DECLARATION (RP_SCA_12345)
DECLARATION (RP_SCA_23456)
DECLARATION (RP_SCA_0811112222)

DECLARATION (RP_DATA_CMD_STAT_REQ)
DECLARATION (RP_DATA_CMD_ENQ)
DECLARATION (RP_DATA_CMD_CANCEL_REP)
DECLARATION (RP_DATA_CMD_DEL)
DECLARATION (RP_DATA_DELIVER)
DECLARATION (RP_DATA_DELIVER_2)
DECLARATION (RP_DATA_DELIVER_7CL0)
DECLARATION (RP_DATA_DELIVER_7CL0_DEF)
DECLARATION (RP_DATA_DELIVER_7CL0_GDC)
DECLARATION (RP_DATA_DELIVER_8CL0_DEF)
DECLARATION (RP_DATA_DELIVER_8CL0_GDC)
DECLARATION (RP_DATA_DELIVER_16CL0_GDC)
DECLARATION (RP_DATA_DELIVER_7CL0L)
DECLARATION (RP_DATA_DELIVER_7CL0_43)
DECLARATION (RP_DATA_DELIVER_7CL1)
DECLARATION (RP_DATA_DELIVER_7CL1_42)
DECLARATION (RP_DATA_DELIVER_7CL1_43)
DECLARATION (RP_DATA_DELIVER_7CL1_43S)
DECLARATION (RP_DATA_DELIVER_7CL1_43O)

DECLARATION (RP_DATA_DELIVER_7CL2)
DECLARATION (RP_DATA_DELIVER_7CL2_43)
DECLARATION (RP_DATA_DELIVER_7CL3)
DECLARATION (RP_DATA_DELIVER_7CL3_43)
DECLARATION (RP_DATA_DELIVER_7DEF)
DECLARATION (RP_DATA_DELIVER_121_A)
DECLARATION (RP_DATA_DELIVER_121_B)
DECLARATION (RP_DATA_DELIVER_121_C)
DECLARATION (RP_DATA_DELIVER_EMPTY)
DECLARATION (RP_DATA_DELIVER_7CL2_SAT1)
DECLARATION (RP_DATA_DELIVER_7CL2_SAT2)
DECLARATION (RP_DATA_DELIVER_7CL1_SAT3)
DECLARATION (RP_DATA_DELIVER_8CL2_SAT1)
DECLARATION (RP_DATA_DELIVER_8CL2_SAT2)
DECLARATION (RP_DATA_STATUS_REP)
DECLARATION (RP_DATA_STATUS_REQ)
DECLARATION (RP_DATA_SUBMIT_ABS)
DECLARATION (RP_DATA_SUBMIT_PID5F_ABS)
DECLARATION (RP_DATA_SUBMIT_REP)
DECLARATION (RP_DATA_SUBMIT_MO)
DECLARATION (RP_DATA_SUBMIT_DA)
DECLARATION (RP_DATA_SUBMIT_SCA)
DECLARATION (RP_DATA_SUBMIT_DA_SCA)
DECLARATION (RP_DATA_SUBMIT_7DEF)
DECLARATION (RP_DATA_SUBMIT_7DEF_DA)
DECLARATION (RP_DATA_SUBMIT_7DEF_SCA)
DECLARATION (RP_DATA_SUBMIT_7DEF_DA_SCA)

DECLARATION (RP_DELIVER)
DECLARATION (RP_DELIVER_7CL0)
DECLARATION (RP_DELIVER_7CL0_DEF)
DECLARATION (RP_DELIVER_7CL0_GDC)
DECLARATION (RP_DELIVER_8CL0_DEF)
DECLARATION (RP_DELIVER_8CL0_GDC)
DECLARATION (RP_DELIVER_16CL0_GDC)
DECLARATION (RP_DELIVER_7CL0L)
DECLARATION (RP_DELIVER_7CL0_43)
DECLARATION (RP_DELIVER_7CL1)
DECLARATION (RP_DELIVER_7CL1_42)
DECLARATION (RP_DELIVER_7CL1_43)
DECLARATION (RP_DELIVER_7CL1_43S)
DECLARATION (RP_DELIVER_7CL1_43O)
DECLARATION (RP_DELIVER_7CL2)
DECLARATION (RP_DELIVER_121_A)
DECLARATION (RP_DELIVER_121_B)
DECLARATION (RP_DELIVER_121_C)
DECLARATION (RP_DELIVER_EMPTY)
DECLARATION (RP_DELIVER_7CL2_SAT1)
DECLARATION (RP_DELIVER_7CL2_SAT2)
DECLARATION (RP_DELIVER_7CL1_SAT3)
DECLARATION (RP_DELIVER_8CL2_SAT1)
DECLARATION (RP_DELIVER_8CL2_SAT2)
DECLARATION (RP_DELIVER_7CL2_43)
DECLARATION (RP_DELIVER_7CL3)
DECLARATION (RP_DELIVER_7CL3_43)
DECLARATION (RP_DELIVER_7DEF)
DECLARATION (RP_STATUS_REP)
DECLARATION (RP_STATUS_REQ)
DECLARATION (RP_SUBMIT_ABS)
DECLARATION (RP_SUBMIT_PID5F_ABS)
DECLARATION (RP_SUBMIT_REP)

```
DECLARATION (RP_SUBMIT_MO)
DECLARATION (RP_SUBMIT_DA)
DECLARATION (RP_SUBMIT_SCA)
DECLARATION (RP_SUBMIT_DA_SCA)
DECLARATION (RP_SUBMIT_7DEF)
DECLARATION (RP_SUBMIT_7DEF_DA)
DECLARATION (RP_SUBMIT_7DEF_SCA)
DECLARATION (RP_SUBMIT_7DEF_DA_SCA)

DECLARATION (RP_UD_CMD_STAT_REQ)
DECLARATION (RP_UD_CMD_ENQ)
DECLARATION (RP_UD_CMD_CANCEL_REP)
DECLARATION (RP_UD_CMD_DEL)
DECLARATION (RP_UD_DELIVER)
DECLARATION (RP_UD_DELIVER_7CL0)
DECLARATION (RP_UD_DELIVER_7CL0_DEF)
DECLARATION (RP_UD_DELIVER_7CL0_GDC)
DECLARATION (RP_UD_DELIVER_8CL0_DEF)
DECLARATION (RP_UD_DELIVER_8CL0_GDC)
DECLARATION (RP_UD_DELIVER_16CL0_GDC)
DECLARATION (RP_UD_DELIVER_7CL0L)
DECLARATION (RP_UD_DELIVER_7CL0_43)
DECLARATION (RP_UD_DELIVER_7CL1)
DECLARATION (RP_UD_DELIVER_7CL1_42)
DECLARATION (RP_UD_DELIVER_7CL1_43)
DECLARATION (RP_UD_DELIVER_7CL1_43S)
DECLARATION (RP_UD_DELIVER_7CL1_43O)
DECLARATION (RP_UD_DELIVER_7CL2)
DECLARATION (RP_UD_DELIVER_121_A)
DECLARATION (RP_UD_DELIVER_121_B)
DECLARATION (RP_UD_DELIVER_121_C)
DECLARATION (RP_UD_DELIVER_EMPTY)
DECLARATION (RP_UD_DELIVER_7CL2_SAT1)
DECLARATION (RP_UD_DELIVER_7CL2_SAT2)
DECLARATION (RP_UD_DELIVER_7CL1_SAT3)
DECLARATION (RP_UD_DELIVER_8CL2_SAT1)
DECLARATION (RP_UD_DELIVER_8CL2_SAT2)
DECLARATION (RP_UD_DELIVER_7CL2_43)
DECLARATION (RP_UD_DELIVER_7CL3)
DECLARATION (RP_UD_DELIVER_7CL3_43)
DECLARATION (RP_UD_DELIVER_7DEF)
DECLARATION (RP_UD_STATUS_REP)
DECLARATION (RP_UD_STATUS_REQ)
DECLARATION (RP_UD_SUBMIT_ABS)
DECLARATION (RP_UD_SUBMIT_PID5F_ABS)
DECLARATION (RP_UD_SUBMIT_REP)
DECLARATION (RP_UD_SUBMIT_MO)
DECLARATION (RP_UD_SUBMIT_DA)
DECLARATION (RP_UD_SUBMIT_7DEF)
DECLARATION (RP_UD_SUBMIT_7DEF_DA)

DECLARATION (TPDU_SUBMIT_ABS)
DECLARATION (TPDU_SUBMIT_MO)
DECLARATION (TPDU_SUBMIT_DA)
DECLARATION (TPDU_SUBMIT_7DEF)
DECLARATION (TPDU_SUBMIT_7DEF_DA)
DECLARATION (TPDU_DELIVER_7DEF)
DECLARATION (TPDU_DELIVER_7CL0)
DECLARATION (TPDU_DELIVER_7CL0_DEF)
DECLARATION (TPDU_DELIVER_7CL0_GDC)
DECLARATION (TPDU_DELIVER_8CL0_DEF)
DECLARATION (TPDU_DELIVER_8CL0_GDC)
DECLARATION (TPDU_DELIVER_16CL0_GDC)
DECLARATION (TPDU_DELIVER_7CL0L)
DECLARATION (TPDU_DELIVER_7CL1)
DECLARATION (TPDU_DELIVER_7CL2)
DECLARATION (TPDU_DELIVER_7CL3)
DECLARATION (TPDU_DELIVER_7CL0_43)
DECLARATION (TPDU_DELIVER_7CL1_43)
DECLARATION (TPDU_DELIVER_7CL2_43)
DECLARATION (TPDU_DELIVER_7CL3_43)
DECLARATION (TPDU_DELIVER_7CL1_42)
DECLARATION (TPDU_DELIVER_7CL1_43S)
DECLARATION (TPDU_DELIVER_7CL1_43O)
DECLARATION (TPDU_DELIVER_121_A)
DECLARATION (TPDU_DELIVER_121_B)
DECLARATION (TPDU_DELIVER_121_C)
DECLARATION (TPDU_DELIVER_EMPTY)
DECLARATION (TPDU_DELIVER_7CL2_SAT1)
DECLARATION (TPDU_DELIVER_7CL2_SAT2)
DECLARATION (TPDU_DELIVER_7CL1_SAT3)
DECLARATION (TPDU_DELIVER_8CL2_SAT1)
DECLARATION (TPDU_DELIVER_8CL2_SAT2)

DECLARATION (TPDU_COMMAND_STAT_REQ)
DECLARATION (TPDU_COMMAND_CANCEL_REP)
DECLARATION (TPDU_COMMAND_ENQ)
DECLARATION (TPDU_COMMAND_DEL)

DECLARATION (RP_CAUSE_SEM_INC)
DECLARATION (RP_ERROR_SEM_INC)
DECLARATION (RP_ERR_SEM_INC)
DECLARATION (TPDU_FCS_UNSPEC)
DECLARATION (RP_UD_FCS_UNSPEC)
DECLARATION (RP_ERROR_FCS_UNSPEC)
DECLARATION (TPDU_DLVR_REP_ACK)
DECLARATION (RP_UD_DLVR_REP_ACK)
DECLARATION (RP_ACKNL_DLVR_REP)
DECLARATION (TPDU_DLVR_REP_ERR)
DECLARATION (RP_UD_DLVR_REP_ERR)
DECLARATION (RP_ERROR_DLVR_REP)
DECLARATION (SIMREC_SMSS_MSG_REF)
DECLARATION (SIMREC_SMSS_MSG_REF_N2)
DECLARATION (SIM_SMS_MT_DELIVER_7DEF)

DECLARATION (SMS_SDU_EMPTY)
DECLARATION (SMS_SDU_MO)
DECLARATION (SMS_SDU_MO_ABS)
DECLARATION (SMS_SDU_MT)
DECLARATION (SMS_SDU_MT_7CL1)
DECLARATION (SMS_SDU_SBM_DEF)
DECLARATION (SMS_SDU_SBM_DEF_X)
/*DECLARATION (SMS_SDU_SBM_DEF_BUF)*/
DECLARATION (SMS_SDU_SUBMIT_ABS)
DECLARATION (SMS_SDU_DELIVER_7CL0)
DECLARATION (SMS_SDU_DELIVER_7CL0_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL0_DEF)
DECLARATION (SMS_SDU_DELIVER_7CL0_DEF_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL0_GDC)
DECLARATION (SMS_SDU_DELIVER_7CL0_GDC_BUF)
DECLARATION (SMS_SDU_DELIVER_8CL0_DEF)
DECLARATION (SMS_SDU_DELIVER_8CL0_DEF_BUF)
DECLARATION (SMS_SDU_DELIVER_8CL0_GDC)
DECLARATION (SMS_SDU_DELIVER_8CL0_GDC_BUF)
DECLARATION (SMS_SDU_DELIVER_16CL0_GDC)
DECLARATION (SMS_SDU_DELIVER_16CL0_GDC_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL0L)
DECLARATION (SMS_SDU_DELIVER_7CL0L_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL1)
DECLARATION (SMS_SDU_DELIVER_7CL1_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL2)
DECLARATION (SMS_SDU_DELIVER_7CL2_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL3)
DECLARATION (SMS_SDU_DELIVER_7CL3_BUF)
DECLARATION (SMS_SDU_DELIVER_7DEF)
DECLARATION (SMS_SDU_DELIVER_7DEF_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL0_43)
DECLARATION (SMS_SDU_DELIVER_7CL0_43_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL1_43)
DECLARATION (SMS_SDU_DELIVER_7CL1_43_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL2_43)
DECLARATION (SMS_SDU_DELIVER_7CL2_43_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL3_43)
DECLARATION (SMS_SDU_DELIVER_7CL3_43_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL1_42)
DECLARATION (SMS_SDU_DELIVER_7CL1_42_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL1_43S)
DECLARATION (SMS_SDU_DELIVER_7CL1_43S_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL1_43O)
DECLARATION (SMS_SDU_DELIVER_7CL1_43O_BUF)
DECLARATION (SMS_SDU_DELIVER_121_A)
DECLARATION (SMS_SDU_DELIVER_121_A_BUF)
DECLARATION (SMS_SDU_DELIVER_121_B)
DECLARATION (SMS_SDU_DELIVER_121_B_BUF)
DECLARATION (SMS_SDU_DELIVER_121_C)
DECLARATION (SMS_SDU_DELIVER_121_C_BUF)
DECLARATION (SMS_SDU_MO_CHANGE)
DECLARATION (SMS_SDU_MO_CHANGE_BUF)

DECLARATION (SMS_SDU_COMMAND_STAT_REQ)
DECLARATION (SMS_SDU_COMMAND_ENQ)
DECLARATION (SMS_SDU_COMMAND_CANCEL_REP)
DECLARATION (SMS_SDU_COMMAND_DEL)

DECLARATION (SMS_SDU_STATUS_REP)

DECLARATION (SMS_SDU_DLVR_REP_ACK)
DECLARATION (SMS_SDU_DLVR_REP_ERR)

DECLARATION (SMS_SDU_DELIVER_7CL2_SAT1)
DECLARATION (SMS_SDU_DELIVER_7CL2_SAT1_BUF)
DECLARATION (SMS_SDU_DELIVER_7CL1_SAT3)
DECLARATION (SMS_SDU_DELIVER_7CL1_SAT3_BUF)

/* SIM Toolkit Commands*/
DECLARATION (ENVELOPE_SMS_1)
DECLARATION (ENVELOPE_SMS_1_CMD)
DECLARATION (ENVELOPE_SMS_2)
DECLARATION (ENVELOPE_SMS_2_CMD)
DECLARATION (ENVELOPE_SMS_3)
DECLARATION (ENVELOPE_SMS_3_CMD)
DECLARATION (ENVELOPE_SMS_4)
DECLARATION (ENVELOPE_SMS_4_CMD)
DECLARATION (ENVELOPE_SMS_121_C)
DECLARATION (ENVELOPE_SMS_121_C_CMD)
DECLARATION (STK_CMD_EMPTY)
DECLARATION (STK_CMD_EMPTY_CMD)
DECLARATION (STK_CMD_TPDU1)
DECLARATION (STK_CMD_TPDU2)
DECLARATION (STK_CMD_TPDU_7BIT)
DECLARATION (STK_CMD_TPDU_8BIT)

DECLARATION (TPDU_STATUS_REP)

DECLARATION (CPHS_VMW_DATA)
DECLARATION (IMSI_NORMAL)
DECLARATION (IMSI_ONE2ONE)

/* Bytes*/
BYTE            BYTE_00             0x00
BYTE            BYTE_55             0x55
BYTE            BYTE_AA             0xAA

BYTE            REQ_ID_0            0

/* ti*/
BYTE            TI_MO               0x00
BYTE            TI_MO_TO_MS         0x08
BYTE            TI_MT               0x00
BYTE            TI_MT_FROM_MS       0x08
BYTE            TI_MT_2             0x01
BYTE            TI_MT_2_FROM_MS     0x09

/* length*/
BYTE            LEN_0               0
BYTE            LEN_1               1
BYTE            LEN_2               2
BYTE            LEN_3               3
BYTE            LEN_5               5
BYTE            LEN_6               6
BYTE            LEN_9               9
BYTE            LEN_12              12
BYTE            LEN_176             176

BYTE            LENGTH_5            5
BYTE            LENGTH_6            6
BYTE            LENGTH_9            9
BYTE            LENGTH_10           10
BYTE            LENGTH_11           11
BYTE            LENGTH_18           18
BYTE            LENGTH_160          160
BYTE            LENGTH_SMS          176

/* msg ref*/
BYTE            MSG_REF_00          0x00
BYTE            MSG_REF_01          0x01
BYTE            MSG_REF_02          0x02
BYTE            MSG_REF_AA          0xAA
BYTE            MSG_REF_AB          0xAB
BYTE            MSG_REF_AC          0xAC

BYTE            TP_MR_3             3
BYTE            TP_MR_3N1           (TP_MR_3+1)
BYTE            TP_MR_3N2           (TP_MR_3+2)
BYTE            TP_MR_3N3           (TP_MR_3+3)
BYTE            TP_MR_3_1           (TP_MR_3-1)

/* reply path*/
BYTE            REPLY_PATH_0        0x00
BYTE            REPLY_PATH_1        0x01
```

```
/* more messages*/
BYTE        MORE_MSG_0            0x00
BYTE        MORE_MSG_1            0x01

/* dcs*/
BYTE        DCS_DEF               0x00     /* no class, GSM alphabet*/
BYTE        DCS_CL0_GDC0          0x10     /* class 0, GSM alphabet*/
BYTE        DCS_CL0_GDC1          0x30     /* class 0, GSM alphabet, compressed*/
BYTE        DCS_CL0_GDC2          0x14     /* class 0, 8 bit data*/
BYTE        DCS_CL0_GDC4          0x18     /* class 0, UCS2 data*/
BYTE        DCS_CL0_DEF           0xF0     /* class 0, GSM alphabet*/
BYTE        DCS_CL0_8BIT          0xF4     /* class 0, 8 bit data*/
BYTE        DCS_CL1_GDC0          0x11     /* class 1, GSM alphabet*/
BYTE        DCS_CL1_GDC1          0x31     /* class 1, GSM alphabet, compressed*/
BYTE        DCS_CL1_GDC2          0x15     /* class 1, 8 bit data*/
BYTE        DCS_CL1_GDC4          0x19     /* class 1, UCS2 data*/
BYTE        DCS_CL1_DEF           0xF1     /* class 1, GSM alphabet*/
BYTE        DCS_CL1_8BIT          0xF5     /* class 1, 8 bit data*/
BYTE        DCS_CL2_GDC0          0x12     /* class 2, GSM alphabet*/
BYTE        DCS_CL2_GDC1          0x32     /* class 2, GSM alphabet, compressed*/
BYTE        DCS_CL2_GDC2          0x16     /* class 2, 8 bit data*/
BYTE        DCS_CL2_GDC3          0x36     /* class 2, 8 bit data, compressed*/
BYTE        DCS_CL2_GDC4          0x1A     /* class 2, UCS2 data*/
BYTE        DCS_CL2_DEF           0xF2     /* class 2, GSM alphabet*/
BYTE        DCS_CL2_8BIT          0xF6     /* class 2, 8 bit data*/
BYTE        DCS_CL3_GDC0          0x13     /* class 3, GSM alphabet*/
BYTE        DCS_CL3_GDC1          0x33     /* class 3, GSM alphabet, compressed*/
BYTE        DCS_CL3_GDC2          0x17     /* class 3, 8 bit data*/
BYTE        DCS_CL3_GDC4          0x1B     /* class 3, UCS2 data*/
BYTE        DCS_CL3_DEF           0xF3     /* class 3, GSM alphabet*/
BYTE        DCS_CL3_8BIT          0xF7     /* class 3, 8 bit data*/
BYTE        DCS_MWI_DISCD         0xC0     /* message waiting indication, discard, GSM alphabet*/
BYTE        DCS_MWI_STR_DEF       0xD0     /* message waiting indication, store, GSM alphabet*/
BYTE        DCS_MWI_STR_UCS2      0xE0     /* message waiting indication, store, UCS2 data*/

/* Protocol Identifier*/
BYTE        PID_0                 0x00

/* message types*/
BYTE        MSG_MO_1              1
BYTE        MSG_MT_1              4

BYTE        MSG_TYPE_02           0x02
BYTE        MSG_TYPE_04           0x04
BYTE        MSG_TYPE_06           0x06
BYTE        MSG_TYPE_1D           0x1D

BYTE        SMS_CONDX_OVR_UNDEF                3

/* composed cause values */
SHORT   SMS_TX_CS_MSG_NOT_COMP        (0x4000 | (SMSCP_ORIGINATING_ENTITY<<8) | SMS_CP_CS_MSG_NOT_COMP)
SHORT   SMS_TX_CS_INFO_NON_EXIST      (0x4000 | (SMSCP_ORIGINATING_ENTITY<<8) | SMS_CP_CS_INFO_NON_EXIST)
SHORT   SMS_RX_CS_NETWORK_FAILURE     (0x0000 | (SMSCP_ORIGINATING_ENTITY<<8) | SMS_CP_CS_NETWORK_FAILURE)
SHORT   SMS_RX_CS_CONGESTION          (0x0000 | (SMSRP_ORIGINATING_ENTITY<<8) | SMS_RP_CS_CONGESTION)
SHORT   SMS_TX_CS_PROTOCOL_ERROR      (0x4000 | (SMSRP_ORIGINATING_ENTITY<<8) | SMS_RP_CS_PROTOCOL_ERROR)
SHORT   SMS_RX_CS_TEMP_FAILURE        (0x0000 | (SMSRP_ORIGINATING_ENTITY<<8) | SMS_RP_CS_TEMP_FAILURE)

/* SIM record*/
SHORT       SIM_RECORD_0          0
SHORT       SIM_RECORD_1          1
SHORT       SIM_RECORD_2          2
SHORT       SIM_RECORD_3          3
SHORT           SIM_RECORD_4        4
SHORT           SIM_RECORD_5        5

/* offset */
SHORT           OFFSET_0            0
SHORT           OFFSET_1            1

/* Timezone */
BYTE            TIMEZONE_GMT_PLS_1HR            0x40
BYTE            TIMEZONE_GMT                    0x00

/* CPHS Voice Message Waiting Flag (Byte 1) */
BYTE            CPHS_VMW_BYTE1_L1W              0x5A

/* Definitions for EM */
LONG            Bitm_L              0x10100
LONG            Bitm_H              0x0000
BYTE            EM_ENTITY           0x07

/* Definitions for repeated usage of bytes sequences */

#define TO_BE_IGNORED                       /* 5 bytes which must be ignored */\
        0x5A, 0xA5, 0x0F, 0xF0, 0xFE

#define RP_ADDR_12345\
        0x04, 0x91, 0x21, 0x43, 0xF5

#define RP_ADDR_0811112222                  /* 7 bytes */\
        0x06, 0x81, 0x80, 0x11, 0x11, 0x22, 0x22

#define RP_ADDR_23456\
        0x04, 0xA1, 0x32, 0x54, 0xF6

#define TP_ADDR_SBM                         /* 5 bytes */\
        LENGTH_6, 0x91, 0x56, 0x34, 0x12

#define TP_ADDR_SBM1                        /* 5 bytes */\
        LENGTH_5, 0xA1, 0x89, 0x67, 0xF5

#define TP_ADDR_SBM2                        /* 7 bytes */\
        LENGTH_9, 0x80, 0x00, 0x89, 0x67, 0x45, 0xF3

#define TP_ADDR_DLV                         /* 5 bytes */\
        LENGTH_6, 0x81, 0x89, 0x67, 0x45

#define TP_ADDR_DLV1                        /* 5 bytes */\
        LENGTH_6, 0x81, 0x21, 0x43, 0x65

#define TP_ADDR_DLV_LONG                    /* 8 bytes */\
        LENGTH_10, 0x81, 0x10, 0x32, 0x89, 0x67, 0x45, 0xF8

#define TP_ADDR_121_SPEC                    /* 7 bytes */\
        LENGTH_9, 0xD0, 0x3E, 0x78, 0x59, 0x3E, 0x07

#define TP_ADDR_EMPTY                       /* 2 bytes */\
        0, 0x80

#define SM7_ABCDEFGHI                       /* 9 bytes */\
        LENGTH_9, 0x41, 0xE1, 0x90, 0x58, 0x34, 0x1E, 0x91, 0x49

#define TEXT7_RSTUVWXYZ                     /* 8 bytes */\
        0xD2, 0x29, 0xB5, 0x6A, 0xBD, 0x62, 0xB3, 0x5A

#define SM7_RSTUVWXYZ                       /* 9 bytes */\
        LENGTH_9, TEXT7_RSTUVWXYZ

#define TEXT8_ABCDEFGHI                     /* 9 bytes */\
        0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48, 0x49

#define SM8_ABCDEFGHI                       /* 10 bytes */\
        LENGTH_9, TEXT8_ABCDEFGHI
```

```
#define SM16_ABCDEFGHI                     /* 19 bytes */\
            LENGTH_18,\
            0x01, 0x00, 0x00, 0x42, 0x00, 0x43, 0x00, 0x44, 0x00, 0x45,\
            0x00, 0x46, 0x00, 0x47, 0x00, 0x48, 0x00, 0x49

#define SM7_LONG                           /* 141 bytes */\
            LENGTH_160,\
            0x54, 0x74, 0x7a, 0x0e, 0x4a, 0xcf, 0x41, 0x74, 0x74, 0x19, 0x44, 0x2e, 0x9b, 0xc3, \
            0x75, 0x36, 0x1d, 0x44, 0x2f, 0xcf, 0xe9, 0xa0, 0x76, 0x79, 0x3e, 0x0f, 0x9f, 0xcb, \
            0x80, 0x80, 0x60, 0x40, 0x28, 0x18, 0x0e, 0x88, 0x84, 0x62, 0xc1, 0x68, 0x38, 0x1e, \
            0x90, 0x88, 0x64, 0x42, 0xa9, 0x58, 0x2e, 0x98, 0x8c, 0xc6, 0xc5, 0xe9, 0x78, 0x3e, \
            0xa0, 0x90, 0x68, 0x44, 0x2a, 0x99, 0x4e, 0xa8, 0x94, 0x6a, 0xc5, 0x6a, 0xb9, 0x5e, \
            0xb0, 0x98, 0x6c, 0x46, 0xab, 0xd9, 0x6e, 0xb8, 0x9c, 0x6e, 0xc7, 0xeb, 0xf9, 0x7e, \
            0xc0, 0xa0, 0x70, 0x48, 0x2c, 0x1a, 0x8f, 0xc8, 0xa4, 0x72, 0xc9, 0x6c, 0x3a, 0x9f, \
            0xd0, 0xa8, 0x74, 0x4a, 0xad, 0x5a, 0xaf, 0xd8, 0xac, 0x76, 0xcb, 0xed, 0x7a, 0xbf, \
            0xe0, 0xb0, 0x78, 0x4c, 0x2e, 0x9b, 0xcf, 0xe8, 0xb4, 0x7a, 0xcd, 0x6e, 0xbb, 0xdf, \
            0xf0, 0xb8, 0x7c, 0x4e, 0xaf, 0xdb, 0xef, 0xf8, 0xbc, 0x7e, 0xcf, 0xef, 0xfb, 0xff

#define TIME_GMT\
            0x89, 0x21, 0x20, 0x10, 0x25, 0x31, BYTE_00

#define TIME_GMT1\
            0x89, 0x80, 0x13, 0x10, 0x25, 0x31, 0x01

#define TIME_GMT_PLS_1HR\
            0x89, 0x10, 0x70, 0x21, 0x43, 0x65, TIMEZONE_GMT_PLS_1HR

#define MO_INIT                            /* 18 bytes */\
            SMS_SUBMIT,\
            NOT_PRESENT_8BIT,\
            TP_ADDR_SBM,\
            SMS_PID_SM_TYPE_0, DCS_CL2_DEF,\
            SM7_ABCDEFGHI
SHORT       BITLEN_MO_INIT                 144

#define MO_ABS_INIT                        /* 25 bytes */\
            (SMS_SUBMIT | 0x18),           /* VP-ABS is set */\
            0x08,\
            TP_ADDR_SBM1,\
            SMS_PID_SM_TYPE_0, DCS_CL0_DEF,\
            TIME_GMT_PLS_1HR,\
            SM7_RSTUVWXYZ
SHORT       BITLEN_SUBMIT_ABS              200

#define MT_INIT                            /* 24 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_SBM1,\
            SMS_PID_SM_TYPE_0, DCS_CL2_DEF,\
            TIME_GMT1,\
            SM7_ABCDEFGHI
SHORT       BITLEN_MT_INIT                 192

#define SUBMIT_ABS                         /* 25 bytes */\
            0x19,                          /* SUBMIT, VP-ABS */\
            TP_MR_3N1,\
            TP_ADDR_SBM,\
            SMS_PID_SM_TYPE_0, DCS_DEF,\
            TIME_GMT_PLS_1HR,\
            SM7_ABCDEFGHI
SHORT       BITLEN_SUBMIT_ABS              200
```

```
#define DELIVER_7DEF                          /* 24 bytes */\
            SMS_DELIVER,\
            TP_ADDR_DLV,\
            SMS_PID_SM_TYPE_0, DCS_DEF,\
            TIME_GMT_PLS_1HR,\
            SM7_ABCDEFGHI
SHORT       BITLEN_DELIVER_7DEF               192

#define DELIVER_7CL0                          /* 24 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_DLV,\
            SMS_PID_SM_TYPE_0, DCS_CL0_GDC0,\
            TIME_GMT_PLS_1HR,\
            SM7_ABCDEFGHI
SHORT       BITLEN_DELIVER_7CL0               192

#define DELIVER_7CL0_DEF                      /* 24 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_DLV,\
            PID_0, DCS_CL0_DEF,\
            TIME_GMT_PLS_1HR,\
            SM7_ABCDEFGHI
SHORT       BITLEN_DELIVER_7CL0_DEF           192

#define DELIVER_7CL0_GDC                      /* 24 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_DLV,\
            PID_0, DCS_CL0_GDC0,\
            TIME_GMT,\
            SM7_ABCDEFGHI
SHORT       BITLEN_DELIVER_7CL0_GDC           192

#define DELIVER_8CL0_DEF                      /* 25 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_DLV,\
            PID_0, DCS_CL0_8BIT,\
            TIME_GMT_PLS_1HR,\
            SM8_ABCDEFGHI
SHORT       BITLEN_DELIVER_8CL0_DEF           200

#define DELIVER_8CL0_GDC                      /* 25 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_DLV,\
            PID_0, DCS_CL0_GDC2,\
            TIME_GMT,\
            SM8_ABCDEFGHI
SHORT       BITLEN_DELIVER_8CL0_GDC           200

#define DELIVER_16CL0_GDC                     /* 34 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_DLV,\
            PID_0, DCS_CL0_GDC4,\
            TIME_GMT,\
            SM16_ABCDEFGHI
SHORT       BITLEN_DELIVER_16CL0_GDC          272

#define DELIVER_7CL0L                         /* 159 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_DLV_LONG,\
            SMS_PID_SM_TYPE_0, DCS_CL0_GDC0,\
            TIME_GMT,\
            SM7_LONG
SHORT       BITLEN_DELIVER_7CL0L              1272
```

```
#define DELIVER_7CL1                        /* 24 bytes */\
        (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
        TP_ADDR_DLV,\
        SMS_PID_SM_TYPE_0, DCS_CL1_DEF,\
        TIME_GMT_PLS_1HR,\
        SM7_ABCDEFGHI
SHORT   BITLEN_DELIVER_7CL1                 192

#define DELIVER_7CL2                        /* 24 bytes */\
        (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
        TP_ADDR_DLV,\
        SMS_PID_SM_TYPE_0, DCS_CL2_GDC0,\
        TIME_GMT_PLS_1HR,\
        SM7_ABCDEFGHI
SHORT   BITLEN_DELIVER_7CL2                 192

#define DELIVER_7CL3                        /* 24 bytes */\
        (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
        TP_ADDR_DLV,\
        SMS_PID_SM_TYPE_0, DCS_CL3_GDC0,\
        TIME_GMT_PLS_1HR,\
        SM7_ABCDEFGHI
SHORT   BITLEN_DELIVER_7CL3                 192

#define DELIVER_7CL0_43                     /* 24 bytes */\
        (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
        TP_ADDR_DLV,\
        SMS_PID_REP_SM_TYPE_3, DCS_CL0_GDC0,\
        TIME_GMT_PLS_1HR,\
        SM7_ABCDEFGHI
SHORT   BITLEN_DELIVER_7CL0_43              192

#define DELIVER_7CL1_43                     /* 24 bytes */\
        (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
        TP_ADDR_DLV,\
        SMS_PID_REP_SM_TYPE_3, DCS_CL1_GDC0,\
        TIME_GMT_PLS_1HR,\
        SM7_ABCDEFGHI
SHORT   BITLEN_DELIVER_7CL1_43              192

#define DELIVER_7CL2_43                     /* 24 bytes */\
        (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
        TP_ADDR_DLV,\
        SMS_PID_REP_SM_TYPE_3, DCS_CL2_GDC0,\
        TIME_GMT_PLS_1HR,\
        SM7_ABCDEFGHI
SHORT   BITLEN_DELIVER_7CL2_43              192

#define DELIVER_7CL3_43                     /* 24 bytes */\
        (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
        TP_ADDR_DLV,\
        SMS_PID_REP_SM_TYPE_3, DCS_CL3_GDC0,\
        TIME_GMT_PLS_1HR,\
        SM7_ABCDEFGHI
SHORT   BITLEN_DELIVER_7CL3_43              192

#define DELIVER_7CL1_42                     /* 24 bytes */\
        (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
        TP_ADDR_DLV,\
        SMS_PID_REP_SM_TYPE_2, DCS_CL1_GDC0,\
        TIME_GMT_PLS_1HR,\
        SM7_RSTUVWXYZ
SHORT   BITLEN_DELIVER_7CL1_42              192
```

```
#define DELIVER_7CL1_43O                        /* 24 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_DLV1,\
            SMS_PID_REP_SM_TYPE_3, DCS_CL1_GDC0,\
            TIME_GMT_PLS_1HR,\
            SM7_RSTUVWXYZ
SHORT       BITLEN_DELIVER_7CL1_43O             192

#define SBM_DEF                                 /* 18 bytes */\
            SMS_SUBMIT,\
            NOT_PRESENT_8BIT,\
            TP_ADDR_SBM,\
            SMS_PID_DEFAULT,\
            DCS_DEF,\
            SM7_RSTUVWXYZ

#define MO_CHANGE                               /* 12 bytes */\
            SMS_SUBMIT,\
            NOT_PRESENT_8BIT,\
            TP_ADDR_SBM2,\
            SMS_PID_DEFAULT, DCS_DEF,\
            BYTE_00
SHORT       BITLEN_MO_CHANGE                    96

#define SBM_MO                                  /* 18 bytes */\
            SMS_SUBMIT,\
            TP_MR_3N1,                          /* as MO_INIT, but message is sent */\
            TP_ADDR_SBM,\
            SMS_PID_SM_TYPE_0, DCS_CL2_DEF,\
            SM7_ABCDEFGHI
SHORT       BITLEN_SBM_MO                       144

#define SBM_DA                                  /* 20 bytes */\
            SMS_SUBMIT,\
            TP_MR_3N1,\
            TP_ADDR_SBM2,\
            SMS_PID_SM_TYPE_0, DCS_CL2_DEF,\
            SM7_ABCDEFGHI
SHORT       BITLEN_SBM_DA                       160

#define SBM_7DEF                                /* 18 bytes */\
            SMS_SUBMIT,\
            TP_MR_3N1,\
            TP_ADDR_DLV,\
            SMS_PID_DEFAULT, DCS_DEF,\
            SM7_ABCDEFGHI
SHORT       BITLEN_SBM_7DEF                     144

#define SBM_7DEF_DA                             /* 20 bytes */\
            SMS_SUBMIT,\
            TP_MR_3N1,\
            TP_ADDR_SBM2,\
            SMS_PID_DEFAULT, DCS_DEF,\
            SM7_ABCDEFGHI
SHORT       BITLEN_SBM_7DEF_DA                  160

#define SBM_INIT                                /* 18 bytes */\
            SMS_SUBMIT,\
            TP_MR_3N1,\
            TP_ADDR_SBM1,\
            SMS_PID_SM_TYPE_0, DCS_CL2_DEF,\
            SM7_ABCDEFGHI
SHORT       BITLEN_SBM_INIT                     144
```

```
#define SBM_INIT_DA                              /* 20 bytes */\
            SMS_SUBMIT,\
            TP_MR_3N1,\
            TP_ADDR_SBM2,\
            SMS_PID_SM_TYPE_0, DCS_CL2_DEF,\
            SM7_ABCDEFGHI
SHORT       BITLEN_SBM_INIT_DA                   160

#define COMMAND_STAT_REQ                         /* 11 bytes */\
            SMS_COMMAND,\
            TP_MR_3N1,\
            SMS_PID_SM_TYPE_0,\
            SMS_CT_ENABLE,\
            MSG_REF_01,\
            TP_ADDR_SBM,\
            LEN_0

#define COMMAND_ENQ                              /* 11 bytes */\
            SMS_COMMAND,\
            TP_MR_3N1,\
            SMS_PID_SM_TYPE_0,\
            SMS_CT_ENQUIRY,\
            MSG_REF_01,\
            TP_ADDR_SBM,\
            LEN_0

#define COMMAND_CANCEL_REP                       /* 11 bytes */\
            SMS_COMMAND,\
            TP_MR_3N1,\
            SMS_PID_SM_TYPE_0,\
            SMS_CT_CANCEL_REP,\
            MSG_REF_01,\
            TP_ADDR_SBM,\
            LEN_0

#define COMMAND_DEL                              /* 11 bytes */\
            SMS_COMMAND,\
            TP_MR_3N2,\
            SMS_PID_SM_TYPE_0,\
            SMS_CT_DELETE,\
            MSG_REF_01,\
            TP_ADDR_SBM,\
            LEN_0

#define STATUS_REP                               /* 23 bytes */\
            (SMS_STATUS_REPORT | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_MR_3N2,\
            TP_ADDR_SBM,\
            TIME_GMT_PLS_1HR,\
            TIME_GMT,\
            SMS_ST_SM_REC_BY_SME,\
            BYTE_00                              /* Parameter Indicator not set */

#define DLVR_REP_ACK                             /* 6 bytes */\
            SMS_DELIVER_REPORT,\
            0x06,                                /* TP-DCS and TP-UD present */\
            DCS_CL0_8BIT,\
            0x02,'O','K'
SHORT BITLEN_DLVR_REP_ACK                        48
```

```
#define DLVR_REP_ERR                        /* 3 bytes */\
            SMS_DELIVER_REPORT,\
            SMS_FCS_ERROR_IN_MS,\
            BYTE_00                         /* Parameter Indicator not set */
SHORT BITLEN_DLVR_REP_ERR                   24

#define DELIVER_7CL2_SAT1                   /* 24 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_DLV,\
            SMS_PID_SIM_DOWNLOAD, DCS_CL2_DEF,\
            TIME_GMT_PLS_1HR,\
            SM7_ABCDEFGHI
SHORT       BITLEN_DELIVER_7CL2_SAT1        192

#define DELIVER_7CL2_SAT2                   /* 156 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_DLV,\
            SMS_PID_SIM_DOWNLOAD, DCS_CL2_GDC0,\
            TIME_GMT_PLS_1HR,\
            SM7_LONG
SHORT       BITLEN_DELIVER_7CL2_SAT2        1248

#define DELIVER_7CL1_SAT3                   /* 24 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_DLV,\
            SMS_PID_SIM_DOWNLOAD, DCS_CL1_GDC0,\
            TIME_GMT_PLS_1HR,\
            SM7_RSTUVWXYZ
SHORT       BITLEN_DELIVER_7CL1_SAT3        192

#define DELIVER_8CL2_SAT1                   /* 25 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_DLV,\
            SMS_PID_SIM_DOWNLOAD, DCS_CL2_GDC2,\
            TIME_GMT_PLS_1HR,\
            SM8_ABCDEFGHI
SHORT       BITLEN_DELIVER_8CL2_SAT1        200

#define DELIVER_8CL2_SAT2                   /* 25 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_DLV,\
            SMS_PID_SIM_DOWNLOAD, DCS_CL2_8BIT,\
            TIME_GMT_PLS_1HR,\
            SM8_ABCDEFGHI
SHORT       BITLEN_DELIVER_8CL2_SAT2        200

#define DELIVER_121_A                       /* 26 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_121_SPEC,\
            SMS_PID_DEFAULT, DCS_DEF,\
            TIME_GMT_PLS_1HR,\
            SM7_RSTUVWXYZ
SHORT       BITLEN_DELIVER_121_A            208

#define DELIVER_121_B                       /* 26 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_121_SPEC,\
            SMS_PID_SM_TYPE_0, DCS_CL2_DEF,\
            TIME_GMT_PLS_1HR,\
            SM7_ABCDEFGHI
SHORT       BITLEN_DELIVER_121_B            208
```

```
#define DELIVER_121_C                   /* 158 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_121_SPEC,\
            SMS_PID_SIM_DOWNLOAD, DCS_CL2_DEF,\
            TIME_GMT_PLS_1HR,\
            SM7_LONG
SHORT       BITLEN_DELIVER_121_C        1264

#define DELIVER_EMPTY                   /* 13 bytes */\
            (SMS_DELIVER | (SMS_MMS_NO_MORE_MESSAGES << 2)),\
            TP_ADDR_EMPTY,\
            SMS_PID_DEFAULT, DCS_CL2_DEF,\
            TIME_GMT_PLS_1HR,\
            0
SHORT       BITLEN_DELIVER_EMPTY        104

/* Fields for SMS_SDU */

FIELD (SMS_SDU_EMPTY_BUF)
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SMS_SDU_EMPTY_BUF, 175)

FIELD (SMS_SDU_MO_BUF)
            RP_ADDR_12345,
            MO_INIT,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF
ENDFIELD (SMS_SDU_MO_BUF, 175)

FIELD (SMS_SDU_MO_ABS_BUF)
            RP_ADDR_12345,
            MO_ABS_INIT,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
ENDFIELD (SMS_SDU_MO_ABS_BUF, 175)
```

```
FIELD (SMS_SDU_MT_BUF)
        RP_ADDR_12345,
        MT_INIT,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
ENDFIELD (SMS_SDU_MT_BUF, 175)

FIELD (SMS_SDU_MT_7CL1_BUF)
        RP_ADDR_12345,
        DELIVER_7CL1,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
ENDFIELD (SMS_SDU_MT_7CL1_BUF, 175)

FIELD (SMS_SDU_SUBMIT_ABS_BUF)
        RP_ADDR_12345,
        SUBMIT_ABS,
        TO_BE_IGNORED
ENDFIELD (SMS_SDU_SUBMIT_ABS_BUF, 35)

FIELD (SMS_SDU_SBM_DEF_BUF)
        RP_ADDR_12345,
        SBM_DEF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF
ENDFIELD (SMS_SDU_SBM_DEF_BUF, 175)

/* SMS Commands */

FIELD (SMS_SDU_COMMAND_STAT_REQ_BUF)
        RP_ADDR_12345,
        COMMAND_STAT_REQ,
        TO_BE_IGNORED,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00
ENDFIELD (SMS_SDU_COMMAND_STAT_REQ_BUF, 175)

FIELD (SMS_SDU_COMMAND_ENQ_BUF)
        RP_ADDR_12345,
        COMMAND_ENQ,
        TO_BE_IGNORED,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00
ENDFIELD (SMS_SDU_COMMAND_ENQ_BUF, 175)

FIELD (SMS_SDU_COMMAND_CANCEL_REP_BUF)
        RP_ADDR_12345,
        COMMAND_CANCEL_REP,
        TO_BE_IGNORED,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00
ENDFIELD (SMS_SDU_COMMAND_CANCEL_REP_BUF, 175)

FIELD (SMS_SDU_COMMAND_DEL_BUF)
        RP_ADDR_12345,
        COMMAND_DEL,
        TO_BE_IGNORED,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00
ENDFIELD (SMS_SDU_COMMAND_DEL_BUF, 175)

FIELD (SMS_SDU_STATUS_REP_BUF)
        RP_ADDR_12345,
        STATUS_REP,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SMS_SDU_STATUS_REP_BUF, 175)

FIELD (SMS_SDU_DLVR_REP_ACK_BUF)
        BYTE_00,                        /* no SCA */
        DLVR_REP_ACK
ENDFIELD (SMS_SDU_DLVR_REP_ACK_BUF, 7)

FIELD (SMS_SDU_DLVR_REP_ERR_BUF)
        BYTE_00,                        /* no SCA */
        DLVR_REP_ERR
ENDFIELD (SMS_SDU_DLVR_REP_ERR_BUF, 4)

/* Decoded RP address values */
FIELD (BCD_0811112222) 0x00, 0x08, 0x01, 0x01, 0x01, 0x01, 0x02, 0x02, 0x02, 0x02
ENDFIELD (BCD_0811112222, 10)

FIELD (BCD_0123987654) 0x00, 0x01, 0x02, 0x03, 0x09, 0x08, 0x07, 0x06, 0x05, 0x04
ENDFIELD (BCD_0123987654, 10)

FIELD (BCD_123456) 0x01, 0x02, 0x03, 0x04, 0x05, 0x06
ENDFIELD (BCD_123456, 6)

FIELD (BCD_23456) 0x02, 0x03, 0x04, 0x05, 0x06
ENDFIELD (BCD_23456, 5)

FIELD (BCD_12345) 0x01, 0x02, 0x03, 0x04, 0x05
ENDFIELD (BCD_12345, 5)

FIELD (BCD_654321) 0x06, 0x05, 0x04, 0x03, 0x02, 0x01
ENDFIELD (BCD_654321, 6)

FIELD (BCD_987654) 0x09, 0x08, 0x07, 0x06, 0x05, 0x04
ENDFIELD (BCD_987654, 6)

FIELD (BCD_98765) 0x09, 0x08, 0x07, 0x06, 0x05
ENDFIELD (BCD_98765, 5)

FIELD (MSG7_ABCDEFGHI)
        0x0E, 0xE1, 0x90, 0x58, 0x34, 0x1E, 0x91, 0x49
ENDFIELD (MSG7_ABCDEFGHI, 8)

FIELD (SM7_ABCDEFGHI_CNT)
        0x0E, 0xE1, 0x90, 0x58, 0x34, 0x1E, 0x91, 0x49,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SM7_ABCDEFGHI_CNT, 140)

FIELD (SM8_ABCDEFGHI_CNT)
        0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48, 0x49,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SM8_ABCDEFGHI_CNT, 140)

FIELD (MSG8_ABCDEFGHI) 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48, 0x49
ENDFIELD (MSG8_ABCDEFGHI, 9)

FIELD (MSG16_ABCDEFGHI)
        0x01, 0x00, 0x00, 0x42, 0x00, 0x43, 0x00, 0x44, 0x00, 0x45,
        0x00, 0x46, 0x00, 0x47, 0x00, 0x48, 0x00, 0x49
ENDFIELD (MSG16_ABCDEFGHI, 18)

FIELD (SM16_ABCDEFGHI_CNT)
        0x01, 0x00, 0x00, 0x42, 0x00, 0x43, 0x00, 0x44, 0x00, 0x45,
        0x00, 0x46, 0x00, 0x47, 0x00, 0x48, 0x00, 0x49,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00
ENDFIELD (SM16_ABCDEFGHI_CNT, 140)

FIELD (UNKN_SMS_MO_MSG)
        0x10, 0x00, 0x18, 0x00, 0x00, 0x00, 0x00,
        0x89, 0xFE, 0xFD, 0x00, 0x00
ENDFIELD (UNKN_SMS_MO_MSG, 12)

FIELD (UNKN_SMS_MT_MSG)
        0x10, 0x00, 0x18, 0x00, 0x00, 0x00, 0x00,
        0x09, 0xFE, 0xFD, 0x00, 0x00
ENDFIELD (UNKN_SMS_MT_MSG, 12)

BYTE SIMREC_SMSS_MSG_REF_LEN 0x01

FIELD (SIMREC_SMSS_MCEF)
        0xFE,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIMREC_SMSS_MCEF, 256)
```

```
BYTE SIMREC_SMSS_MEM_FLAG_LEN 0x01

FIELD (SIMREC_SMSS_MEM_FLAG_AVAIL) 0xFF,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIMREC_SMSS_MEM_FLAG_AVAIL, 256)

FIELD (SIM_NO_DATA)
        TO_BE_IGNORED
ENDFIELD (SIM_NO_DATA, 5)

FIELD (SIM_SMS_EMPTY) 0x00,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_EMPTY, 256)

FIELD (SIM_SMS_MO)
        SMS_RECORD_STO_UNSENT,
        RP_ADDR_12345,
        MO_INIT,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_MO, 256)

FIELD (SIM_SMS_SBM_MO)
        SMS_RECORD_STO_SENT,
        RP_ADDR_12345,                  /* TP-SC: 12345*/
        SBM_MO,                         /* TP-UDL, TP-UD*/
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_SBM_MO, 256)

FIELD (SIM_SMS_SBM_DA)
        SMS_RECORD_STO_SENT,
        RP_ADDR_12345,
        SBM_DA,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_SBM_DA, 256)

FIELD (SIM_SMS_SBM_SCA)
        SMS_RECORD_STO_SENT,
        RP_ADDR_0811112222,
        SBM_MO,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_SBM_SCA, 256)

FIELD (SIM_SMS_SBM_DA_SCA)
        SMS_RECORD_STO_SENT,
        RP_ADDR_0811112222,
        SBM_DA,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_SBM_DA_SCA, 256)

FIELD (SIM_SMS_SBM_7DEF)
        SMS_RECORD_STO_SENT,
        RP_ADDR_12345,                      /* TP-SC: 12345*/
        SBM_INIT,                           /* TP-UDL, TP-UD*/
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_SBM_7DEF, 256)

FIELD (SIM_SMS_SBM_7DEF_DA)
        SMS_RECORD_STO_SENT,
        RP_ADDR_12345,
        SBM_INIT_DA,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_SBM_7DEF_DA, 256)

FIELD (SIM_SMS_SBM_7DEF_SCA)
        SMS_RECORD_STO_SENT,
        RP_ADDR_0811112222,
        SBM_INIT,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_SBM_7DEF_SCA, 256)

FIELD (SIM_SMS_SBM_7DEF_DA_SCA)
        SMS_RECORD_STO_SENT,
        RP_ADDR_0811112222,
        SBM_INIT_DA,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_SBM_7DEF_DA_SCA, 256)
```

```
FIELD (SIM7_SMS_MO_PID5F_SENT)
 0x05,
 0x04, 0x81, 0x21, 0x43, 0xF5,                /* TP-SC: 12345*/
 0x1D,                                        /* TP-MTI, TP-RD ...*/
 0x04,                                        /* TP-MR*/
 0x06, 0x81, 0x56, 0x34, 0x12,                /* TP-DA: 654321*/
 0x5F,                                        /* TP-PID*/
 0x00,                                        /* TP-DCS*/
 0x89, 0x10, 0x70, 0x21, 0x43, 0x65, 0x40,    /* TP-VP*/
 0x09, 0x0E, 0xE1, 0x90, 0x58, 0x34, 0x1E, 0x91, 0x49, /* TP-UDL, TP-UD*/
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM7_SMS_MO_PID5F_SENT, 256)

FIELD (SIM_SMS_MO_READ)
 MAX_SIM_CMD,
 0x05,
 0x04, 0x81, 0x21, 0x43, 0xF5,
 0x01,
 MSG_REF_AB,
 0x05, 0x81, 0x89, 0x67, 0xF5,
 0x40,
 0xF2,
 0x09, 0x41, 0xE1, 0x90, 0x58, 0x34, 0x1E, 0x91, 0x49,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF
ENDFIELD (SIM_SMS_MO_READ, 177)

FIELD (SIM_SMS_MT_READ)
              SMS_RECORD_REC_READ,
              RP_ADDR_12345,
              MT_INIT,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_MT_READ, 256)

FIELD (SIM_SMS_COMMAND_SENT)
              0x05,                               /* store sent*/
              0x04, 0x81, 0x21, 0x43, 0xF5,       /* SCA*/
              0x22,                               /* SMS-COMMAND, TP-SRR*/
              MSG_REF_01,                         /* TP-MR*/
              0x00,                               /* TP-PID*/
              0x00,                               /* TP-CT = Enquiry*/
              MSG_REF_AB,                         /* TP-MN*/
              0x05, 0x81, 0x89, 0x67, 0xF5,       /* TP-DA*/
              0x00,                               /* TP-CDL = 0: no TP-CD*/
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_COMMAND_SENT, 256)

FIELD (SIM_SMS_MT)
              SMS_RECORD_REC_UNREAD,
              RP_ADDR_12345,
              MT_INIT,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_MT, 256)

FIELD (SIM_SMS_ST_REPORT)
              0x03,                                   /* received unread*/
              0x04, 0x81, 0x21, 0x43, 0xF5,           /* SCA*/
              0x02,                                   /* SMS-STATUS-REPORT*/
              MSG_REF_AB,                             /* TP-MR*/
              0x05, 0x81, 0x89, 0x67, 0xF5,           /* TP-RA*/
              0x89, 0x80, 0x13, 0x10, 0x25, 0x31, 0x01, /* TP-SCTS*/
              0x89, 0x80, 0x13, 0x00, 0x43, 0x51, 0x01, /* TP-DT*/
              0x21,                                   /* TP-ST*/
              0x00,                                   /* TP-PI = 0: no further parameters*/
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_ST_REPORT, 176)

BYTE  SIM_SMSS_MEM_CAP_ONLY_LEN  0x01

FIELD (SIM_SMSS_NO_MEM_CAP)
              0xFE,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMSS_NO_MEM_CAP, 256)

FIELD (SIM_SMS_MO_ABS)
              SMS_RECORD_STO_SENT,
              RP_ADDR_12345,
              MO_ABS_INIT,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_MO_ABS, 256)

FIELD (SIM_SMS_CLASS_0_43)
              SMS_RECORD_REC_UNREAD,
              RP_ADDR_12345,
              DELIVER_7CL0_43,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_CLASS_0_43, MAX_SIM_CMD)

FIELD (SIM_SMS_CLASS_0L)
              SMS_RECORD_REC_UNREAD,
              RP_ADDR_0811112222,
              DELIVER_7CL0L,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_CLASS_0L, MAX_SIM_CMD)

FIELD (SIM_SMS_CLASS_1)
              SMS_RECORD_REC_UNREAD,
              RP_ADDR_12345,
              DELIVER_7CL1,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_CLASS_1, 256)

FIELD (SIM_SMS_CLASS_1_42)
              SMS_RECORD_REC_UNREAD,
              RP_ADDR_12345,
              DELIVER_7CL1_42,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_CLASS_1_42, 256)

FIELD (SIM_SMS_CLASS_1_43)
              SMS_RECORD_REC_UNREAD,
              RP_ADDR_12345,
              DELIVER_7CL1_43,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_CLASS_1_43, 256)

FIELD (SIM_SMS_CLASS_1_43S)
              SMS_RECORD_REC_UNREAD,
              RP_ADDR_23456,
              DELIVER_7CL1_43,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_CLASS_1_43S, 256)

FIELD (SIM_SMS_CLASS_1_43O)
              SMS_RECORD_REC_UNREAD,
              RP_ADDR_12345,
              DELIVER_7CL1_43O,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_CLASS_1_43O, 256)

FIELD (SIM_SMS_CLASS_2)
              SMS_RECORD_REC_UNREAD,
              RP_ADDR_12345,
              DELIVER_7CL2,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_CLASS_2, MAX_SIM_CMD)

FIELD (SIM_SMS_CLASS_2_SAT)
              SMS_RECORD_REC_UNREAD,
              RP_ADDR_12345,
              DELIVER_7CL2_SAT1,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_CLASS_2_SAT, 256)

FIELD (SIM_SMS_CLASS_2_INV_DCS)
              SMS_RECORD_REC_UNREAD,
              RP_ADDR_12345,
              DELIVER_7CL1_SAT3,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_CLASS_2_INV_DCS, 256)

FIELD (SIM_SMS_CLASS_2_43)
              SMS_RECORD_REC_UNREAD,
              RP_ADDR_12345,
              DELIVER_7CL2_43,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
              0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_CLASS_2_43, 256)
```

```
FIELD (SIM_SMS_CLASS_3)
            SMS_RECORD_REC_UNREAD,
            RP_ADDR_12345,
            DELIVER_7CL3,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_CLASS_3, MAX_SIM_CMD)

FIELD (SIM_SMS_CLASS_3_43)
            SMS_RECORD_REC_UNREAD,
            RP_ADDR_12345,
            DELIVER_7CL3_43,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_CLASS_3_43, 256)

FIELD (SIM_SMS_SBM_DEF)
            SMS_RECORD_STO_UNSENT,
            RP_ADDR_12345,
            SBM_DEF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_SBM_DEF, 256)

/* One2One operator-specific messages (TP-DELIVER) */

FIELD (SIM_SMS_DELIVER_121_A)
            SMS_RECORD_REC_UNREAD,
            RP_ADDR_12345,
            DELIVER_121_A,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_DELIVER_121_A, 256)

FIELD (SIM_SMS_DELIVER_121_B)
            SMS_RECORD_REC_UNREAD,
            RP_ADDR_12345,
            DELIVER_121_B,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_DELIVER_121_B, 256)

FIELD (SIM_SMS_DELIVER_121_C)
            SMS_RECORD_REC_UNREAD,
            RP_ADDR_12345,
            DELIVER_121_C,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
            0x00, 0x00, 0x00, 0x00
ENDFIELD (SIM_SMS_DELIVER_121_C, 256)

/* SMS MT record with variable DCS*/

FIELD (SIM_SMS_MT_DEF_CNF)
            0x01,
            0x04, 0x81, 0x21, 0x43, 0xF5,
            0x04,
            0x05, 0x81, 0x89, 0x67, 0xF5,
            PID_0,
            DCS_DEF,
            0x89, 0x80, 0x13, 0x10, 0x25, 0x31, 0x01,
            9, 0x0E, 0xE1, 0x90, 0x58, 0x34, 0x1E, 0x91, 0x49,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
ENDFIELD (SIM_SMS_MT_DEF_CNF, 176)

FIELD (SIM_SMS_MT_CL0_DEF_CNF)
            0x01,
            0x04, 0x81, 0x21, 0x43, 0xF5,
            0x04,
            0x05, 0x81, 0x89, 0x67, 0xF5,
            PID_0,
            DCS_CL0_DEF,
            0x89, 0x80, 0x13, 0x10, 0x25, 0x31, 0x01,
            9, 0x0E, 0xE1, 0x90, 0x58, 0x34, 0x1E, 0x91, 0x49,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
ENDFIELD (SIM_SMS_MT_CL0_DEF_CNF, 176)

FIELD (SIM_SMS_MT_CL0_GDC0_CNF)
            0x01,
            0x04, 0x81, 0x21, 0x43, 0xF5,
            0x04,
            0x05, 0x81, 0x89, 0x67, 0xF5,
            PID_0,
            DCS_CL0_GDC0,
            0x89, 0x80, 0x13, 0x10, 0x25, 0x31, 0x01,
            9, 0x0E, 0xE1, 0x90, 0x58, 0x34, 0x1E, 0x91, 0x49,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
ENDFIELD (SIM_SMS_MT_CL0_GDC0_CNF, 176)

FIELD (SIM_SMS_MT_CL0_8BIT_CNF)
            0x01,
            0x04, 0x81, 0x21, 0x43, 0xF5,
            0x04,
            0x05, 0x81, 0x89, 0x67, 0xF5,
            PID_0,
            DCS_CL0_8BIT,
            0x89, 0x80, 0x13, 0x10, 0x25, 0x31, 0x01,
            9,
            0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48, 0x49,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
ENDFIELD (SIM_SMS_MT_CL0_8BIT_CNF, 176)

FIELD (SIM_SMS_MT_CL0_GDC2_CNF)
            0x01,
            0x04, 0x81, 0x21, 0x43, 0xF5,
            0x04,
            0x05, 0x81, 0x89, 0x67, 0xF5,
            PID_0,
            DCS_CL0_GDC2,
            0x89, 0x80, 0x13, 0x10, 0x25, 0x31, 0x01,
            9,
            0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48, 0x49,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
ENDFIELD (SIM_SMS_MT_CL0_GDC2_CNF, 176)

FIELD (SIM_SMS_MT_CL0_GDC4_CNF)
            0x01,
            0x04, 0x81, 0x21, 0x43, 0xF5,
            0x04,
            0x05, 0x81, 0x89, 0x67, 0xF5,
            PID_0,
            DCS_CL0_GDC4,
            0x89, 0x80, 0x13, 0x10, 0x25, 0x31, 0x01,
            18,
            0x01, 0x00, 0x00, 0x42, 0x00, 0x43, 0x00, 0x44, 0x00, 0x45,
            0x00, 0x46, 0x00, 0x47, 0x00, 0x48, 0x00, 0x49,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF
ENDFIELD (SIM_SMS_MT_CL0_GDC4_CNF, 176)

FIELD (SIM_SMS_MT_MWI_DISCD_CNF)
            0x01,
            0x04, 0x81, 0x21, 0x43, 0xF5,
            0x04,
            0x05, 0x81, 0x89, 0x67, 0xF5,
            PID_0,
            DCS_MWI_DISCD,
            0x89, 0x80, 0x13, 0x10, 0x25, 0x31, 0x01,
            9, 0x0E, 0xE1, 0x90, 0x58, 0x34, 0x1E, 0x91, 0x49,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
ENDFIELD (SIM_SMS_MT_MWI_DISCD_CNF, 176)

FIELD (SIM_SMS_MT_MWI_STR_DEF_CNF)
            0x01,
            0x04, 0x81, 0x21, 0x43, 0xF5,
            0x04,
            0x05, 0x81, 0x89, 0x67, 0xF5,
            PID_0,
            DCS_MWI_STR_DEF,
            0x89, 0x80, 0x13, 0x10, 0x25, 0x31, 0x01,
            9, 0x0E, 0xE1, 0x90, 0x58, 0x34, 0x1E, 0x91, 0x49,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF
ENDFIELD (SIM_SMS_MT_MWI_STR_DEF_CNF, 176)

FIELD (SIM_SMS_MT_MWI_STR_UCS2_CNF)
            0x01,
            0x04, 0x81, 0x21, 0x43, 0xF5,
            0x04,
            0x05, 0x81, 0x89, 0x67, 0xF5,
            PID_0,
            DCS_MWI_STR_UCS2,
            0x89, 0x80, 0x13, 0x10, 0x25, 0x31, 0x01,
            18,
            0x01, 0x00, 0x00, 0x42, 0x00, 0x43, 0x00, 0x44, 0x00, 0x45,
            0x00, 0x46, 0x00, 0x47, 0x00, 0x48, 0x00, 0x49,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
            0xFF
ENDFIELD (SIM_SMS_MT_MWI_STR_UCS2_CNF, 176)

/* CP-DATA RP-ERROR SAT BUSY */

FIELD (CP_DATA_RP_ERROR_SAT_BUSY)
            0x70, 0x00,
            0x18, 0x00,
            0x00, 0x00, 0x00,
            0x89, 0x01,                  /* CP-DATA */
            11,                          /* length RPDU */
            0x04,                        /* RP-ERROR */
            0x01,                        /* message reference */
            0x01,                        /* length rp-cause */
            0x6F,                        /* protocol error unspecified */
            0x41,                        /* IEI tpdu */
            5,                           /* length tpdu */
            0x00,                        /* message type deliver report */
            0xd4,                        /* TP-FCS SAT busy */
            0x03,                        /* TP-PI : PID and DCS will follow */
            SMS_PID_SIM_DOWNLOAD,        /* TP-PID */
            DCS_CL2_GDC0                 /* TP-DCS */
ENDFIELD (CP_DATA_RP_ERROR_SAT_BUSY, 21)

FIELD (CP_DATA_RP_ACK_TPDU1)
            0xA0, 0x00,
            0x18, 0x00,
            0x00, 0x00, 0x00,
            0x89, 0x01,                  /* CP-DATA */
            17,                          /* length RPDU */
            RP_ACK_UL,                   /* RP-ACK */
            MSG_REF_01,                  /* message reference */
            0x41,                        /* IEI tpdu */
            13,                          /* length tpdu */
            SMS_DELIVER_REPORT,
            0x07,                        /* TP-PID */
            SMS_PID_SIM_DOWNLOAD,        /* TP-PI */
            DCS_CL2_DEF,                 /* TP-DCS */
            SM7_RSTUVWXYZ                /* TP-UD */
ENDFIELD (CP_DATA_RP_ACK_TPDU1, 27)

FIELD (CP_DATA_RP_ACK_TPDU3)
            0xA8, 0x00,
            0x18, 0x00,
            0x00, 0x00, 0x00,
            0x89, 0x01,                  /* CP-DATA */
            18,                          /* length RPDU */
            RP_ACK_UL,                   /* RP-ACK */
            MSG_REF_01,                  /* message reference */
            0x41,                        /* IEI tpdu */
            14,                          /* length tpdu */
            SMS_DELIVER_REPORT,
            0x07,                           /* TP-PID */
            SMS_PID_SIM_DOWNLOAD,           /* TP-PI */
            DCS_CL2_GDC2,                   /* TP-DCS */
            SM8_ABCDEFGHI                   /* TP-UD */
ENDFIELD (CP_DATA_RP_ACK_TPDU3, 28)

FIELD (CP_DATA_RP_ERROR_TPDU2)
            0xB8, 0x00,
            0x18, 0x00,
            0x00, 0x00, 0x00,
            0x89, 0x01,                     /* CP-DATA */
            20,                             /* length RPDU */
            RP_ERROR_UL,                    /* RP-ERROR */
            MSG_REF_01,                     /* message reference */
            1,                              /* length rp-cause */
            SMS_RP_CS_PROTOCOL_ERROR,       /* protocol error unspecified */
            0x41,                           /* IEI tpdu */
            14,                             /* length tpdu */
            SMS_DELIVER_REPORT,
            SMS_FCS_SAT_DNL_ERROR,          /* TP-FCS */
            0x07,                           /* TP-PID */
            SMS_PID_SIM_DOWNLOAD,           /* TP-PI */
            DCS_CL2_GDC0,                   /* TP-DCS */
            SM7_RSTUVWXYZ                   /* TP-DU */
ENDFIELD (CP_DATA_RP_ERROR_TPDU2, 30)

FIELD (CP_DATA_RP_ERROR_TPDU4)
            0xC0, 0x00,
            0x18, 0x00,
            0x00, 0x00, 0x00,
            0x89, 0x01,                     /* CP-DATA */
            21,                             /* length RPDU */
            RP_ERROR_UL,                    /* RP-ERROR */
            MSG_REF_01,                     /* message reference */
            1,                              /* length rp-cause */
            SMS_RP_CS_PROTOCOL_ERROR,       /* protocol error unspecified */
            0x41,                           /* IEI tpdu */
            15,                             /* length tpdu */
            SMS_DELIVER_REPORT,
            SMS_FCS_SAT_DNL_ERROR,          /* TP-FCS */
            0x07,                           /* TP-PID */
            SMS_PID_SIM_DOWNLOAD,           /* TP-PI */
            DCS_CL2_8BIT,                   /* TP-DCS */
            SM8_ABCDEFGHI                   /* TP-UD */
ENDFIELD (CP_DATA_RP_ERROR_TPDU4, 31)
```

/*--- End of declaration part, begin of executable part of definitions ---*/

```
/* Store last used TP-MR in EF(SMSS) */
BEGINARRAY_PART (SIMREC_SMSS_MSG_REF, 1)
                TP_MR_3N1
ENDARRAY

BEGINARRAY_PART (SIMREC_SMSS_MSG_REF_N2, 1)
                TP_MR_3N2
ENDARRAY

BEGINARRAY (SIM_SMS_MT_DELIVER_7DEF, MAX_SIM_CMD)
                SMS_RECORD_REC_UNREAD,
                RP_ADDR_12345,
                DELIVER_7DEF,
                0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00
ENDARRAY

/* Empty SMS_SDU */

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_EMPTY)
        SET_COMP ("l_buf", 0x0000)
        SET_COMP ("o_buf", 0x0000)
        SET_COMP ("buf", SMS_SDU_EMPTY_BUF)
ENDSTRUCT

/* MO SMS-SDU */

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_MO)
        SET_COMP ("l_buf", 184)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_MO_BUF)
ENDSTRUCT

/* MO SMS-SDU (VP-ABS) */

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_MO_ABS)
        SET_COMP ("l_buf", 240)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_MO_ABS_BUF)
ENDSTRUCT

/* MT SMS-SDU */

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_MT)
        SET_COMP ("l_buf", 232)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_MT_BUF)
ENDSTRUCT

/* SUBMIT SMS-SDU (VP-ABS) */

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_SUBMIT_ABS)
        SET_COMP ("l_buf", 240)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_SUBMIT_ABS_BUF)
ENDSTRUCT

/* MT SMS-SDU */

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL0_BUF, 29)
        RP_ADDR_12345,
        DELIVER_7CL0
ENDARRAY
```

```
BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL0)
            SET_COMP ("l_buf", 232)
            SET_COMP ("o_buf", 0)
            SET_COMP ("buf", SMS_SDU_DELIVER_7CL0_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL0_DEF_BUF, 29)
            RP_ADDR_12345,
            DELIVER_7CL0_DEF
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL0_DEF)
            SET_COMP ("l_buf", 232)
            SET_COMP ("o_buf", 0)
            SET_COMP ("buf", SMS_SDU_DELIVER_7CL0_DEF_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL0_GDC_BUF, 29)
            RP_ADDR_12345,
            DELIVER_7CL0_GDC
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL0_GDC)
            SET_COMP ("l_buf", 232)
            SET_COMP ("o_buf", 0)
            SET_COMP ("buf", SMS_SDU_DELIVER_7CL0_GDC_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_8CL0_DEF_BUF, 30)
            RP_ADDR_12345,
            DELIVER_8CL0_DEF
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_8CL0_DEF)
            SET_COMP ("l_buf", 240)
            SET_COMP ("o_buf", 0)
            SET_COMP ("buf", SMS_SDU_DELIVER_8CL0_DEF_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_8CL0_GDC_BUF, 30)
            RP_ADDR_12345,
            DELIVER_8CL0_GDC
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_8CL0_GDC)
            SET_COMP ("l_buf", 240)
            SET_COMP ("o_buf", 0)
            SET_COMP ("buf", SMS_SDU_DELIVER_8CL0_GDC_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_16CL0_GDC_BUF, 39)
            RP_ADDR_12345,
            DELIVER_16CL0_GDC
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_16CL0_GDC)
            SET_COMP ("l_buf", 312)
            SET_COMP ("o_buf", 0)
            SET_COMP ("buf", SMS_SDU_DELIVER_16CL0_GDC_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL0L_BUF, 166)
            RP_ADDR_0811112222,
            DELIVER_7CL0L
ENDARRAY
```

```
BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL0L)
                SET_COMP ("l_buf", 1328)
                SET_COMP ("o_buf", 0)
                SET_COMP ("buf", SMS_SDU_DELIVER_7CL0L_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL1_BUF, 29)
                RP_ADDR_12345,
                DELIVER_7CL1
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL1)
                SET_COMP ("l_buf", 232)
                SET_COMP ("o_buf", 0)
                SET_COMP ("buf", SMS_SDU_DELIVER_7CL1_BUF)
ENDSTRUCT

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_MT_7CL1)
                SET_COMP ("l_buf", 232)
                SET_COMP ("o_buf", 0)
                SET_COMP ("buf", SMS_SDU_MT_7CL1_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL2_BUF, 29)
                RP_ADDR_12345,
                DELIVER_7CL2
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL2)
                SET_COMP ("l_buf", 232)
                SET_COMP ("o_buf", 0)
                SET_COMP ("buf", SMS_SDU_DELIVER_7CL2_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL3_BUF, 29)
                RP_ADDR_12345,
                DELIVER_7CL3
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL3)
                SET_COMP ("l_buf", 232)
                SET_COMP ("o_buf", 0)
                SET_COMP ("buf", SMS_SDU_DELIVER_7CL3_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7DEF_BUF, 29)
                RP_ADDR_12345,
                DELIVER_7DEF
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7DEF)
                SET_COMP ("l_buf", 232)
                SET_COMP ("o_buf", 0)
                SET_COMP ("buf", SMS_SDU_DELIVER_7DEF_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL0_43_BUF, 29)
                RP_ADDR_12345,
                DELIVER_7CL0_43
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL0_43)
                SET_COMP ("l_buf", 232)
                SET_COMP ("o_buf", 0)
                SET_COMP ("buf", SMS_SDU_DELIVER_7CL0_43_BUF)
ENDSTRUCT
```

```
BEGINARRAY_PART (SMS_SDU_DELIVER_7CL1_43_BUF, 29)
        RP_ADDR_12345,
        DELIVER_7CL1_43
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL1_43)
        SET_COMP ("l_buf", 232)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_DELIVER_7CL1_43_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL2_43_BUF, 29)
        RP_ADDR_12345,
        DELIVER_7CL2_43
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL2_43)
        SET_COMP ("l_buf", 232)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_DELIVER_7CL2_43_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL3_43_BUF, 29)
        RP_ADDR_12345,
        DELIVER_7CL3_43
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL3_43)
        SET_COMP ("l_buf", 232)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_DELIVER_7CL3_43_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL1_42_BUF, 29)
        RP_ADDR_12345,
        DELIVER_7CL1_42
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL1_42)
        SET_COMP ("l_buf", 232)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_DELIVER_7CL1_42_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL1_43S_BUF, 29)
        RP_ADDR_23456,
        DELIVER_7CL1_43
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL1_43S)
        SET_COMP ("l_buf", 232)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_DELIVER_7CL1_43S_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL1_43O_BUF, 29)
        RP_ADDR_12345,
        DELIVER_7CL1_43O
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL1_43O)
        SET_COMP ("l_buf", 232)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_DELIVER_7CL1_43O_BUF)
ENDSTRUCT
```

```
BEGINARRAY_PART (SMS_SDU_DELIVER_121_A_BUF, 31)
        RP_ADDR_12345,
        DELIVER_121_A
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_121_A)
        SET_COMP ("l_buf", 248)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_DELIVER_121_A_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_121_B_BUF, 31)
        RP_ADDR_12345,
        DELIVER_121_B
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_121_B)
        SET_COMP ("l_buf", 248)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_DELIVER_121_B_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_121_C_BUF, 163)
        RP_ADDR_12345,
        DELIVER_121_C
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_121_C)
        SET_COMP ("l_buf", 1304)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_DELIVER_121_C_BUF)
ENDSTRUCT

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_SBM_DEF)
        SET_COMP ("l_buf", 184)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_SBM_DEF_BUF)
ENDSTRUCT

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_SBM_DEF_X)
        SET_COMP ("l_buf", 1400)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_SBM_DEF_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_MO_CHANGE_BUF, 19)
        RP_ADDR_0811112222,
        MO_CHANGE
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_MO_CHANGE)
        SET_COMP ("l_buf", 152)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_MO_CHANGE_BUF)
ENDSTRUCT

/* SMS SDU Command */

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_COMMAND_STAT_REQ)
        SET_COMP ("l_buf", 128)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_COMMAND_STAT_REQ_BUF)
ENDSTRUCT
```

```
BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_COMMAND_ENQ)
        SET_COMP ("l_buf", 128)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_COMMAND_ENQ_BUF)
ENDSTRUCT

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_COMMAND_CANCEL_REP)
        SET_COMP ("l_buf", 128)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_COMMAND_CANCEL_REP_BUF)
ENDSTRUCT

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_COMMAND_DEL)
        SET_COMP ("l_buf", 128)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_COMMAND_DEL_BUF)
ENDSTRUCT

/* SMS SDU Status Indication */

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_STATUS_REP)
        SET_COMP ("l_buf", 224)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_STATUS_REP_BUF)
ENDSTRUCT

/* SMS SDU Deliver Report */

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DLVR_REP_ACK)
        SET_COMP ("l_buf", 56)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_DLVR_REP_ACK_BUF)
ENDSTRUCT

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DLVR_REP_ERR)
        SET_COMP ("l_buf", 32)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_DLVR_REP_ERR_BUF)
ENDSTRUCT

/* SMS SDU failed SAT message */

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL2_SAT1_BUF, 29)
        RP_ADDR_12345,
        DELIVER_7CL2_SAT1
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL2_SAT1)
        SET_COMP ("l_buf", 232)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_DELIVER_7CL2_SAT1_BUF)
ENDSTRUCT

BEGINARRAY_PART (SMS_SDU_DELIVER_7CL1_SAT3_BUF, 29)
        RP_ADDR_12345,
        DELIVER_7CL1_SAT3
ENDARRAY

BEGIN_PSTRUCT ("sms_sdu", SMS_SDU_DELIVER_7CL1_SAT3)
        SET_COMP ("l_buf", 232)
        SET_COMP ("o_buf", 0)
        SET_COMP ("buf", SMS_SDU_DELIVER_7CL1_SAT3_BUF)
ENDSTRUCT
```

```
/* rp smma uplink */

BEGIN_MSTRUCT ("cp_user_data_ul", RP_SMMA)
        SET_COMP ("rp_mti",             RP_SMMA_UL)
        SET_COMP ("reference",          TP_MR_3)
        SKIP_COMP ("rp_data_ul")
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

BEGIN_MSTRUCT ("cp_user_data_ul", RP_SMMA_REP)
        SET_COMP ("rp_mti",             RP_SMMA_UL)
        SET_COMP ("reference",          TP_MR_3_1)
        SKIP_COMP ("rp_data_ul")
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp ack downlink for RP-SMMA */

BEGIN_MSTRUCT ("cp_user_data_dl", RP_ACK_SMMA)
        SET_COMP ("rp_mti",             RP_ACK_DL)
        SET_COMP ("reference",          TP_MR_3)
        SKIP_COMP ("rp_data_dl")
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

BEGIN_MSTRUCT ("cp_user_data_dl", RP_ACK_SMMA_REP)
        SET_COMP ("rp_mti",             RP_ACK_DL)
        SET_COMP ("reference",          TP_MR_3_1)
        SKIP_COMP ("rp_data_dl")
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* SDU tp submit */
SET_BITBUF ("tpdu", TPDU_SUBMIT_ABS, 200)
        SUBMIT_ABS
ENDBITBUF

/* rp user data submit */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_SUBMIT_ABS)
        SET_COMP ("tp_mti",             SMS_SUBMIT)
        SET_COMP ("tpdu",               TPDU_SUBMIT_ABS)
ENDSTRUCT

/* rp service center address */
BEGIN_MSTRUCT ("rp_addr", RP_SCA_12345)
        SET_COMP ("ton",                SMS_TON_INTERNATIONAL)
        SET_COMP ("npi",                SMS_NPI_ISDN)
        SET_COMP ("num",                BCD_12345)
ENDSTRUCT

BEGIN_MSTRUCT ("rp_addr", RP_SCA_0811112222)
        SET_COMP ("ton",                SMS_TON_UNKNOWN)
        SET_COMP ("npi",                SMS_NPI_ISDN)
        SET_COMP ("num",                BCD_0811112222)
ENDSTRUCT

BEGIN_MSTRUCT ("rp_addr", RP_SCA_23456)
        SET_COMP ("ton",                SMS_TON_NATIONAL)
        SET_COMP ("npi",                SMS_NPI_ISDN)
        SET_COMP ("num",                BCD_23456)
ENDSTRUCT
```

```
/* rp submit */
BEGIN_MSTRUCT ("rp_data_ul", RP_SUBMIT_ABS)
                SET_COMP ("rp_addr",                RP_SCA_12345)
                SET_COMP ("rp_user_data",           RP_UD_SUBMIT_ABS)
ENDSTRUCT

/* rp data submit */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_SUBMIT_ABS)
                SET_COMP ("rp_mti",                 RP_DATA_UL)
                SET_COMP ("reference",              TP_MR_3N1)
                SET_COMP ("rp_data_ul",             RP_SUBMIT_ABS)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp ack downlink */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_ACK_DLNK)
                SET_COMP ("rp_mti",                 RP_ACK_DL)
                SET_COMP ("reference",              TP_MR_3N1)
                SKIP_COMP ("rp_data_dl")
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp ack downlink with wrong reference value */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_ACK_DLNK_REF_ERR)
                SET_COMP ("rp_mti",                 RP_ACK_DL)
                SET_COMP ("reference",              BYTE_AA)
                SKIP_COMP ("rp_data_dl")
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp-cause */
BEGIN_MSTRUCT ("rp_cause", RP_CAUSE_CONGESTION)
                SET_COMP ("rp_cause_value",         SMS_RP_CS_CONGESTION)
                SKIP_COMP ("diag")
ENDSTRUCT

/* rp error cause congestion*/
BEGIN_MSTRUCT ("rp_error", RP_ERROR_CONGESTION)
                SET_COMP ("rp_cause",               RP_CAUSE_CONGESTION)
                SKIP_COMP ("rp_user_data")
ENDSTRUCT

/* rp error congestion */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_ERR_CONGESTION)
                SET_COMP ("rp_mti",                 RP_ERROR_DL)
                SET_COMP ("reference",              TP_MR_3N1)
                SKIP_COMP ("rp_data_dl")
                SET_COMP ("rp_error",               RP_ERROR_CONGESTION)
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp-cause*/
BEGIN_MSTRUCT ("rp_cause", RP_CAUSE_PROTOCOL_ERROR)
                SET_COMP ("rp_cause_value",         SMS_RP_CS_PROTOCOL_ERROR)
                SKIP_COMP ("diag")
ENDSTRUCT
```

```
/* rp error cause protocol */
BEGIN_MSTRUCT ("rp_error", RP_ERROR_PROTOCOL)
            SET_COMP ("rp_cause",           RP_CAUSE_PROTOCOL_ERROR)
            SKIP_COMP ("rp_user_data")
ENDSTRUCT

/* rp error protocol*/
BEGIN_MSTRUCT ("cp_user_data_ul", RP_ERR_PROTOCOL)
            SET_COMP ("rp_mti",             RP_ERROR_UL)
            SET_COMP ("reference",          MSG_REF_01)
            SKIP_COMP ("rp_data_ul")
            SET_COMP ("rp_error",           RP_ERROR_PROTOCOL)
            SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp-cause*/
BEGIN_MSTRUCT ("rp_cause", RP_CAUSE_MEM_CAP_EXCEEDED)
            SET_COMP ("rp_cause_value",     SMS_RP_CS_MEM_CAP_EXCEEDED)
            SKIP_COMP ("diag")
ENDSTRUCT

/* rp error cause memory capacity exceeded*/
BEGIN_MSTRUCT ("rp_error", RP_ERROR_MEM_CAP_EXC)
            SET_COMP ("rp_cause",           RP_CAUSE_MEM_CAP_EXCEEDED)
            SKIP_COMP ("rp_user_data")
ENDSTRUCT

/*  rp error memory capacity exceeded*/
BEGIN_MSTRUCT ("cp_user_data_ul", RP_ERR_MEM_CAP_EXC)
            SET_COMP ("rp_mti",             RP_ERROR_UL)
            SET_COMP ("reference",          MSG_REF_01)
            SKIP_COMP ("rp_data_ul")
            SET_COMP ("rp_error",           RP_ERROR_MEM_CAP_EXC)
            SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp-cause*/
BEGIN_MSTRUCT ("rp_cause", RP_CAUSE_TEMP_FAILURE)
            SET_COMP ("rp_cause_value",     SMS_RP_CS_TEMP_FAILURE)
            SKIP_COMP ("diag")
ENDSTRUCT

/* rp error cause protocol */
BEGIN_MSTRUCT ("rp_error", RP_ERROR_TEMP_FAILURE)
            SET_COMP ("rp_cause",           RP_CAUSE_TEMP_FAILURE)
            SKIP_COMP ("rp_user_data")
ENDSTRUCT

/* rp error protocol*/
BEGIN_MSTRUCT ("cp_user_data_dl", RP_ERR_TEMP_FAILURE)
            SET_COMP ("rp_mti",             RP_ERROR_DL)
            SET_COMP ("reference",          MSG_REF_01)
            SKIP_COMP ("rp_data_dl")
            SET_COMP ("rp_error",           RP_ERROR_TEMP_FAILURE)
            SKIP_COMP ("rp_ack")
ENDSTRUCT

/* SDU tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7DEF, 192)
            DELIVER_7DEF
ENDBITBUF
```

```
/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7DEF)
        SET_COMP ("tp_mti",                SMS_DELIVER)
        SET_COMP ("tpdu",                  TPDU_DELIVER_7DEF)
ENDSTRUCT

/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7DEF)
        SET_COMP ("rp_addr",               RP_SCA_12345)
        SET_COMP ("rp_user_data",          RP_UD_DELIVER_7DEF)
ENDSTRUCT

/* rp ack uplink (ref = 0x01) */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_ACK_ULNK)
        SET_COMP ("rp_mti",                RP_ACK_UL)
        SET_COMP ("reference",             MSG_REF_01)
        SKIP_COMP ("rp_data_ul")
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp error protocol (ref = 0x01), with default FCS */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_ERR_ULNK_RESP)
        SET_COMP ("rp_mti",                RP_ERROR_UL)
        SET_COMP ("reference",             MSG_REF_01)
        SKIP_COMP ("rp_data_ul")
        SET_COMP ("rp_error",              RP_ERROR_FCS_UNSPEC)
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7DEF)
        SET_COMP ("rp_mti",                RP_DATA_DL)
        SET_COMP ("reference",             MSG_REF_01)
        SET_COMP ("rp_data_dl",            RP_DELIVER_7DEF)
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp data second deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_2)
        SET_COMP ("rp_mti",                RP_DATA_DL)
        SET_COMP ("reference",             MSG_REF_02)
        SET_COMP ("rp_data_dl",            RP_DELIVER_7DEF)
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp error protocol */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_ERR_PROTOCOL_SECOND)
        SET_COMP ("rp_mti",                RP_ERROR_UL)
        SET_COMP ("reference",             MSG_REF_02)
        SKIP_COMP ("rp_data_ul")
        SET_COMP ("rp_error",              RP_ERROR_PROTOCOL)
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* SDU tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL0, 192)
        DELIVER_7CL0
ENDBITBUF
```

```
/* rp user data deliver*/
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL0)
            SET_COMP ("tp_mti",                 SMS_DELIVER)
            SET_COMP ("tpdu",                   TPDU_DELIVER_7CL0)
ENDSTRUCT

/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL0)
            SET_COMP ("rp_addr",                RP_SCA_12345)
            SET_COMP ("rp_user_data",           RP_UD_DELIVER_7CL0)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL0)
            SET_COMP ("rp_mti",                 RP_DATA_DL)
            SET_COMP ("reference",              MSG_REF_AA)
            SET_COMP ("rp_data_dl",             RP_DELIVER_7CL0)
            SKIP_COMP ("rp_error")
            SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp ack uplink (ref = 0xAA) */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_ACK_RESP)
            SET_COMP ("rp_mti",                 RP_ACK_UL)
            SET_COMP ("reference",              MSG_REF_AA)
            SKIP_COMP ("rp_data_ul")
            SKIP_COMP ("rp_error")
            SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp failure cause (default) */
SET_BITBUF ("tpdu", TPDU_FCS_UNSPEC, 24)
            SMS_DELIVER_REPORT,
            SMS_FCS_UNSPECIFIED,
            BYTE_00
ENDBITBUF

/* rp error data deliver report*/
BEGIN_MSTRUCT ("rp_user_data", RP_UD_FCS_UNSPEC)
            SET_COMP ("tp_mti",                 SMS_DELIVER_REPORT)
            SET_COMP ("tpdu",                   TPDU_FCS_UNSPEC)
ENDSTRUCT

/* rp error cause protocol error with default deliver report */
BEGIN_MSTRUCT ("rp_error", RP_ERROR_FCS_UNSPEC)
            SET_COMP ("rp_cause",               RP_CAUSE_PROTOCOL_ERROR)
            SET_COMP ("rp_user_data",           RP_UD_FCS_UNSPEC)
ENDSTRUCT

/* rp error protocol (ref = 0xAA) */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_ERR_RESP)
            SET_COMP ("rp_mti",                 RP_ERROR_UL)
            SET_COMP ("reference",              MSG_REF_AA)
            SKIP_COMP ("rp_data_ul")
            SET_COMP ("rp_error",               RP_ERROR_FCS_UNSPEC)
            SKIP_COMP ("rp_ack")
ENDSTRUCT

/* SDU tp deliver report */
SET_BITBUF ("tpdu", TPDU_DLVR_REP_ACK, BITLEN_DLVR_REP_ACK)
            DLVR_REP_ACK
ENDBITBUF
```

```
/* rp user data deliver report (acknowledge) */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DLVR_REP_ACK)
                SET_COMP ("tp_mti",                     SMS_DELIVER_REPORT)
                SET_COMP ("tpdu",                       TPDU_DLVR_REP_ACK)
ENDSTRUCT

/* rp ack data */
BEGIN_MSTRUCT ("rp_ack", RP_ACKNL_DLVR_REP)
                SET_COMP ("rp_user_data",               RP_UD_DLVR_REP_ACK)
ENDSTRUCT

/* rp ack uplink (ref = 0xAA) with user data */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_ACK_DLVR_REP)
                SET_COMP ("rp_mti",                     RP_ACK_UL)
                SET_COMP ("reference",                  MSG_REF_AA)
                SKIP_COMP ("rp_data_ul")
                SKIP_COMP ("rp_error")
                SET_COMP ("rp_ack",                     RP_ACKNL_DLVR_REP)
ENDSTRUCT

/* SDU tp deliver report */
SET_BITBUF ("tpdu", TPDU_DLVR_REP_ERR, BITLEN_DLVR_REP_ERR)
                DLVR_REP_ERR
ENDBITBUF

/* rp user data deliver report (error) */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DLVR_REP_ERR)
                SET_COMP ("tp_mti",                     SMS_DELIVER_REPORT)
                SET_COMP ("tpdu",                       TPDU_DLVR_REP_ERR)
ENDSTRUCT

/* rp error cause protocol error with default deliver report */
BEGIN_MSTRUCT ("rp_error", RP_ERROR_DLVR_REP)
                SET_COMP ("rp_cause",                   RP_CAUSE_PROTOCOL_ERROR)
                SET_COMP ("rp_user_data",               RP_UD_DLVR_REP_ERR)
ENDSTRUCT

/* rp error protocol (ref = 0xAA) */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_ERR_DLVR_REP)
                SET_COMP ("rp_mti",                     RP_ERROR_UL)
                SET_COMP ("reference",                  MSG_REF_AA)
                SKIP_COMP ("rp_data_ul")
                SET_COMP ("rp_error",                   RP_ERROR_DLVR_REP)
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp error protocol*/
BEGIN_MSTRUCT ("cp_user_data_ul", RP_ERR_PROTOCOL_AA)
                SET_COMP ("rp_mti",                     RP_ERROR_UL)
                SET_COMP ("reference",                  MSG_REF_AA)
                SKIP_COMP ("rp_data_ul")
                SET_COMP ("rp_error",                   RP_ERROR_PROTOCOL)
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* SDU tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL1, 192)
                DELIVER_7CL1
ENDBITBUF

/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL1)
                SET_COMP ("tp_mti",                     SMS_DELIVER)
                SET_COMP ("tpdu",                       TPDU_DELIVER_7CL1)
ENDSTRUCT
```

```
/*rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL1)
            SET_COMP ("rp_addr",                RP_SCA_12345)
            SET_COMP ("rp_user_data",           RP_UD_DELIVER_7CL1)
ENDSTRUCT

/* rp data deliver*/
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL1)
            SET_COMP ("rp_mti",                 RP_DATA_DL)
            SET_COMP ("reference",              MSG_REF_01)
            SET_COMP ("rp_data_dl",             RP_DELIVER_7CL1)
            SKIP_COMP ("rp_error")
            SKIP_COMP ("rp_ack")
ENDSTRUCT

/* SDU tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL0_43, 192)
            DELIVER_7CL0_43
ENDBITBUF

/* rp user data deliver*/
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_ 7CL0_43)
            SET_COMP ("tp_mti",                 SMS_DELIVER)
            SET_COMP ("tpdu",                   TPDU_DELIVER_7CL0_43)
ENDSTRUCT

/*rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL0_43)
            SET_COMP ("rp_addr",                RP_SCA_12345)
            SET_COMP ("rp_user_data",           RP_UD_DELIVER_7CL0_43)
ENDSTRUCT

/* rp data deliver*/
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL0_43)
            SET_COMP ("rp_mti",                 RP_DATA_DL)
            SET_COMP ("reference",              MSG_REF_01)
            SET_COMP ("rp_data_dl",             RP_DELIVER_7CL0_43)
            SKIP_COMP ("rp_error")
            SKIP_COMP ("rp_ack")
ENDSTRUCT

/* SDU tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL1_43, 192)
            DELIVER_7CL1_43
ENDBITBUF

/* rp user data deliver*/
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL1_43)
            SET_COMP ("tp_mti",                 SMS_DELIVER)
            SET_COMP ("tpdu",                   TPDU_DELIVER_7CL1_43)
ENDSTRUCT

/*rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL1_43)
            SET_COMP ("rp_addr",                RP_SCA_12345)
            SET_COMP ("rp_user_data",           RP_UD_DELIVER_7CL1_43)
ENDSTRUCT
```

```
/* rp data deliver*/
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL1_43)
        SET_COMP ("rp_mti",             RP_DATA_DL)
        SET_COMP ("reference",          MSG_REF_01)
        SET_COMP ("rp_data_dl",         RP_DELIVER_7CL1_43)
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* SDU tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL2_43, 192)
        DELIVER_7CL2_43
ENDBITBUF

/* rp user data deliver*/
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL2_43)
        SET_COMP ("tp_mti",             SMS_DELIVER)
        SET_COMP ("tpdu",               TPDU_DELIVER_7CL2_43)
ENDSTRUCT

/* rp deliver*/
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL2_43)
        SET_COMP ("rp_addr",            RP_SCA_12345)
        SET_COMP ("rp_user_data",       RP_UD_DELIVER_7CL2_43)
ENDSTRUCT

/* rp data deliver*/
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL2_43)
        SET_COMP ("rp_mti",             RP_DATA_DL)
        SET_COMP ("reference",          MSG_REF_01)
        SET_COMP ("rp_data_dl",         RP_DELIVER_7CL2_43)
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* SDU tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL3_43, 192)
        DELIVER_7CL3_43
ENDBITBUF

/* rp user data deliver*/
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL3_43)
        SET_COMP ("tp_mti",             SMS_DELIVER)
        SET_COMP ("tpdu",               TPDU_DELIVER_7CL3_43)
ENDSTRUCT

/*rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL3_43)
        SET_COMP ("rp_addr",            RP_SCA_12345)
        SET_COMP ("rp_user_data",       RP_UD_DELIVER_7CL3_43)
ENDSTRUCT

/* rp data deliver*/
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL3_43)
        SET_COMP ("rp_mti",             RP_DATA_DL)
        SET_COMP ("reference",          MSG_REF_01)
        SET_COMP ("rp_data_dl",         RP_DELIVER_7CL3_43)
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* SDU tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL0L, 1272)
        DELIVER_7CL0L
ENDBITBUF
```

```
/* rp user data deliver*/
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL0L)
                SET_COMP ("tp_mti",             SMS_DELIVER)
                SET_COMP ("tpdu",               TPDU_DELIVER_7CL0L)
ENDSTRUCT

/* rp deliver*/
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL0L)
                SET_COMP ("rp_addr",            RP_SCA_0811112222)
                SET_COMP ("rp_user_data",       RP_UD_DELIVER_7CL0L)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL0L)
                SET_COMP ("rp_mti",             RP_DATA_DL)
                SET_COMP ("reference",          MSG_REF_AA)
                SET_COMP ("rp_data_dl",         RP_DELIVER_7CL0L)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

SET_BITBUF ("tpdu", TPDU_DELIVER_7CL0_DEF, BITLEN_DELIVER_7CL0_DEF)
                DELIVER_7CL0_DEF
ENDBITBUF

BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL0_DEF)
                SET_COMP ("tp_mti",             SMS_DELIVER)
                SET_COMP ("tpdu",               TPDU_DELIVER_7CL0_DEF)
ENDSTRUCT

BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL0_DEF)
                SET_COMP ("rp_addr",            RP_SCA_12345)
                SET_COMP ("rp_user_data",       RP_UD_DELIVER_7CL0_DEF)
ENDSTRUCT

BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL0_DEF)
                SET_COMP ("rp_mti",             RP_DATA_DL)
                SET_COMP ("reference",          MSG_REF_AA)
                SET_COMP ("rp_data_dl",         RP_DELIVER_7CL0_DEF)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

SET_BITBUF ("tpdu", TPDU_DELIVER_7CL0_GDC, BITLEN_DELIVER_7CL0_GDC)
                DELIVER_7CL0_GDC
ENDBITBUF

BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL0_GDC)
                SET_COMP ("tp_mti",             SMS_DELIVER)
                SET_COMP ("tpdu",               TPDU_DELIVER_7CL0_GDC)
ENDSTRUCT

BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL0_GDC)
                SET_COMP ("rp_addr",            RP_SCA_12345)
                SET_COMP ("rp_user_data",       RP_UD_DELIVER_7CL0_GDC)
ENDSTRUCT

BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL0_GDC)
                SET_COMP ("rp_mti",             RP_DATA_DL)
                SET_COMP ("reference",          MSG_REF_AA)
                SET_COMP ("rp_data_dl",         RP_DELIVER_7CL0_GDC)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT
```

```
SET_BITBUF ("tpdu", TPDU_DELIVER_8CL0_DEF, BITLEN_DELIVER_8CL0_DEF)
                DELIVER_8CL0_DEF
ENDBITBUF

BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_8CL0_DEF)
                SET_COMP ("tp_mti",                 SMS_DELIVER)
                SET_COMP ("tpdu",                   TPDU_DELIVER_8CL0_DEF)
ENDSTRUCT

BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_8CL0_DEF)
                SET_COMP ("rp_addr",                RP_SCA_12345)
                SET_COMP ("rp_user_data",           RP_UD_DELIVER_8CL0_DEF)
ENDSTRUCT

BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_8CL0_DEF)
                SET_COMP ("rp_mti",                 RP_DATA_DL)
                SET_COMP ("reference",              MSG_REF_AA)
                SET_COMP ("rp_data_dl",             RP_DELIVER_8CL0_DEF)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

SET_BITBUF ("tpdu", TPDU_DELIVER_8CL0_GDC, BITLEN_DELIVER_8CL0_GDC)
                DELIVER_8CL0_GDC
ENDBITBUF

BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_8CL0_GDC)
                SET_COMP ("tp_mti",                 SMS_DELIVER)
                SET_COMP ("tpdu",                   TPDU_DELIVER_8CL0_GDC)
ENDSTRUCT

BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_8CL0_GDC)
                SET_COMP ("rp_addr",                RP_SCA_12345)
                SET_COMP ("rp_user_data",           RP_UD_DELIVER_8CL0_GDC)
ENDSTRUCT

BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_8CL0_GDC)
                SET_COMP ("rp_mti",                 RP_DATA_DL)
                SET_COMP ("reference",              MSG_REF_AA)
                SET_COMP ("rp_data_dl",             RP_DELIVER_8CL0_GDC)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

SET_BITBUF ("tpdu", TPDU_DELIVER_16CL0_GDC, BITLEN_DELIVER_16CL0_GDC)
                DELIVER_16CL0_GDC
ENDBITBUF

BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_16CL0_GDC)
                SET_COMP ("tp_mti",                 SMS_DELIVER)
                SET_COMP ("tpdu",                   TPDU_DELIVER_16CL0_GDC)
ENDSTRUCT

BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_16CL0_GDC)
                SET_COMP ("rp_addr",                RP_SCA_12345)
                SET_COMP ("rp_user_data",           RP_UD_DELIVER_16CL0_GDC)
ENDSTRUCT

BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_16CL0_GDC)
                SET_COMP ("rp_mti",                 RP_DATA_DL)
                SET_COMP ("reference",              MSG_REF_AA)
                SET_COMP ("rp_data_dl",             RP_DELIVER_16CL0_GDC)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT
```

```
/* tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL2, BITLEN_DELIVER_7CL2)
                DELIVER_7CL2
ENDBITBUF

/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL2)
                SET_COMP ("tp_mti",                     SMS_DELIVER)
                SET_COMP ("tpdu",                       TPDU_DELIVER_7CL2)
ENDSTRUCT

/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL2)
                SET_COMP ("rp_addr",                    RP_SCA_12345)
                SET_COMP ("rp_user_data",               RP_UD_DELIVER_7CL2)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL2)
                SET_COMP ("rp_mti",                     RP_DATA_DL)
                SET_COMP ("reference",                  MSG_REF_01)
                SET_COMP ("rp_data_dl",                 RP_DELIVER_7CL2)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL3, BITLEN_DELIVER_7CL3)
                DELIVER_7CL3
ENDBITBUF

/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL3)
                SET_COMP ("tp_mti",                     SMS_DELIVER)
                SET_COMP ("tpdu",                       TPDU_DELIVER_7CL3)
ENDSTRUCT

/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL3)
                SET_COMP ("rp_addr",                    RP_SCA_12345)
                SET_COMP ("rp_user_data",               RP_UD_DELIVER_7CL3)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL3)
                SET_COMP ("rp_mti",                     RP_DATA_DL)
                SET_COMP ("reference",                  MSG_REF_01)
                SET_COMP ("rp_data_dl",                 RP_DELIVER_7CL3)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL1_42, BITLEN_DELIVER_7CL1_42)
                DELIVER_7CL1_42
ENDBITBUF

/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL1_42)
                SET_COMP ("tp_mti",                     SMS_DELIVER)
                SET_COMP ("tpdu",                       TPDU_DELIVER_7CL1_42)
ENDSTRUCT
```

```
/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL1_42)
                SET_COMP ("rp_addr",                RP_SCA_12345)
                SET_COMP ("rp_user_data",           RP_UD_DELIVER_7CL1_42)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL1_42)
                SET_COMP ("rp_mti",                 RP_DATA_DL)
                SET_COMP ("reference",              MSG_REF_01)
                SET_COMP ("rp_data_dl",             RP_DELIVER_7CL1_42)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/*rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL1_43S)
                SET_COMP ("rp_addr",                RP_SCA_23456)
                SET_COMP ("rp_user_data",           RP_UD_DELIVER_7CL1_43)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL1_43S)
                SET_COMP ("rp_mti",                 RP_DATA_DL)
                SET_COMP ("reference",              MSG_REF_01)
                SET_COMP ("rp_data_dl",             RP_DELIVER_7CL1_43S)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL1_43O, BITLEN_DELIVER_7CL1_43O)
                DELIVER_7CL1_43O
ENDBITBUF

/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL1_43O)
                SET_COMP ("tp_mti",                 SMS_DELIVER)
                SET_COMP ("tpdu",                   TPDU_DELIVER_7CL1_43O)
ENDSTRUCT

/*rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL1_43O)
                SET_COMP ("rp_addr",                RP_SCA_12345)
                SET_COMP ("rp_user_data",           RP_UD_DELIVER_7CL1_43O)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL1_43O)
                SET_COMP ("rp_mti",                 RP_DATA_DL)
                SET_COMP ("reference",              MSG_REF_01)
                SET_COMP ("rp_data_dl",             RP_DELIVER_7CL1_43O)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp submit */
SET_BITBUF ("tpdu", TPDU_SUBMIT_MO, BITLEN_SBM_MO)
                SBM_MO
ENDBITBUF
```

```
/* rp user data submit */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_SUBMIT_MO)
          SET_COMP ("tp_mti",             SMS_SUBMIT)
          SET_COMP ("tpdu",               TPDU_SUBMIT_MO)
ENDSTRUCT

/* rp submit */
BEGIN_MSTRUCT ("rp_data_ul", RP_SUBMIT_MO)
          SET_COMP ("rp_addr",            RP_SCA_12345)
          SET_COMP ("rp_user_data",       RP_UD_SUBMIT_MO)
ENDSTRUCT

/* rp data submit */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_SUBMIT_MO)
          SET_COMP ("rp_mti",             RP_DATA_UL)
          SET_COMP ("reference",          TP_MR_3N1)
          SET_COMP ("rp_data_ul",         RP_SUBMIT_MO)
          SKIP_COMP ("rp_error")
          SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp submit */
SET_BITBUF ("tpdu", TPDU_SUBMIT_DA, BITLEN_SBM_DA)
          SBM_DA
ENDBITBUF

/* rp user data submit */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_SUBMIT_DA)
          SET_COMP ("tp_mti",             SMS_SUBMIT)
          SET_COMP ("tpdu",               TPDU_SUBMIT_DA)
ENDSTRUCT

/* rp submit */
BEGIN_MSTRUCT ("rp_data_ul", RP_SUBMIT_DA)
          SET_COMP ("rp_addr",            RP_SCA_12345)
          SET_COMP ("rp_user_data",       RP_UD_SUBMIT_DA)
ENDSTRUCT

/* rp data submit */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_SUBMIT_DA)
          SET_COMP ("rp_mti",             RP_DATA_UL)
          SET_COMP ("reference",          TP_MR_3N1)
          SET_COMP ("rp_data_ul",         RP_SUBMIT_DA)
          SKIP_COMP ("rp_error")
          SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp submit */
BEGIN_MSTRUCT ("rp_data_ul", RP_SUBMIT_SCA)
          SET_COMP ("rp_addr",            RP_SCA_0811112222)
          SET_COMP ("rp_user_data",       RP_UD_SUBMIT_MO)
ENDSTRUCT

/* rp data submit */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_SUBMIT_SCA)
          SET_COMP ("rp_mti",             RP_DATA_UL)
          SET_COMP ("reference",          TP_MR_3N1)
          SET_COMP ("rp_data_ul",         RP_SUBMIT_SCA)
          SKIP_COMP ("rp_error")
          SKIP_COMP ("rp_ack")
ENDSTRUCT
```

```
/* rp submit */
BEGIN_MSTRUCT ("rp_data_ul", RP_SUBMIT_DA_SCA)
        SET_COMP ("rp_addr",             RP_SCA_0811112222)
        SET_COMP ("rp_user_data",        RP_UD_SUBMIT_DA)
ENDSTRUCT

/* rp data submit */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_SUBMIT_DA_SCA)
        SET_COMP ("rp_mti",              RP_DATA_UL)
        SET_COMP ("reference",           TP_MR_3N1)
        SET_COMP ("rp_data_ul",          RP_SUBMIT_DA_SCA)
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp submit */
SET_BITBUF ("tpdu", TPDU_SUBMIT_7DEF, BITLEN_SBM_7DEF)
        SBM_INIT
ENDBITBUF

/* rp user data submit */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_SUBMIT_7DEF)
        SET_COMP ("tp_mti",              SMS_SUBMIT)
        SET_COMP ("tpdu",                TPDU_SUBMIT_7DEF)
ENDSTRUCT

/* rp submit */
BEGIN_MSTRUCT ("rp_data_ul", RP_SUBMIT_7DEF)
        SET_COMP ("rp_addr",             RP_SCA_12345)
        SET_COMP ("rp_user_data",        RP_UD_SUBMIT_7DEF)
ENDSTRUCT

/* rp data submit */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_SUBMIT_7DEF)
        SET_COMP ("rp_mti",              RP_DATA_UL)
        SET_COMP ("reference",           TP_MR_3N1)
        SET_COMP ("rp_data_ul",          RP_SUBMIT_7DEF)
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp submit */
SET_BITBUF ("tpdu", TPDU_SUBMIT_7DEF_DA, BITLEN_SBM_7DEF_DA)
        SBM_INIT_DA
ENDBITBUF

/* rp user data submit */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_SUBMIT_7DEF_DA)
        SET_COMP ("tp_mti",              SMS_SUBMIT)
        SET_COMP ("tpdu",                TPDU_SUBMIT_7DEF_DA)
ENDSTRUCT

/* rp submit */
BEGIN_MSTRUCT ("rp_data_ul", RP_SUBMIT_7DEF_DA)
        SET_COMP ("rp_addr",             RP_SCA_12345)
        SET_COMP ("rp_user_data",        RP_UD_SUBMIT_7DEF_DA)
ENDSTRUCT
```

```
/* rp data submit */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_SUBMIT_7DEF_DA)
            SET_COMP ("rp_mti",                 RP_DATA_UL)
            SET_COMP ("reference",              TP_MR_3N1)
            SET_COMP ("rp_data_ul",             RP_SUBMIT_7DEF_DA)
            SKIP_COMP ("rp_error")
            SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp submit */
BEGIN_MSTRUCT ("rp_data_ul", RP_SUBMIT_7DEF_SCA)
            SET_COMP ("rp_addr",                RP_SCA_0811112222)
            SET_COMP ("rp_user_data",           RP_UD_SUBMIT_7DEF)
ENDSTRUCT

/* rp data submit */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_SUBMIT_7DEF_SCA)
            SET_COMP ("rp_mti",                 RP_DATA_UL)
            SET_COMP ("reference",              TP_MR_3N1)
            SET_COMP ("rp_data_ul",             RP_SUBMIT_7DEF_SCA)
            SKIP_COMP ("rp_error")
            SKIP_COMP ("rp_ack")
ENDSTRUCT

/* rp submit */
BEGIN_MSTRUCT ("rp_data_ul", RP_SUBMIT_7DEF_DA_SCA)
            SET_COMP ("rp_addr",                RP_SCA_0811112222)
            SET_COMP ("rp_user_data",           RP_UD_SUBMIT_7DEF_DA)
ENDSTRUCT

/* rp data submit */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_SUBMIT_7DEF_DA_SCA)
            SET_COMP ("rp_mti",                 RP_DATA_UL)
            SET_COMP ("reference",              TP_MR_3N1)
            SET_COMP ("rp_data_ul",             RP_SUBMIT_7DEF_DA_SCA)
            SKIP_COMP ("rp_error")
            SKIP_COMP ("rp_ack")
ENDSTRUCT

/* SDU tp command enquiry status request */
SET_BITBUF ("tpdu", TPDU_COMMAND_STAT_REQ, 88)
            COMMAND_STAT_REQ
ENDBITBUF

/* rp user data command status request */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_CMD_STAT_REQ)
            SET_COMP ("tp_mti",                 SMS_COMMAND)
            SET_COMP ("tpdu",                   TPDU_COMMAND_STAT_REQ)
ENDSTRUCT

/* rp command status request */
BEGIN_MSTRUCT ("rp_data_ul", RP_CMD_STAT_REQ)
            SET_COMP ("rp_addr",                RP_SCA_12345)
            SET_COMP ("rp_user_data",           RP_UD_CMD_STAT_REQ)
ENDSTRUCT

/* rp data command status request */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_CMD_STAT_REQ)
            SET_COMP ("rp_mti",                 RP_DATA_UL)
            SET_COMP ("reference",              TP_MR_3N1)
            SET_COMP ("rp_data_ul",             RP_CMD_STAT_REQ)
            SKIP_COMP ("rp_error")
            SKIP_COMP ("rp_ack")
ENDSTRUCT
```

```
/* SDU tp command enquiry on previously submitted short message */
SET_BITBUF ("tpdu", TPDU_COMMAND_ENQ, 88)
                COMMAND_ENQ
ENDBITBUF

/* rp user data enquiry on previously submitted short message */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_CMD_ENQ)
                SET_COMP ("tp_mti",             SMS_COMMAND)
                SET_COMP ("tpdu",               TPDU_COMMAND_ENQ)
ENDSTRUCT

/* rp command enquiry to prev. submitted short message */
BEGIN_MSTRUCT ("rp_data_ul", RP_CMD_ENQ)
                SET_COMP ("rp_addr",            RP_SCA_12345)
                SET_COMP ("rp_user_data",       RP_UD_CMD_ENQ)
ENDSTRUCT

/* rp data command enquiry to prev. submitted short message */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_CMD_ENQ)
                SET_COMP ("rp_mti",             RP_DATA_UL)
                SET_COMP ("reference",          TP_MR_3N1)
                SET_COMP ("rp_data_ul",         RP_CMD_ENQ)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* SDU tp command cancel previously requested status report */
SET_BITBUF ("tpdu", TPDU_COMMAND_CANCEL_REP, 88)
                COMMAND_CANCEL_REP
ENDBITBUF

/* rp user data command cancel previously requested status report */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_CMD_CANCEL_REP)
                SET_COMP ("tp_mti",             SMS_COMMAND)
                SET_COMP ("tpdu",               TPDU_COMMAND_CANCEL_REP)
ENDSTRUCT

/* rp command cancel previously requested status report*/
BEGIN_MSTRUCT ("rp_data_ul", RP_CMD_CANCEL_REP)
                SET_COMP ("rp_addr",            RP_SCA_12345)
                SET_COMP ("rp_user_data",       RP_UD_CMD_CANCEL_REP)
ENDSTRUCT

/* rp data command cancel previously requested status report*/
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_CMD_CANCEL_REP)
                SET_COMP ("rp_mti",             RP_DATA_UL)
                SET_COMP ("reference",          TP_MR_3N1)
                SET_COMP ("rp_data_ul",         RP_CMD_CANCEL_REP)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* SDU tp command delete */
SET_BITBUF ("tpdu", TPDU_COMMAND_DEL, 88)
                COMMAND_DEL
ENDBITBUF

/* rp user data delete */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_CMD_DEL)
                SET_COMP ("tp_mti",             SMS_COMMAND)
                SET_COMP ("tpdu",               TPDU_COMMAND_DEL)
ENDSTRUCT
```

```
/* rp command delete */
BEGIN_MSTRUCT ("rp_data_ul", RP_CMD_DEL)
                SET_COMP ("rp_addr",                RP_SCA_12345)
                SET_COMP ("rp_user_data",           RP_UD_CMD_DEL)
ENDSTRUCT

/* rp data command delete */
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_CMD_DEL)
                SET_COMP ("rp_mti",                 RP_DATA_UL)
                SET_COMP ("reference",              TP_MR_3N2)
                SET_COMP ("rp_data_ul",             RP_CMD_DEL)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp status report */
SET_BITBUF ("tpdu", TPDU_STATUS_REP, 184)
                STATUS_REP
ENDBITBUF

/* rp user data status report */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_STATUS_REP)
                SET_COMP ("tp_mti",                 SMS_STATUS_REPORT)
                SET_COMP ("tpdu",                   TPDU_STATUS_REP)
ENDSTRUCT

/* rp status report */
BEGIN_MSTRUCT ("rp_data_dl", RP_STATUS_REP)
                SET_COMP ("rp_addr",                RP_SCA_12345)
                SET_COMP ("rp_user_data",           RP_UD_STATUS_REP)
ENDSTRUCT

/* rp data status report */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_STATUS_REP)
                SET_COMP ("rp_mti",                 RP_DATA_DL)
                SET_COMP ("reference",              MSG_REF_01)
                SET_COMP ("rp_data_dl",             RP_STATUS_REP)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* CPHS VMW */
BEGINARRAY_PART (CPHS_VMW_DATA, 1)
                CPHS_VMW_BYTE1_L1W
ENDARRAY

/* IMSI normal */
BEGINARRAY_PART (IMSI_NORMAL, 9)
                8, 0x21, 0x21, 0x10, 0x21, 0x43, 0x65, 0x87, 0xF9
ENDARRAY

/* IMSI One2One (234-30) */
BEGINARRAY_PART (IMSI_ONE2ONE, 9)
                7, 0x29, 0x43, 0x03, 0x78, 0x56, 0x34, 0x12, 0xFF
ENDARRAY

/* tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_121_A, BITLEN_DELIVER_121_A)
                DELIVER_121_A
ENDBITBUF
```

```
/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_121_A)
        SET_COMP ("tp_mti",                 SMS_DELIVER)
        SET_COMP ("tpdu",                   TPDU_DELIVER_121_A)
ENDSTRUCT

/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_121_A)
        SET_COMP ("rp_addr",                RP_SCA_12345)
        SET_COMP ("rp_user_data",           RP_UD_DELIVER_121_A)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_121_A)
        SET_COMP ("rp_mti",                 RP_DATA_DL)
        SET_COMP ("reference",              MSG_REF_01)
        SET_COMP ("rp_data_dl",             RP_DELIVER_121_A)
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_121_B, BITLEN_DELIVER_121_B)
        DELIVER_121_B
ENDBITBUF

/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_121_B)
        SET_COMP ("tp_mti",                 SMS_DELIVER)
        SET_COMP ("tpdu",                   TPDU_DELIVER_121_B)
ENDSTRUCT

/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_121_B)
        SET_COMP ("rp_addr",                RP_SCA_12345)
        SET_COMP ("rp_user_data",           RP_UD_DELIVER_121_B)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_121_B)
        SET_COMP ("rp_mti",                 RP_DATA_DL)
        SET_COMP ("reference",              MSG_REF_01)
        SET_COMP ("rp_data_dl",             RP_DELIVER_121_B)
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_121_C, BITLEN_DELIVER_121_C)
        DELIVER_121_C
ENDBITBUF

/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_121_C)
        SET_COMP ("tp_mti",                 SMS_DELIVER)
        SET_COMP ("tpdu",                   TPDU_DELIVER_121_C)
ENDSTRUCT

/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_121_C)
        SET_COMP ("rp_addr",                RP_SCA_12345)
        SET_COMP ("rp_user_data",           RP_UD_DELIVER_121_C)
ENDSTRUCT
```

```
/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_121_C)
                SET_COMP ("rp_mti",             RP_DATA_DL)
                SET_COMP ("reference",          MSG_REF_01)
                SET_COMP ("rp_data_dl",         RP_DELIVER_121_C)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_EMPTY, BITLEN_DELIVER_ EMPTY)
                DELIVER_EMPTY
ENDBITBUF

/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_EMPTY)
                SET_COMP ("tp_mti",             SMS_DELIVER)
                SET_COMP ("tpdu",               TPDU_DELIVER_EMPTY)
ENDSTRUCT

/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_ EMPTY)
                SET_COMP ("rp_addr",            RP_SCA_12345)
                SET_COMP ("rp_user_data",       RP_UD_DELIVER_EMPTY)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_ EMPTY)
                SET_COMP ("rp_mti",             RP_DATA_DL)
                SET_COMP ("reference",          MSG_REF_01)
                SET_COMP ("rp_data_dl",         RP_DELIVER_ EMPTY)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL2_SAT1, BITLEN_DELIVER_7CL2_SAT1)
                DELIVER_7CL2_SAT1
ENDBITBUF

/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL2_SAT1)
                SET_COMP ("tp_mti",             SMS_DELIVER)
                SET_COMP ("tpdu",               TPDU_DELIVER_7CL2_SAT1)
ENDSTRUCT

/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL2_SAT1)
                SET_COMP ("rp_addr",            RP_SCA_12345)
                SET_COMP ("rp_user_data",       RP_UD_DELIVER_7CL2_SAT1)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL2_SAT1)
                SET_COMP ("rp_mti",             RP_DATA_DL)
                SET_COMP ("reference",          MSG_REF_01)
                SET_COMP ("rp_data_dl",         RP_DELIVER_7CL2_SAT1)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL2_ SAT2, BITLEN_DELIVER_7CL2_SAT2)
                DELIVER_7CL2_SAT2
ENDBITBUF
```

```
/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL2_SAT2)
        SET_COMP ("tp_mti",             SMS_DELIVER)
        SET_COMP ("tpdu",               TPDU_DELIVER_7CL2_SAT2)
ENDSTRUCT

/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL2_SAT2)
        SET_COMP ("rp_addr",            RP_SCA_12345)
        SET_COMP ("rp_user_data",       RP_UD_DELIVER_7CL2_SAT2)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL2_SAT2)
        SET_COMP ("rp_mti",             RP_DATA_DL)
        SET_COMP ("reference",          MSG_REF_01)
        SET_COMP ("rp_data_dl",         RP_DELIVER_7CL2_SAT2)
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_7CL1_SAT3, BITLEN_DELIVER_7CL1_SAT3)
        DELIVER_7CL1_SAT3
ENDBITBUF

/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_7CL1_SAT3)
        SET_COMP ("tp_mti",             SMS_DELIVER)
        SET_COMP ("tpdu",               TPDU_DELIVER_7CL1_SAT3)
ENDSTRUCT

/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_7CL1_SAT3)
        SET_COMP ("rp_addr",            RP_SCA_12345)
        SET_COMP ("rp_user_data",       RP_UD_DELIVER_7CL1_SAT3)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_7CL1_SAT3)
        SET_COMP ("rp_mti",             RP_DATA_DL)
        SET_COMP ("reference",          MSG_REF_01)
        SET_COMP ("rp_data_dl",         RP_DELIVER_7CL1_SAT3)
        SKIP_COMP ("rp_error")
        SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_8CL2_SAT1, BITLEN_DELIVER_8CL2_SAT1)
        DELIVER_8CL2_SAT1
ENDBITBUF

/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_8CL2_SAT1)
        SET_COMP ("tp_mti",             SMS_DELIVER)
        SET_COMP ("tpdu",               TPDU_DELIVER_8CL2_SAT1)
ENDSTRUCT

/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_8CL2_SAT1)
        SET_COMP ("rp_addr",            RP_SCA_0811112222)
        SET_COMP ("rp_user_data",       RP_UD_DELIVER_8CL2_SAT1)
ENDSTRUCT
```

```
/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_8CL2_SAT1)
                SET_COMP ("rp_mti",                 RP_DATA_DL)
                SET_COMP ("reference",              MSG_REF_01)
                SET_COMP ("rp_data_dl",             RP_DELIVER_8CL2_SAT1)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* tp deliver */
SET_BITBUF ("tpdu", TPDU_DELIVER_8CL2_SAT2, BITLEN_DELIVER_8CL2_SAT2)
                DELIVER_8CL2_SAT2
ENDBITBUF

/* rp user data deliver */
BEGIN_MSTRUCT ("rp_user_data", RP_UD_DELIVER_8CL2_SAT2)
                SET_COMP ("tp_mti",                 SMS_DELIVER)
                SET_COMP ("tpdu",                   TPDU_DELIVER_8CL2_SAT2)
ENDSTRUCT

/* rp deliver */
BEGIN_MSTRUCT ("rp_data_dl", RP_DELIVER_8CL2_SAT2)
                SET_COMP ("rp_addr",                RP_SCA_12345)
                SET_COMP ("rp_user_data",           RP_UD_DELIVER_8CL2_SAT2)
ENDSTRUCT

/* rp data deliver */
BEGIN_MSTRUCT ("cp_user_data_dl", RP_DATA_DELIVER_8CL2_SAT2)
                SET_COMP ("rp_mti",                 RP_DATA_DL)
                SET_COMP ("reference",              MSG_REF_01)
                SET_COMP ("rp_data_dl",             RP_DELIVER_8CL2_SAT2)
                SKIP_COMP ("rp_error")
                SKIP_COMP ("rp_ack")
ENDSTRUCT

/* Envelope SMS Download*/

BEGINARRAY_PART (ENVELOPE_SMS_1_CMD, 38)
                0xD1, 36,                           /* BER-TLV SMS-PP Download */
                0x82, 0x02, 0x83, 0x81,             /* TLV Device Id. */
                0x06, RP_ADDR_12345,                /* TLV Address */
                0x8B, 24, DELIVER_7CL2_SAT1         /* TLV SMS-TPDU */
ENDARRAY

BEGIN_PSTRUCT ("stk_cmd", ENVELOPE_SMS_1)
                SET_COMP ("l_cmd",                  304)
                SET_COMP ("o_cmd",                  0)
                SET_COMP ("cmd",                    ENVELOPE_SMS_1_CMD)
ENDSTRUCT

BEGINARRAY_PART (ENVELOPE_SMS_2_CMD, 172)
                0xD1, 0x81, 169,
                0x82, 0x02, 0x83, 0x81,
                0x06, RP_ADDR_12345,
                0x8B, 0x81, 156, DELIVER_7CL2_SAT2
ENDARRAY

BEGIN_PSTRUCT ("stk_cmd", ENVELOPE_SMS_2)
                SET_COMP ("l_cmd",                  1376)
                SET_COMP ("o_cmd",                  0)
                SET_COMP ("cmd",                    ENVELOPE_SMS_2_CMD)
ENDSTRUCT
```

```
BEGINARRAY_PART (ENVELOPE_SMS_3_CMD, 41)
            0xD1,                           /* sms-pp download tag*/
            39,                             /* length of command*/
            0x82, 0x02, 0x83, 0x81,         /* device identity*/
            0x06, RP_ADDR_0811112222,       /* address tag, length, address*/
            0x8B, 25,                       /* sms-tpdu tag, length*/
            DELIVER_8CL2_SAT1
ENDARRAY

BEGIN_PSTRUCT ("stk_cmd", ENVELOPE_SMS_3)
            SET_COMP ("l_cmd",              328)
            SET_COMP ("o_cmd",              0)
            SET_COMP ("cmd", ENVELOPE_SMS_3_CMD)
ENDSTRUCT

BEGINARRAY_PART (ENVELOPE_SMS_4_CMD, 39)
            0xD1,                           /* sms-pp download tag*/
            37,                             /* length of command*/
            0x82, 0x02, 0x83, 0x81,         /* device identity*/
            0x06, RP_ADDR_12345,            /* address tag, length, address*/
            0x8B, 25,                       /* sms-tpdu tag, length*/
            DELIVER_8CL2_SAT2
ENDARRAY

BEGIN_PSTRUCT ("stk_cmd", ENVELOPE_SMS_4)
            SET_COMP ("l_cmd",              312)
            SET_COMP ("o_cmd",              0)
            SET_COMP ("cmd", ENVELOPE_SMS_4_CMD)
ENDSTRUCT

BEGINARRAY_PART (ENVELOPE_SMS_121_C_CMD, 174)
            0xD1,                           /* sms-pp download tag*/
            0x81, 171,                      /* length of command*/
            0x82, 0x02, 0x83, 0x81,         /* device identity*/
            0x06, RP_ADDR_12345,            /* address tag, length, address*/
            0x8B, 0x81, 158,                /* sms-tpdu tag, length*/
            DELIVER_121_C
ENDARRAY

BEGIN_PSTRUCT ("stk_cmd", ENVELOPE_SMS_121_C)
            SET_COMP ("l_cmd",              1392)
            SET_COMP ("o_cmd",              0)
            SET_COMP ("cmd", ENVELOPE_SMS_121_C_CMD)
ENDSTRUCT

BEGIN_PSTRUCT ("stk_cmd", STK_CMD_EMPTY)
            SET_COMP ("l_cmd",              0)
            SET_COMP ("o_cmd",              0)
            SKIP_COMP ("cmd")
ENDSTRUCT

#if 0       /* This way of defining a SDU is not supported by TAP2 */
BEGIN_PSTRUCT ("sdu", CP_DATA_RP_ERROR_SAT_BUSY)
            SET_COMP ("l_buf",              0x0070)
            SET_COMP ("o_buf",              0x0018)
            SET_COMP ("buf",                CP_DATA_RP_ERROR_SAT_BUSY_BUF)
ENDSTRUCT
#endif

BEGINARRAY_PART (STK_CMD_TPDU_7BIT, 8)
            TEXT7_RSTUVWXYZ
ENDARRAY
```

```
BEGIN_PSTRUCT ("stk_cmd", STK_CMD_TPDU1)
        SET_COMP ("l_cmd",                  64)
        SET_COMP ("o_cmd",                  0)
        SET_COMP ("cmd",                    STK_CMD_TPDU_7BIT)
ENDSTRUCT

BEGINARRAY_PART (STK_CMD_TPDU_8BIT, 9)
        TEXT8_ABCDEFGHI
ENDARRAY

BEGIN_PSTRUCT ("stk_cmd", STK_CMD_TPDU2)
        SET_COMP ("l_cmd",                  72)
        SET_COMP ("o_cmd",                  0)
        SET_COMP ("cmd",                    STK_CMD_TPDU_8BIT)
ENDSTRUCT
```

#if 0 /* not used */

```
BEGIN_MSTRUCT ("sms_submit_abs", TP_SUBMIT_PID5F_ABS)
        SET_COMP ("tp_mr",                  TP_MR_3N1)
        SET_COMP ("tp_da",                  TP_DA_654321)
        SET_COMP ("tp_pid",                 SMS_PID_RET_CALL_MSG)
        SET_COMP ("tp_dcs",                 DCS_DEF)
        SET_COMP ("tp_scts",                TP_SCTS_9801071234564)
        SET_COMP ("tp_ud",                  TP_UD_SM7_ABCDEFGHI)
ENDSTRUCT

/* rp user data submit*/
BEGIN_MSTRUCT ("rp_user_data_ul", RP_UD_SUBMIT_PID5F_ABS)
        SET_COMP ("tp_rp",                  SMS_RP_NOT_SET)
        SET_COMP ("tp_udhi",                SMS_UDHI_NOT_INCLUDED)
        SET_COMP ("tp_srr",                 SMS_SRR_NOT_REQUESTED)
        SET_COMP ("tp_vpf",                 SMS_VPF_ABSOLUTE)
        SET_COMP ("tp_rd",                  SMS_RD_REJECT)
        SET_COMP ("tp_mti",                 SMS_SUBMIT)
        SKIP_COMP ("sms_command")
        SKIP_COMP ("sms_submit_not")
        SKIP_COMP ("sms_submit_rel")
        SET_COMP ("sms_submit_abs",         TP_SUBMIT_PID5F_ABS)
ENDSTRUCT

/* rp submit*/
BEGIN_MSTRUCT ("rp_data_ul", RP_SUBMIT_PID5F_ABS)
        SET_COMP ("rp_dest_addr",           RP_DA_12345)
        SET_COMP ("rp_user_data_ul",        RP_UD_SUBMIT_PID5F_ABS)
ENDSTRUCT

/* rp data submit for PID5F*/
BEGIN_MSTRUCT ("cp_user_data_ul", RP_DATA_SUBMIT_PID5F_ABS)
        SET_COMP ("mti",                    RP_DATA_UL)
        SET_COMP ("reference",              TP_MR_3N1)
        SET_COMP ("rp_data_ul",             RP_SUBMIT_PID5F_ABS)
        SKIP_COMP ("rp_error_ul")
        SKIP_COMP ("rp_ack_ul")
ENDSTRUCT

/* rp-cause*/
BEGIN_MSTRUCT ("rp_cause", RP_CAUSE_SEM_INC)
        SET_COMP ("rp_cause_value",         SMS_RP_CS_SEM_INC_MSG)
        SKIP_COMP ("diag")
ENDSTRUCT

/* rp error cause semanitcal incorrect message*/
BEGIN_MSTRUCT ("rp_error_ul", RP_ERROR_SEM_INC)
```

```
          SET_COMP ("rp_cause",               RP_CAUSE_SEM_INC)
          SKIP_COMP ("rp_error_data_ul")
ENDSTRUCT

/* rp error protocol*/
BEGIN_MSTRUCT ("cp_user_data_ul", RP_ERR_SEM_INC)
          SET_COMP ("mti",                    RP_ERROR_UL)
          SET_COMP ("reference",              MSG_REF_AA)
          SKIP_COMP ("rp_data_ul")
          SET_COMP ("rp_error_ul",            RP_ERROR_SEM_INC)
          SKIP_COMP ("rp_ack_ul")
ENDSTRUCT
```

#endif

3 TEST CASES

3.1 Routing (internal) and Configuration

3.1.1 SMS000: Setup the routing and PCO view for the SMS test

Description: Routings for the SMS tests are set.
(Note: All test cases need the "new" TAP / TAP2)

Preamble: None

```
   MMI                              SMS                              SIM/MM
    |                                |                                 |
MUTE (1000)
    |                                |                                 |
COMMAND (TAP RESET)
COMMAND (MMI RESET)
COMMAND (CC RESET)
COMMAND (SS RESET)
COMMAND (SMS RESET)
COMMAND (MM RESET)
COMMAND (RR RESET)
COMMAND (DL RESET)
COMMAND (SIM RESET)
COMMAND (PL RESET)
    |                                |                                 |
COMMAND (TAP REDIRECT CLEAR)
COMMAND (MMI REDIRECT CLEAR)
COMMAND (CC REDIRECT CLEAR)
COMMAND (SS REDIRECT CLEAR)
COMMAND (SMS REDIRECT CLEAR)
COMMAND (MM REDIRECT CLEAR)
COMMAND (RR REDIRECT CLEAR)
COMMAND (DL REDIRECT CLEAR)
COMMAND (SIM REDIRECT CLEAR)
COMMAND (PL REDIRECT CLEAR)
    |                                |                                 |
COMMAND (MMI REDIRECT MM NULL)
COMMAND (MMI REDIRECT CC NULL)
COMMAND (MMI REDIRECT SS NULL)
COMMAND (MMI REDIRECT SMS NULL)
COMMAND (MMI REDIRECT PL NULL)
    |                                |                                 |
COMMAND (SMS REDIRECT MMI TAP)
COMMAND (SMS REDIRECT MM TAP)
COMMAND (SMS REDIRECT SIM TAP)
    |                                |                                 |
COMMAND (SS REDIRECT MMI NULL)
COMMAND (SS REDIRECT MM NULL)
    |                                |                                 |
COMMAND (CC REDIRECT MMI NULL)
COMMAND (CC REDIRECT MM NULL)
    |                                |                                 |
COMMAND (MM REDIRECT MMI NULL)
COMMAND (MM REDIRECT CC NULL)
COMMAND (MM REDIRECT SS NULL)
COMMAND (MM REDIRECT SMS NULL)
COMMAND (MM REDIRECT SIM NULL)
COMMAND (MM REDIRECT RR NULL)
COMMAND (MM REDIRECT DL NULL)
    |                                |                                 |
COMMAND (RR REDIRECT PL NULL)
COMMAND (RR REDIRECT DL NULL)
COMMAND (RR REDIRECT MM NULL)
    |                                |                                |
COMMAND (DL REDIRECT RR NULL)
COMMAND (DL REDIRECT MM NULL)
COMMAND (DL REDIRECT PL NULL)
    |                                |                                |
COMMAND (PL REDIRECT RR NULL)
COMMAND (PL REDIRECT DL NULL)
COMMAND (PL REDIRECT MMI NULL)
    |                                |                                |
COMMAND (SIM REDIRECT MM NULL)
COMMAND (SIM REDIRECT MMI NULL)
    |                                |                                |
COMMAND (TAP REDIRECT TAP SMS)
    |                                |                                |
COMMAND (SMS STATUS PARTITION)
    |                                |                                |
```

Parametrization

Primitive	Parameter	Value

History: 6-Jan-98 SZ Initial

3.1.2 SMS001: ME Memory not available

Description: Configuration Command to disable ME memory..

Preamble: SMS000

```
   MMI                              SMS                             SIM/MM
    |                                |                                |
COMMAND (PL CONFIG STD=1)
    |                                |                                |
```

Parametrization

Primitive	Parameter	Value

History: 14-Jun-2002 FK Initial

3.1.3 SMS002: ME Memory available (3 Records, empty)

Description: Configuration Command to enable ME memory..

Preamble: SMS000

```
   MMI                              SMS                             SIM/MM
    |                                |                                |
COMMAND (PL CONFIG STD=3)
    |                                |                                |
```

Parametrization

Primitive	Parameter	Value

History: 14-Jun-2002 FK Initial

3.1.4 SMS003: ME Memory available (3 Records, 2 occupied)

Description: Configuration Command to enable ME memory..

Preamble: SMS000

```
   MMI                               SMS                            SIM/MM
    |                                 |                                 |
COMMAND (PL CONFIG STD=4)
    |                                 |                                 |
```

Parametrization

Primitive	Parameter	Value

History: 14-Jun-2002 FK Initial

3.1.5 SMS004: ME Memory available (3 Records, all occupied)

Description: Configuration Command to enable ME memory..

Preamble: SMS000

```
   MMI                               SMS                            SIM/MM
    |                                 |                                 |
COMMAND (PL CONFIG STD=5)
    |                                 |                                 |
```

Parametrization

Primitive	Parameter	Value

History: 14-Jun-2002 FK Initial

3.2 Initialisation Phase

3.2.1 SMS005: Start with CPHS and IMSI Check

Description: The SIM application sends the initial parameters read from the SIM card. SMS then checks the existence of the CPHS data field 'Voice Message Waiting Flag'. Depending on the result SMS will ckeck MT-SMs for CPHS Voice Mail Indications. Additionally EF(IMSI) is read to determine the necessity for operator-specific Short Message handling. ME memory is not available.

Variants: <A>...<G>

Preamble: SMS001

```
   MMI                               SMS                            SIM/MM
    |                                 |                                 |
(1)  |                                |        SIM_SMS_INSERT_IND       |
    |                                 * <============================== *
(2)  |        MNSMS_REPORT_IND        |                                 |
    * <============================== *                                 |
(3)  |                                |           SIM_READ_REQ          |
    |                                 * ==============================> *
MUTE(500)
(4)  |                                |           SIM_READ_CNF          |
    |                                 * <============================== *
(5)  |                                |           SIM_READ_REQ          |
    |                                 * ==============================> *
MUTE(500)
(6)  |                                |           SIM_READ_CNF          |
    |                                 * <============================== *
    |                                 |                                 |
```

Parametrization

Primitive		Parameter	Value
(1)	SIM_SMS_INSERT_IND		
		phase	PHASE_2_SIM
		tp_mr	TP_MR_3
	<A>	mem_cap_avail	TRUE
	<B>	mem_cap_avail	FALSE
	<C>	mem_cap_avail	TRUE
	<D>	mem_cap_avail	TRUE
	<E>	mem_cap_avail	FALSE
	<F>	mem_cap_avail	TRUE
	<G>	mem_cap_avail	FALSE
	<A>	download_sms	FALSE
	<B>	download_sms	FALSE
	<C>	download_sms	TRUE
	<D>	download_sms	FALSE
	<E>	download_sms	FALSE
	<F>	download_sms	TRUE
	<G>	download_sms	FALSE
		smsr_mem_cap	SIM_SMSR_DISABLE
(2)	MNSMS_REPORT_IND		
		state	SMS_STATE_INITIALISING
(3)	SIM_READ_REQ		
		source	SRC_SMS
		offset	OFFSET_0
		datafield	SIM_CPHS_VMW
		length	LEN_1
		max_length	LEN_0
(4)	SIM_READ_CNF		
		datafield	SIM_CPHS_VMW
	<A>	cause	SIM_CAUSE_UNKN_FILE_ID
	<B>	cause	SIM_CAUSE_UNKN_FILE_ID
	<C>	cause	SIM_CAUSE_UNKN_FILE_ID
	<D>	cause	SIM_NO_ERROR
	<E>	cause	SIM_NO_ERROR
	<F>	cause	SIM_CAUSE_UNKN_FILE_ID
	<G>	cause	SIM_CAUSE_UNKN_FILE_ID
	<A>	length	LEN_0
	<B>	length	LEN_0
	<C>	length	LEN_0
	<D>	length	LEN_1
	<E>	length	LEN_1
	<F>	length	LEN_0
	<G>	length	LEN_0
	<A>	trans_data	SIM_NO_DATA
	<B>	trans_data	SIM_NO_DATA
	<C>	trans_data	SIM_NO_DATA
	<D>	trans_data	CPHS_VMW_DATA
	<E>	trans_data	CPHS_VMW_DATA
	<F>	trans_data	SIM_NO_DATA
	<G>	trans_data	SIM_NO_DATA

(5)	SIM_READ_REQ		
		source	SRC_SMS
		offset	OFFSET_0
		datafield	SIM_IMSI
		length	LEN_9
		max_length	LEN_0
(6)	SIM_READ_CNF		
		datafield	SIM_IMSI
		cause	SIM_NO_ERROR
		length	LEN_9
	<A>	trans_data	IMSI_NORMAL
	<B>	trans_data	IMSI_NORMAL
	<C>	trans_data	IMSI_NORMAL
	<D>	trans_data	IMSI_NORMAL
	<E>	trans_data	IMSI_NORMAL
	<F>	trans_data	IMSI_ONE2ONE
	<G>	trans_data	IMSI_ONE2ONE

History:	20-Dec-2001	FK	Initial
	19-Apr-2002	FK	IMSI read procedure added
	25-Oct-2002	FK	Adaption to Cause Concept

3.2.2 SMS006: Start with CPHS and IMSI Check (with ME Memory)

Description: The SIM application sends the initial parameters read from the SIM card. SMS then checks the existence of the CPHS data field 'Voice Message Waiting Flag'. Depending on the result SMS will ckeck MT-SMs for CPHS Voice Mail Indications. Additionally EF(IMSI) is read to determine the necessity for operator-specific Short Message handling. ME memory is available and empty (variant A, B), partly filled (variant C, D) or full (variant E, F). The Memory Available Indication is set (variant A, C, E) or unset (variant B, D, F).

Variants: <A>...<F>

Preamble:

<A>	SMS002
<B>	SMS002
<C>	SMS003
<D>	SMS003
<E>	SMS004
<F>	SMS004

```
   MMI                                 SMS                                   SIM/MM
    |                                   |                                      |
(7) |                                   |         SIM_SMS_INSERT_IND           |
    |                                   * <================================== *
(8) |         MNSMS_REPORT_IND          |                                      |
    * <============================== *                                      |
    |                                   |                                      |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_SMS_INSERT_IND		
	phase	PHASE_2_SIM
	tp_mr	TP_MR_3
<A>	mem_cap_avail	TRUE
<B>	mem_cap_avail	FALSE
<C>	mem_cap_avail	TRUE
<D>	mem_cap_avail	FALSE
<E>	mem_cap_avail	TRUE
<F>	mem_cap_avail	FALSE
	download_sms	FALSE
	smsr_mem_cap	SIM_SMSR_DISABLE
(2) MNSMS_REPORT_IND		
	state	SMS_STATE_INITIALISING

History: 14-Jun-2002 FK Initial

3.2.3 SMS007: No Short Messages available (ME Memory)

Description: SMS sends an indication to MMI that ME memory is available but empty..

Variants: <A>...<B>

Preamble:

<A> SMS006A
<B> SMS006B

```
   MMI                                 SMS                                SIM/MM
    |                                   |                                   |
(1) |         MNSMS_MESSAGE_IND         |                                   |
    * <============================== *                                   |
    |                                   |                                   |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_MESSAGE_IND		
	mem_type	MEM_ME
	rec_num	SMS_RECORD_NOT_EXIST
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_FREE
	sms_sdu	SMS_SDU_EMPTY

History: 14-Jun-2002 FK Initial

3.2.4 SMS008: MO and MT-SM available (ME Memory)

Description: SMS sends an indication to MMI about each record of ME memory which is occupied.

Variants: <A>...<B>

Preamble:

<A> SMS006C
<B> SMS006D

```
   MMI                                 SMS                                SIM/MM
    |                                   |                                   |
(1) |         MNSMS_MESSAGE_IND         |                                   |
    * <============================== *                                   |
(2) |         MNSMS_MESSAGE_IND         |                                   |
    * <============================== *                                   |
    |                                   |                                   |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_MESSAGE_IND		
	mem_type	MEM_ME
	rec_num	SIM_RECORD_1
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_MT_7CL1
(2) MNSMS_MESSAGE_IND		
	mem_type	MEM_ME
	rec_num	SIM_RECORD_2
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_STO_SENT
	sms_sdu	SMS_SDU_MO

History: 14-Jun-2002 FK Initial

3.2.5 SMS009: MO and MT-SM available (ME Memory full)

Description: SMS sends an indication to MMI about all records of ME memory.

Variants: <A>...<B>

Preamble:

<A> SMS006E

<B> SMS006F

```
   MMI                              SMS                              SIM/MM
    |                                |                                  |
(1) |        MNSMS_MESSAGE_IND       |                                  |
    * <============================== *                                  |
(2) |        MNSMS_MESSAGE_IND       |                                  |
    * <============================== *                                  |
(3) |        MNSMS_MESSAGE_IND       |                                  |
    * <============================== *                                  |
    |                                |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_MESSAGE_IND		
	mem_type	MEM_ME
	rec_num	SIM_RECORD_1
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_MT_7CL1
(2) MNSMS_MESSAGE_IND		
	mem_type	MEM_ME
	rec_num	SIM_RECORD_2
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_STO_SENT
	sms_sdu	SMS_SDU_MO
(3) MNSMS_MESSAGE_IND		
	mem_type	MEM_ME
	rec_num	SIM_RECORD_3
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_STO_UNSENT
	sms_sdu	SMS_SDU_MO_ABS

History: 14-Jun-2002 FK Initial

3.2.6 SMS010: Read and check additonal SIM Properties

Description: After checking the ME memory, the availability of CPHS is checked and EF(IMSI) is read to check for operator-specific features..

Preamble:

<A>	SMS007A
<B>	SMS007B
<C>	SMS008A
<D>	SMS008B
<E>	SMS009A
<F>	SMS009B

```
   MMI                                  SMS                         SIM/MM
     |                                   |                               |
(9)  |                                   |         SIM_READ_REQ          |
     |                                   * ==============================> *
MUTE(500)
(10) |                                   |         SIM_READ_CNF          |
     |                                   * <============================== *
(11) |                                   |         SIM_READ_REQ          |
     |                                   * ==============================> *
MUTE(500)
(12) |                                   |         SIM_READ_CNF          |
     |                                   * <============================== *
     |                                   |                               |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_READ_REQ		
	source	SRC_SMS
	offset	OFFSET_0
	datafield	SIM_CPHS_VMW
	length	LEN_1
	max_length	LEN_0
(2) SIM_READ_CNF		
	datafield	SIM_CPHS_VMW
	cause	SIM_CAUSE_UNKN_FILE_ID
	length	LEN_0
	trans_data	SIM_NO_DATA
(3) SIM_READ_REQ		
	source	SRC_SMS
	offset	OFFSET_0
	datafield	SIM_IMSI
	length	LEN_9
	max_length	LEN_0
(4) SIM_READ_CNF		
	datafield	SIM_IMSI
	cause	SIM_NO_ERROR
	length	LEN_9
	trans_data	IMSI_NORMAL

History:	14-Jun-2002	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.2.7 SMS012: No Short Messages available (SIM Memory)

Description: The SIM application sends the initial parameters read from the SIM card. SMS has checked the memory in the mobile equipment before, if necessary, and starts searching for SMS records on the SIM card. The SIM card memory has three records. All records are unused.

Variants: <A>...<F>

Preamble:

<A>	SMS005A
<B>	SMS005D
<C>	SMS005F
<D>	SMS010A
<E>	SMS010C
<F>	SMS010E

```
   MMI                                SMS                             SIM/MM
     |                                 |                                 |
(2)  |                                 |       SIM_READ_RECORD_REQ       |
     |                                 * ==============================> *
(3)  |                                 |       SIM_READ_RECORD_CNF       |
     |                                 * <=============================== *
(4)  |                                 |       SIM_READ_RECORD_REQ       |
     |                                 * ==============================> *
(5)  |                                 |       SIM_READ_RECORD_CNF       |
     |                                 * <=============================== *
(6)  |                                 |       SIM_READ_RECORD_REQ       |
     |                                 * ==============================> *
(7)  |                                 |       SIM_READ_RECORD_CNF       |
     |                                 * <=============================== *
(9)  |        MNSMS_MESSAGE_IND        |                                 |
     * <============================== *                                 |
(10) |        MNSMS_REPORT_IND         |                                 |
     * <=============================== *                                 |
COMMAND (SMS STATUS PARTITION)
     |                                 |                                 |
```

Parametrization

Primitive	Parameter	Value
(2) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
(3) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_EMPTY
(4) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
(5) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_2
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_EMPTY

(6) SIM_READ_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_3
length	LENGTH_SMS

(7) SIM_READ_RECORD_CNF

datafield	SIM_SMS
cause	SIM_NO_ERROR
record	SIM_RECORD_3
max_record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_EMPTY

(8) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SMS_RECORD_NOT_EXIST
rec_max	SIM_RECORD_3
status	SMS_RECORD_FREE
sms_sdu	SMS_SDU_EMPTY

(9) MNSMS_REPORT_IND

state	SMS_STATE_READY

History:	13-Oct-98	LE	Initial
	08-Mar-2000	FK	SIM_SMS_INSERT_IND extended
	21-Nov-2001	FK	Major rework
	14-Jun-2002	FK	Preambles for ME memory added
	25-Oct-2002	FK	Adaption to Cause Concept

3.2.8 SMS013: MO-SMS message available

Description: The SIM application sends the initial parameters read from the SIM card. SMS will find no memory for SMS in the mobile equipment and starts searching for SMS records on the SIM card. The SIM card memory has three records. Two records are unused and one record is a mobile originated message.

Preamble: SMS005A

```
   MMI                               SMS                               SIM/MM
    |                                 |                                  |
(3) |                                 |       SIM_READ_RECORD_REQ        |
    |                                 * ===============================> *
(4) |                                 |       SIM_READ_RECORD_CNF        |
    |                                 * <=============================== *
(5) |                                 |       SIM_READ_RECORD_REQ        |
    |                                 * ===============================> *
(6) |                                 |       SIM_READ_RECORD_CNF        |
    |                                 * <=============================== *
(7) |        MNSMS_MESSAGE_IND        |                                  |
    * <=============================== *                                 |
(7) |                                 |       SIM_READ_RECORD_REQ        |
    |                                 * ===============================> *
(8) |                                 |       SIM_READ_RECORD_CNF        |
    |                                 * <=============================== *
(9) |        MNSMS_MESSAGE_IND        |                                  |
    * <=============================== *                                 |
(10)|        MNSMS_REPORT_IND         |                                  |
    * <=============================== *                                 |
COMMAND (SMS STATUS PARTITION)
    |                                 |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
(2) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_EMPTY
(3) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
(4) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_2
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MO
(5) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_STO_UNSENT
	sms_sdu	SMS_SDU_MO
(6) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_3
	length	LENGTH_SMS
(7) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_3
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MO_ABS
(8) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_3
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_STO_SENT
	sms_sdu	SMS_SDU_MO_ABS
(9) MNSMS_REPORT_IND		
	state	SMS_STATE_READY

History:	13-Oct-98	LE	Initial
	08-Mar-2000	FK	SIM_SMS_INSERT_IND extended

31-May-2000	FK	Record 3 holds MO stored sent
21-Nov-2001	FK	Major rework
25-Oct-2002	FK	Adaption to Cause Concept

3.2.9 SMS014: MT-SMS message available

Description: The SIM application sends the initial parameters read from the SIM card. SMS will find no memory for SMS in the mobile equipment and starts searching for SMS records on the SIM card. The SIM card memory has three records. Two records are unused and one record is a mobile terminated message.

Preamble: SMS005A

```
   MMI                              SMS                             SIM/MM
     |                                |                                |
(2)  |                                |      SIM_READ_RECORD_REQ       |
     |                                * =============================> *
(3)  |                                |      SIM_READ_RECORD_CNF       |
     |                                * <============================= *
(4)  |                                |      SIM_READ_RECORD_REQ       |
     |                                * =============================> *
(5)  |                                |      SIM_READ_RECORD_CNF       |
     |                                * <============================= *
(6)  |       MNSMS_MESSAGE_IND        |                                |
     * <============================= *                                |
(6)  |                                |      SIM_READ_RECORD_REQ       |
     |                                * =============================> *
(7)  |                                |      SIM_READ_RECORD_CNF       |
     |                                * <============================= *
(8)  |       MNSMS_REPORT_IND         |                                |
     * <============================= *                                |
COMMAND (SMS STATUS PARTITION)
     |                                |                                |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
(2) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_EMPTY
(3) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
(4) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_2
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT

(5) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SIM_RECORD_2
rec_max	SIM_RECORD_3
status	SMS_RECORD_REC_UNREAD
sms_sdu	SMS_SDU_MT

(6) SIM_READ_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_3
length	LENGTH_SMS

(7) SIM_READ_RECORD_CNF

datafield	SIM_SMS
cause	SIM_NO_ERROR
record	SIM_RECORD_3
max_record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_EMPTY

(8) MNSMS_REPORT_IND

state	SMS_STATE_READY

History:	13-Oct-98	LE	Initial
	08-Mar-2000	FK	SIM_SMS_INSERT_IND extended
	26-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.2.10 SMS015: MO and MT-SMS message available

Description: The SIM application sends the initial parameters read from the SIM card. SMS will find no memory for SMS in the mobile equipment and starts searching for SMS records on the SIM card. The SIM card memory has three records. One record is unused, one record is a mobile originated message and one record is a mobile terminated message.

Preamble: SMS005A

```
   MMI                                SMS                               SIM/MM
     |                                  |                                  |
(1)  |                                  |      SIM_READ_RECORD_REQ         |
     |                                  * ===============================> *
(2)  |                                  |      SIM_READ_RECORD_CNF         |
     |                                  * <=============================== *
(3)  |       MNSMS_MESSAGE_IND          |                                  |
     * <=============================== *                                  |
(4)  |                                  |      SIM_READ_RECORD_REQ         |
     |                                  * ===============================> *
(5)  |                                  |      SIM_READ_RECORD_CNF         |
     |                                  * <=============================== *
(6)  |                                  |      SIM_READ_RECORD_REQ         |
     |                                  * ===============================> *
(7)  |                                  |      SIM_READ_RECORD_CNF         |
     |                                  * <=============================== *
(8)  |       MNSMS_MESSAGE_IND          |                                  |
     * <=============================== *                                  |
(8)  |        MNSMS_REPORT_IND          |                                  |
     * <=============================== *                                  |
COMMAND (SMS STATUS PARTITION)
     |                                  |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
(2) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MO
(3) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_1
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_STO_UNSENT
	sms_sdu	SMS_SDU_MO
(4) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
(5) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_2
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_EMPTY
(6) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_3
	length	LENGTH_SMS
(7) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_3
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT
(8) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_3
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_MT
(9) MNSMS_REPORT_IND		
	state	SMS_STATE_READY

History:	13-Oct-98	LE	Initial
	08-Mar-2000	FK	SIM_SMS_INSERT_IND extended

21-Nov-2001	FK	Major rework
25-Oct-2002	FK	Adaption to Cause Concept

3.2.11 SMS016: 1 MO and 2 MT-SMS messages available, SIM memory full

Description: The SIM application sends the initial parameters read from the SIM card. SMS has checked the ME memory and other SIM properties and starts searching for SMS records on the SIM card. The SIM card memory has three records. One record is a mobile originated message and two records are a mobile terminated message.

Variants: <A>...<D>

Preamble:

<A>	SMS005B
<B>	SMS005E
<C>	SMS005G
<D>	SMS010F

```
    MMI                                  SMS                                 SIM/MM
     |                                    |                                     |
(2)  |                                    |       SIM_READ_RECORD_REQ           |
     |                                    * ==================================> *
(3)  |                                    |       SIM_READ_RECORD_CNF           |
     |                                    * <================================== *
(4)  |         MNSMS_MESSAGE_IND          |                                     |
     * <================================= *                                     |
(4)  |                                    |       SIM_READ_RECORD_REQ           |
     |                                    * ==================================> *
(5)  |                                    |       SIM_READ_RECORD_CNF           |
     |                                    * <================================== *
(6)  |         MNSMS_MESSAGE_IND          |                                     |
     * <================================= *                                     |
(6)  |                                    |       SIM_READ_RECORD_REQ           |
     |                                    * ==================================> *
(7)  |                                    |       SIM_READ_RECORD_CNF           |
     |                                    * <================================== *
(8)  |         MNSMS_MESSAGE_IND          |                                     |
     * <================================= *                                     |
(9)  |          MNSMS_REPORT_IND          |                                     |
     * <================================= *                                     |
COMMAND (SMS STATUS PARTITION)
     |                                    |                                     |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
(2) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MO

(3) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SIM_RECORD_1
rec_max	SIM_RECORD_3
status	SMS_RECORD_STO_UNSENT
sms_sdu	SMS_SDU_MO

(4) SIM_READ_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_2
length	LENGTH_SMS

(5) SIM_READ_RECORD_CNF

datafield	SIM_SMS
cause	SIM_NO_ERROR
record	SIM_RECORD_2
max_record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_MT

(6) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SIM_RECORD_2
rec_max	SIM_RECORD_3
status	SMS_RECORD_REC_UNREAD
sms_sdu	SMS_SDU_MT

(7) SIM_READ_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_3
length	LENGTH_SMS

(8) SIM_READ_RECORD_CNF

datafield	SIM_SMS
cause	SIM_NO_ERROR
record	SIM_RECORD_3
max_record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_MT_READ

(9) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SIM_RECORD_3
rec_max	SIM_RECORD_3
status	SMS_RECORD_REC_READ
sms_sdu	SMS_SDU_MT

(10) MNSMS_REPORT_IND

state	SMS_STATE_READY

History:	13-Oct-98	LE	Initial
	08-Mar-2000	FK	SIM_SMS_INSERT_IND extended
	26-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.2.12 SMS017: 1 MO and 2 MT-SMS messages available, SIM memory full not set

Description: The SIM application sends the initial parameters read from the SIM card. SMS has checked the ME memory and other SIM properties and starts searching for SMS records on the SIM card. The SIM card memory has three records. One record is a mobile originated message and two records are a mobile terminated message. The Memory Capacity Exceeding Flag on the SIM is not set

Variants: <A>...<D>

Preamble:

<A>	SMS005A
<B>	SMS010A
<C>	SMS010C
<D>	SMS010E

```
   MMI                                SMS                               SIM/MM
     |                                 |                                 |
(1)  |                                 |       SIM_READ_RECORD_REQ       |
     |                                 * ===============================> *
(2)  |                                 |       SIM_READ_RECORD_CNF       |
     |                                 * <=============================== *
(3)  |        MNSMS_MESSAGE_IND        |                                 |
     * <=============================== *                                 |
(4)  |                                 |       SIM_READ_RECORD_REQ       |
     |                                 * ===============================> *
(5)  |                                 |       SIM_READ_RECORD_CNF       |
     |                                 * <=============================== *
(6)  |        MNSMS_MESSAGE_IND        |                                 |
     * <=============================== *                                 |
(7)  |                                 |       SIM_READ_RECORD_REQ       |
     |                                 * ===============================> *
(8)  |                                 |       SIM_READ_RECORD_CNF       |
     |                                 * <=============================== *
(9)  |        MNSMS_MESSAGE_IND        |                                 |
     * <=============================== *                                 |
(10) |         MNSMS_REPORT_IND        |                                 |
     * <=============================== *                                 |
COMMAND (SMS STATUS PARTITION)
     |                                 |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
(2) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MO
(3) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_1
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_STO_UNSENT
	sms_sdu	SMS_SDU_MO
(4) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS

(5) SIM_READ_RECORD_CNF

datafield	SIM_SMS
cause	SIM_NO_ERROR
record	SIM_RECORD_2
max_record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_MT

(6) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SIM_RECORD_2
rec_max	SIM_RECORD_3
status	SMS_RECORD_REC_UNREAD
sms_sdu	SMS_SDU_MT

(7) SIM_READ_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_3
length	LENGTH_SMS

(8) SIM_READ_RECORD_CNF

datafield	SIM_SMS
cause	SIM_NO_ERROR
record	SIM_RECORD_3
max_record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_MT

(9) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SIM_RECORD_3
rec_max	SIM_RECORD_3
status	SMS_RECORD_REC_UNREAD
sms_sdu	SMS_SDU_MT

(10) MNSMS_REPORT_IND

state	SMS_STATE_READY

History:	18-Jan-2000	FK	Initial
	08-Mar-2000	FK	SIM_SMS_INSERT_IND extended
	27-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.2.13 SMS018: MO and MT-SMS message available, Memory full is set

Description: The SIM application sends the initial parameters read from the SIM card, which indicate 'Memory Full'. SMS has checked the ME memory and other SIM properties and starts searching for SMS records on the SIM card. The SIM card memory has three records. One record is unused, one record is a mobile originated message and one record is a mobile terminated message.

Variants: <A>...<D>

Preamble:

<A>	SMS005B
<B>	SMS010B
<C>	SMS010D
<D>	SMS010F

```
   MMI                                       SMS                                   SIM/MM
     |                                        |                                        |
(1)  |                                        |        SIM_READ_RECORD_REQ             |
     |                                        * ==============================> *
```

```
(2)  |                                |       SIM_READ_RECORD_CNF        |
     |                                * <=============================== *
(3)  |       MNSMS_MESSAGE_IND        |                                  |
     * <============================= *                                  |
(4)  |                                |       SIM_READ_RECORD_REQ        |
     |                                * ===============================> *
(5)  |                                |       SIM_READ_RECORD_CNF        |
     |                                * <=============================== *
(6)  |                                |       SIM_READ_RECORD_REQ        |
     |                                * ===============================> *
(7)  |                                |       SIM_READ_RECORD_CNF        |
     |                                * <=============================== *
(8)  |       MNSMS_MESSAGE_IND        |                                  |
     * <============================= *                                  |
     |                                |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
(2) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MO
(3) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_1
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_STO_UNSENT
	sms_sdu	SMS_SDU_MO
(4) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
(5) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_2
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_EMPTY
(6) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_3
	length	LENGTH_SMS

(7) SIM_READ_RECORD_CNF

	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_3
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT

(8) MNSMS_MESSAGE_IND

	mem_type	MEM_SM
	rec_num	SIM_RECORD_3
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_MT

History:	18-Jan-2000	FK	Initial
	08-Mar-2000	FK	SIM_SMS_INSERT_IND extended
	27-Nov-2001	FK	Major rework
	17-Jun-2002	FK	Split with TC SMS480
	25-Oct-2002	FK	Adaption to Cause Concept

3.2.14 SMS019: SIM memory full, ME memory existent, MCEF set

Description: The SIM application sends the initial parameters read from the SIM card, which indicates that the Memory Capacity Exceeding Flag is set. SMS has checked the ME memory and other SIM properties and starts searching for SMS records on the SIM card. The SIM card memory has three records. One record is a mobile originated message and two records are a mobile terminated message. No MNSMS_REPORT_IND is sent.

Variants: <A>...<B>

Preamble:

<A> SMS010B
<B> SMS010D

```
   MMI                                 SMS                                SIM/MM
    |                                   |                                   |
(1) |                                   |       SIM_READ_RECORD_REQ         |
    |                                   * ===============================> *
(2) |                                   |       SIM_READ_RECORD_CNF         |
    |                                   * <=============================== *
(3) |        MNSMS_MESSAGE_IND          |                                   |
    * <=============================== *                                   |
(4) |                                   |       SIM_READ_RECORD_REQ         |
    |                                   * ===============================> *
(5) |                                   |       SIM_READ_RECORD_CNF         |
    |                                   * <=============================== *
(6) |        MNSMS_MESSAGE_IND          |                                   |
    * <=============================== *                                   |
(6) |                                   |       SIM_READ_RECORD_REQ         |
    |                                   * ===============================> *
(7) |                                   |       SIM_READ_RECORD_CNF         |
    |                                   * <=============================== *
(8) |        MNSMS_MESSAGE_IND          |                                   |
    * <=============================== *                                   |
    |                                   |                                   |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
(2) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MO
(3) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_1
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_STO_UNSENT
	sms_sdu	SMS_SDU_MO
(4) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
(5) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_2
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT
(6) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_MT
(7) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_3
	length	LENGTH_SMS
(8) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_3
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT_READ

Primitive	Parameter	Value
(9) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_3
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_READ
	sms_sdu	SMS_SDU_MT

History:	17-Jun-2002	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.2.15 SMS021: Unexpected Messages stored on SIM

Description: The SIM keeps 2 Short Messages whose status indicates one MO and one MT, but with TPDUs not being a SMS-Submit and a SMS-Deliver. The SMS Entity shall recognize these SMS records as free.

Preamble: SMS005A

```
   MMI                                   SMS                                SIM/MM
     |                                    |                                    |
(2)  |                                    |       SIM_READ_RECORD_REQ          |
     |                                    * =================================> *
(3)  |                                    |       SIM_READ_RECORD_CNF          |
     |                                    * <================================= *
(4)  |                                    |       SIM_READ_RECORD_REQ          |
     |                                    * =================================> *
(5)  |                                    |       SIM_READ_RECORD_CNF          |
     |                                    * <================================= *
(6)  |                                    |       SIM_READ_RECORD_REQ          |
     |                                    * =================================> *
(7)  |                                    |       SIM_READ_RECORD_CNF          |
     |                                    * <================================= *
(9)  |        MNSMS_MESSAGE_IND           |                                    |
     * <================================= *                                    |
(10) |         MNSMS_REPORT_IND           |                                    |
     * <================================= *                                    |
COMMAND (SMS STATUS PARTITION)
     |                                    |                                    |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
(2) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_ST_REPORT
(3) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS

(4) SIM_READ_RECORD_CNF

datafield	SIM_SMS
cause	SIM_NO_ERROR
record	SIM_RECORD_2
max_record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_EMPTY

(5) SIM_READ_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_3
length	LENGTH_SMS

(6) SIM_READ_RECORD_CNF

datafield	SIM_SMS
cause	SIM_NO_ERROR
record	SIM_RECORD_3
max_record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_COMMAND_SENT

(7) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SMS_RECORD_NOT_EXIST
rec_max	SIM_RECORD_3
status	SMS_RECORD_FREE
sms_sdu	SMS_SDU_EMPTY

(8) MNSMS_REPORT_IND

state	SMS_STATE_READY

History:	30-May-2000	FK	Initial
	27-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.2.16 SMS022: No SMS messages available, but SMS download possible

Description: The SIM application sends the initial parameters read from the SIM card. SMS will find no memory for SMS in the mobile equipment and starts searching for SMS records on the SIM card. The SIM card memory has three records. All records are unused. SIM indicates the possiblity of SMS Download. MMI configures the New Message Indication to MT1.

Preamble: SMS005C

```
   MMI                               SMS                          SIM/MM
    |                                 |                              |
(2) |                                 |      SIM_READ_RECORD_REQ     |
    |                                 * ============================> *
(3) |                                 |      SIM_READ_RECORD_CNF     |
    |                                 * <============================ *
(4) |                                 |      SIM_READ_RECORD_REQ     |
    |                                 * ============================> *
(5) |                                 |      SIM_READ_RECORD_CNF     |
    |                                 * <============================ *
(6) |                                 |      SIM_READ_RECORD_REQ     |
    |                                 * ============================> *
(7) |                                 |      SIM_READ_RECORD_CNF     |
    |                                 * <============================ *
(9) |        MNSMS_MESSAGE_IND        |                              |
    * <============================== *                              |
(10)|         MNSMS_REPORT_IND        |                              |
    * <============================== *                              |
MUTE(500)
```

```
(11) |       MNSMS_CONFIGURE_REQ       |                              |
     * ==============================> *                              |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                 |                              |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
(2) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_EMPTY
(3) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
(4) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_2
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_EMPTY
(5) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_3
	length	LENGTH_SMS
(6) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_3
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_EMPTY
(7) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SMS_RECORD_NOT_EXIST
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_FREE
	sms_sdu	SMS_SDU_EMPTY
(8) MNSMS_REPORT_IND		
	state	SMS_STATE_READY

```
(9)  MNSMS_CONFIGURE_REQ
                                        pref_mem_3          MEM_SM
                                        mt                  MT2
                                        ds                  DS0
                                        mhc                 SMS_MHC_DEF
```

History:			
	13-Oct-98	LE	Initial
	08-Mar-2000	FK	SIM_SMS_INSERT_IND extended
	30-Jan-2002	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.2.17 SMS031: Configuring the SMS Entity

Description: Preamble for other test cases. ACI sends the necessary parameters to configure SMS. The <mt> configuration parameter and the preamble are variable, all other parameters are default. Variant G provides a case with 1 record left available on SIM, variant H, I provide cases with no record left available.

Variants: <A>...<P>

Preamble:

<A>	SMS012A
<B>	SMS012A
<C>	SMS012A
<D>	SMS012A
<E>	SMS012A
<F>	SMS012A
<G>	SMS013
<H>	SMS014
<I>	SMS015
<J>	SMS016A
<K>	SMS017A
<L>	SMS012B
<M>	SMS016B
<N>	SMS012C
<O>	SMS016C
<P>	SMS017D

```
   MMI                          SMS                          SIM/MM
    |                            |                              |
(1) |      MNSMS_CONFIGURE_REQ   |                              |
    * ========================> *                              |
MUTE(500)
    |                            |                              |
```

Parametrization

Primitive		Parameter	Value
(1)	MNSMS_CONFIGURE_REQ		
		pref_mem_3	MEM_SM
	<A>	mt	MT_DEF
	<B>	mt	MT0
	<C>	mt	MT1
	<D>	mt	MT2
	<E>	mt	MT3
	<F>	mt	NOT_PRESENT_8BIT
	<G>	mt	MT2
	<H>	mt	MT2
	<I>	mt	MT2
	<J>	mt	MT2
	<K>	mt	MT2
	<L>	mt	MT2
	<M>	mt	MT2
	<N>	mt	MT2
	<O>	mt	MT2
	<P>	mt	MT2
		ds	DS0
		mhc	SMS_MHC_DEF

History:	24-Nov-99	FK	Initial
	27-Nov-2001	FK	Major rework
	18-Jun-2002	FK	Preambles with ME memory support added

3.2.18 SMS032: Configuring the SMS Entity (supporting STATUS REQUEST)

Description: Preamble for other test cases. ACI sends the necessary parameters to configure SMS. The <mt> configuration parameter is variable, parameter <ds> is set to DS1, all other parameters are default

Variants: <A>...<F>

Preamble: SMS012A

```
   MMI                                SMS                                   SIM/MM
     |                                  |                                      |
(1)  |       MNSMS_CONFIGURE_REQ        |                                      |
     * ==============================> *                                      |
MUTE(500)
     |                                  |                                      |
```

Parametrization

Primitive		Parameter	Value
(1)	MNSMS_CONFIGURE_REQ		
		pref_mem_3	MEM_SM
	<A>	mt	MT_DEF
	<B>	mt	MT0
	<C>	mt	MT1
	<D>	mt	MT2
	<E>	mt	MT3
	<F>	mt	NOT_PRESENT_8BIT
		ds	DS1
		mhc	SMS_MHC_PH2

History:	24-Nov-99	FK	Initial
	27-Nov-2001	FK	Major rework

3.2.19 SMS033: Configuring the SMS Entity (support MT Acknowledge Response)

Description: Preamble for other test cases. ACI sends the necessary parameters to configure SMS. The MT configuration parameter is variable, parameter mhc is set to SMS_MHC_PH2PLUS to notify support of AT command +CNMA, all other parameters are default except for variant (G), which also sets DS2

Variants: <A>...<G>

Preamble: SMS012A

```
   MMI                                  SMS                                  SIM/MM
     |                                   |                                      |
(1)  |       MNSMS_CONFIGURE_REQ         |                                      |
     * ==============================> *                                        |
MUTE(500)
     |                                   |                                      |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_CONFIGURE_REQ		
		pref_mem_3	MEM_SM
	<A>	mt	MT_DEF
	<B>	mt	MT0
	<C>	mt	MT1
	<D>	mt	MT2
	<E>	mt	MT3
	<F>	mt	NOT_PRESENT_8BIT
	<G>	mt	MT1
	<A>	ds	DS0
	<B>	ds	DS0
	<C>	ds	DS0
	<D>	ds	DS0
	<E>	ds	DS0
	<F>	ds	DS0
	<G>	ds	DS1
		mhc	SMS_MHC_PH2PLUS

History:	24-Nov-99	FK	Initial
	27-Nov-2001	FK	Major rework

3.2.20 SMS034: Configuring the SMS Entity (supporting ME Memory)

Description: Preamble for other test cases. ACI sends the necessary parameters to configure SMS. SMS supports ME memory, therefore <mem3> is set to MEM_ME in some variants. The <mt> configuration parameter is variable, all other parameters are default

Variants: <A>...<P>

Preamble:

<A>	SMS012D
<B>	SMS012E
<C>	SMS012F
<D>	SMS012D
<E>	SMS012E
<F>	SMS012F
<G>	SMS012D
<H>	SMS012E
<I>	SMS012F
<J>	SMS012D
<K>	SMS012E
<L>	SMS012F
<M>	SMS016D

<N>	SMS017B
<O>	SMS017C
<P>	SMS017D

```
   MMI                              SMS                              SIM/MM
    |                                |                                 |
(1) |       MNSMS_CONFIGURE_REQ      |                                 |
    * ==============================> *                                |
MUTE(500)
    |                                |                                 |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_CONFIGURE_REQ		
	<A>	pref_mem_3	MEM_SM
	<B>	pref_mem_3	MEM_SM
	<C>	pref_mem_3	MEM_SM
	<D>	pref_mem_3	MEM_ME
	<E>	pref_mem_3	MEM_ME
	<F>	pref_mem_3	MEM_ME
	<G>	pref_mem_3	MEM_SM
	<H>	pref_mem_3	MEM_SM
	<I>	pref_mem_3	MEM_SM
	<J>	pref_mem_3	MEM_ME
	<K>	pref_mem_3	MEM_ME
	<L>	pref_mem_3	MEM_ME
	<M>	pref_mem_3	MEM_ME
	<N>	pref_mem_3	MEM_ME
	<O>	pref_mem_3	MEM_ME
	<P>	pref_mem_3	MEM_ME
	<A>	mt	MT1
	<B>	mt	MT3
	<C>	mt	MT1
	<D>	mt	MT1
	<E>	mt	MT3
	<F>	mt	MT1
	<G>	mt	MT2
	<H>	mt	MT2
	<I>	mt	MT2
	<J>	mt	MT2
	<K>	mt	MT2
	<L>	mt	MT2
	<M>	mt	MT2
	<N>	mt	MT2
	<O>	mt	MT2
	<P>	mt	MT2
		ds	DS0
		mhc	SMS_MHC_PH2

History: 18-Jun-2002 FK Initial

3.3 Mobile Originated Short Message

3.3.1 SMS041: Initiation by MMI

Description: The user starts sending of a short message. Each short message contains a message reference. This number is used to identify this short message for later transactions. The message reference is stored on the SIM card, if it is a phase 2 SIM card. The RP DATA message is created and forwarded to the Control Protocol layer. The timer TR1M is started. The RP DATA message is stored and the establishment of the SMS-Connection is requested by MM.

Variants: <A>...<T>

Preamble:

<A>	SMS031C
<B>	SMS031C
<C>	SMS031C
<D>	SMS031C
<E>	SMS031C
<F>	SMS042A
<G>	SMS044
<H>	SMS047
<I>	SMS048
<J>	SMS049
<K>	SMS050
<L>	SMS051
<M>	SMS053
<N>	SMS062
<O>	SMS063
<P>	SMS064
<Q>	SMS065
<R>	SMS066
<S>	SMS067
<T>	SMS068

```
   MMI                               SMS                            SIM/MM
     |                                |                                |
(1)  |       MNSMS_SUBMIT_REQ         |                                |
     * =============================> *                                |
(2)  |                                |      MMSMS_ESTABLISH_REQ       |
     |                                * =============================> *
(3)  |                                |         SIM_UPDATE_REQ         |
     |                                * =============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_SUBMIT_REQ		
	<A>	mem_type	NOT_PRESENT_8BIT
	<B>	mem_type	BYTE_00
	<C>	mem_type	MEM_ME
	<D>	mem_type	MEM_SM
	<E>	mem_type	MEM_SR
	<F>	mem_type	NOT_PRESENT_8BIT
	<G>	mem_type	NOT_PRESENT_8BIT
	<H>	mem_type	NOT_PRESENT_8BIT
	<I>	mem_type	NOT_PRESENT_8BIT
	<J>	mem_type	NOT_PRESENT_8BIT
	<K>	mem_type	NOT_PRESENT_8BIT

	<L>	mem_type	NOT_PRESENT_8BIT
	<M>	mem_type	NOT_PRESENT_8BIT
	<N>	mem_type	NOT_PRESENT_8BIT
	<O>	mem_type	NOT_PRESENT_8BIT
	<P>	mem_type	NOT_PRESENT_8BIT
	<Q>	mem_type	NOT_PRESENT_8BIT
	<R>	mem_type	NOT_PRESENT_8BIT
	<S>	mem_type	NOT_PRESENT_8BIT
	<T>	mem_type	NOT_PRESENT_8BIT
		rec_num	SMS_RECORD_NOT_EXIST
	<A>	condx	SMS_CONDX_OVR_NON
	<B>	condx	SMS_CONDX_OVR_MO
	<C>	condx	SMS_CONDX_OVR_ANY
	<D>	condx	NOT_PRESENT_8BIT
	<E>	condx	BYTE_55
	<F>	condx	SMS_CONDX_OVR_NON
	<G>	condx	SMS_CONDX_OVR_NON
	<H>	condx	SMS_CONDX_OVR_NON
	<I>	condx	SMS_CONDX_OVR_NON
	<J>	condx	SMS_CONDX_OVR_NON
	<K>	condx	SMS_CONDX_OVR_NON
	<L>	condx	SMS_CONDX_OVR_NON
	<M>	condx	SMS_CONDX_OVR_NON
	<N>	condx	SMS_CONDX_OVR_NON
	<O>	condx	SMS_CONDX_OVR_NON
	<P>	condx	SMS_CONDX_OVR_NON
	<Q>	condx	SMS_CONDX_OVR_NON
	<R>	condx	SMS_CONDX_OVR_NON
	<S>	condx	SMS_CONDX_OVR_NON
	<T>	condx	SMS_CONDX_OVR_NON
	<A>	modify	SMS_MODIFY_NON
	<B>	modify	SMS_MODIFY_TPOA
	<C>	modify	SMS_MODIFY_SCA
	<D>	modify	SMS_MODIFY_TPOA_SCA
	<E>	modify	NOT_PRESENT_8BIT
	<F>	modify	SMS_MODIFY_NON
	<G>	modify	SMS_MODIFY_NON
	<H>	modify	SMS_MODIFY_NON
	<I>	modify	SMS_MODIFY_NON
	<J>	modify	SMS_MODIFY_NON
	<K>	modify	SMS_MODIFY_NON
	<L>	modify	SMS_MODIFY_NON
	<M>	modify	SMS_MODIFY_NON
	<N>	modify	SMS_MODIFY_NON
	<O>	modify	SMS_MODIFY_NON
	<P>	modify	SMS_MODIFY_NON
	<Q>	modify	SMS_MODIFY_NON
	<R>	modify	SMS_MODIFY_NON
	<S>	modify	SMS_MODIFY_NON
	<T>	modify	SMS_MODIFY_NON
		sms_sdu	SMS_SDU_SUBMIT_ABS
（2）	MMSMS_ESTABLISH_REQ		
		ti	TI_MO

(3) SIM_UPDATE_REQ		
	source	SRC_SMS
	offset	OFFSET_0
	datafield	SIM_SMSS
	length	SIMREC_SMSS_MSG_REF_LEN
<A>	trans_data	SIMREC_SMSS_MSG_REF
<B>	trans_data	SIMREC_SMSS_MSG_REF
<C>	trans_data	SIMREC_SMSS_MSG_REF
<D>	trans_data	SIMREC_SMSS_MSG_REF
<E>	trans_data	SIMREC_SMSS_MSG_REF
<F>	trans_data	SIMREC_SMSS_MSG_REF_N2
<G>	trans_data	SIMREC_SMSS_MSG_REF_N2
<H>	trans_data	SIMREC_SMSS_MSG_REF_N2
<I>	trans_data	SIMREC_SMSS_MSG_REF_N2
<J>	trans_data	SIMREC_SMSS_MSG_REF_N2
<K>	trans_data	SIMREC_SMSS_MSG_REF_N2
<L>	trans_data	SIMREC_SMSS_MSG_REF_N2
<M>	trans_data	SIMREC_SMSS_MSG_REF_N2
<N>	trans_data	SIMREC_SMSS_MSG_REF_N2
<O>	trans_data	SIMREC_SMSS_MSG_REF_N2
<P>	trans_data	SIMREC_SMSS_MSG_REF_N2
<Q>	trans_data	SIMREC_SMSS_MSG_REF_N2
<R>	trans_data	SIMREC_SMSS_MSG_REF_N2
<S>	trans_data	SIMREC_SMSS_MSG_REF_N2
<T>	trans_data	SIMREC_SMSS_MSG_REF_N2

History:	6-Jan-98	SZ	Initial
	13-Apr-2000	FK	Primitive change
	28-Nov-2001	FK	Major rework
	08-Feb-2002	FK	Additonal preambles: SUBMIT error cases

3.3.2 SMS042: No SMS Connection established

Description: MM indicates that the SMS-connection has not been established. The control protocol indicates the error to the relay layer. The error is reported to the user.

Variants: <A>...<E>

Preamble:

<A>	SMS041A
<B>	SMS041B
<C>	SMS041C
<D>	SMS041D
<E>	SMS041E

```
   MMI                               SMS                            SIM/MM
     |                                 |                                 |
(1)  |                                 |        MMSMS_RELEASE_IND        |
     |                                 * <============================== *
(2)  |        MNSMS_SUBMIT_CNF         |                                 |
     * <============================== *                                 |
MUTE(500)
(3)  |                                 |          SIM_UPDATE_CNF         |
     |                                 * <============================== *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                 |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_RELEASE_IND		
	ti	TI_MO
<A>	cause	RRCS_DL_EST_FAIL
<B>	cause	MMCS_NETWORK_FAILURE
<C>	cause	RRCS_MO_MT_COLL
<D>	cause	MMCS_SERVICE_ORDER
<E>	cause	RRCS_ABORT_RAD_LNK_FAIL
(2) MNSMS_SUBMIT_CNF		
<A>	mem_type	NOT_PRESENT_8BIT
<B>	mem_type	BYTE_00
<C>	mem_type	MEM_ME
<D>	mem_type	MEM_SM
<E>	mem_type	MEM_SR
	rec_num	SMS_RECORD_NOT_EXIST
<A>	cause	RRCS_DL_EST_FAIL
<B>	cause	MMCS_NETWORK_FAILURE
<C>	cause	RRCS_MO_MT_COLL
<D>	cause	MMCS_SERVICE_ORDER
<E>	cause	RRCS_ABORT_RAD_LNK_FAIL
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY
(3) SIM_UPDATE_CNF		
	datafield	SIM_SMSS
	cause	SIM_NO_ERROR

History:	6-Jan-98	SZ	Initial
	28-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.3 SMS043: Confirmation of SMS Connection

Description: MM confirms the establishment of the SMS connection. The control Protocol builds a CP_DATA message containing the stored RP_DATA message. The message is forwarded to MM. The timer TC1M is started to control the answer of the network. The retransmission counter is set to the default value.

Variants: <A>...<B>

Preamble:

<A> SMS041A

<B> SMS045A

```
   MMI                               SMS                             SIM/MM
     |                                |                                 |
MUTE(500)
(1)  |                                |         SIM_UPDATE_CNF          |
     |                                * <============================== *
MUTE(500)
(2)  |                                |       MMSMS_ESTABLISH_CNF       |
     |                                * <============================== *
(3)  |                                |         MMSMS_DATA_REQ          |
     |                                |            (CP_DATA)            |
     |                                * ==============================> *
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_UPDATE_CNF		
	datafield	SIM_SMSS
	cause	SIM_NO_ERROR
(2) MMSMS_ESTABLISH_CNF		
	ti	TI_MO
(3) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MO
	cp_user_data_ul	RP_DATA_SUBMIT_ABS
	}	

History:	6-Jan-98	SZ	Initial
	28-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.4 SMS044: Timeout TR1M

Description: In the relay layer the timer TR1M times-out. That means that no answer has received from the infrastructure. The control protocol is informed about the abort. The requested SMS connection is released. The user is informed about the abort of the mobile originated short message service.

Preamble: SMS041A

```
   MMI                               SMS                             SIM/MM
     |                                |                                 |
MUTE(2500)
(1)  |                                |         SIM_UPDATE_CNF          |
     |                                * <============================== *
MUTE(35000)
(2)  |       MNSMS_SUBMIT_CNF         |                                 |
    * <============================== *                                 |
(3)  |                                |       MMSMS_RELEASE_REQ         |
     |                                * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_UPDATE_CNF		
	datafield	SIM_SMSS
	cause	SIM_NO_ERROR
(2) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	SMS_CAUSE_NET_TIMEOUT
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY

Primitive	Parameter	Value
(3) MMSMS_RELEASE_REQ		
	ti	TI_MO

History:	6-Jan-98	SZ	Initial
	13-Apr-2000	FK	Primitive change
	7-Nov-2000	LW	Timeout
	28-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.5 SMS045: Multiple Request by MMI

Description: The user tries to send an additional mobile originated short message although the current transaction has not finished. Simultaneous transactions are not allowed. The user is informed about rejection of the mobile originated short message request.

Variants: <A>...<C>

Preamble:

<A>	SMS041A
<B>	SMS043A
<C>	SMS046A

```
   MMI                                 SMS                                  SIM/MM
     |                                   |                                    |
(1)  |         MNSMS_SUBMIT_REQ          |                                    |
     * ==============================> *                                    |
(2)  |         MNSMS_SUBMIT_CNF          |                                    |
     * <============================== *                                    |
MUTE(500)
     |                                   |                                    |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_SUBMIT_REQ		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	condx	SMS_CONDX_OVR_NON
	modify	SMS_MODIFY_NON
	sms_sdu	SMS_SDU_SUBMIT_ABS
(2) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	SMS_CAUSE_ENTITY_BUSY
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY

History:	6-Jan-98	SZ	Initial
	13-Apr-2000	FK	Primitive change
	12-Dec-2000	LW	Adaption to new TAP
	28-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.6 SMS046: Confirmation to CP-DATA Message

Description: Control Protocol is in the state 'Wait for CP-ACK'. That means it waits for the acknowledgement of the previous CP-DATA message. This acknowlegment receives. The timer TC1M is stopped and control protocol enters the state 'SMS CONNECTION ESTABLISHED'.

Variants: <A>...<C>

Preamble:

<A>	SMS043A

<B>	SMS043B
<C>	SMS045B

```
   MMI                                    SMS                                    SIM/MM
     |                                     |                                       |
(1)  |                                     |           MMSMS_DATA_IND              |
     |                                     |              (CP_ACK)                 |
     |                                     * <================================== *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                     |                                       |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MO_TO_MS
	}	

History:	6-Jan-98	SZ	Initial
	28-Nov-2001	FK	Major rework

3.3.7 SMS047: Wrong Message received (CP_DATA)

Description: Control Protocol expects a CP-ACK message as response to its own CP-DATA message. Instead of this a CP-DATA message receives. The timer TC1M is stopped. Control Protocol answers with a CP-ERROR message with cause #98. The Relay Layer and the user are informed about the error. The SMS-Connection is not longer necessary and will be released. MM is informed about the release of the SMS connection.

Preamble: SMS043A

```
   MMI                                    SMS                                    SIM/MM
     |                                     |                                       |
MUTE(1000)
(1)  |                                     |           MMSMS_DATA_IND              |
     |                                     |             (CP_DATA)                 |
     |                                     * <================================== *
(2)  |                                     |           MMSMS_DATA_REQ              |
     |                                     |             (CP_ERROR)                |
     |                                     * ==================================> *
(3)  |         MNSMS_SUBMIT_CNF            |                                       |
     * <================================== *                                       |
(4)  |                                     |          MMSMS_RELEASE_REQ            |
     |                                     * ==================================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                     |                                       |
```

Parametrization

	Primitive	Parameter	Value
(1)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	D_CP_DATA
		ti	TI_MO_TO_MS
		cp_user_data_dl	RP_DATA_DELIVER_7DEF
		}	
(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	B_CP_ERROR
		ti	TI_MO
		cp_cause	SMS_CP_CS_MSG_NOT_COMP
		}	
(3)	MNSMS_SUBMIT_CNF		
		mem_type	NOT_PRESENT_8BIT
		rec_num	SMS_RECORD_NOT_EXIST
		cause	SMS_TX_CS_MSG_NOT_COMP
		tp_mr	NOT_USED
		sms_sdu	SMS_SDU_EMPTY
(4)	MMSMS_RELEASE_REQ		
		ti	TI_MO

History:	6-Jan-98	SZ	Initial
	28-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.8 SMS048: Wrong Message received (unknown)

Description: Control Protocol expects a CP-ACK message as response to its own CP-DATA message. Instead of this a unknown message receives. The timer TC1M is stopped. Control Protocol answers with a CP-ERROR message with cause #97. The Relay Layer and the user are informed about the error. The SMS-Connection is not longer necessary and will be released. MM is informed about the release of the SMS connection.

Preamble: SMS043A

```
   MMI                                SMS                                 SIM/MM
    |                                  |                                    |
MUTE(2000)
(1) |                                  |          MMSMS_DATA_IND            |
    |                                  |            (unknown)               |
    |                                  * <================================ *
(2) |                                  |          MMSMS_DATA_REQ            |
    |                                  |            (CP_ERROR)              |
    |                                  * ================================> *
(3) |         MNSMS_SUBMIT_CNF         |                                    |
    * <============================== *                                    |
(4)  |                                |        MMSMS_RELEASE_REQ        |
     |                                * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	UNKN_SMS_MO_MSG
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ERROR
	ti	TI_MO
	cp_cause	SMS_CP_CS_INFO_NON_EXIST
	}	
(3) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	SMS_TX_CS_INFO_NON_EXIST
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY
(4) MMSMS_RELEASE_REQ		
	ti	TI_MO

History:	6-Jan-98	SZ	Initial
	6-Apr-2001	FK	Error code for ACI changed
	28-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.9 SMS049: CP Error received

Description: Control Protocol expects a CP-ACK message as response of its own previous CP-DATA message. Instead of that a CP-ERROR message receives. The timer TC1M is stopped. The error cause is forwarded to the relay layer. The user is informed about the error. The release of the SMS connection is requested. MM is informed about the release of the SMS connection.

Preamble: SMS043A

```
   MMI                               SMS                             SIM/MM
     |                                |                                 |
(1)  |                                |         MMSMS_DATA_IND          |
     |                                |            (CP_ERROR)           |
     |                                * <============================== *
(2)  |       MNSMS_SUBMIT_CNF         |                                 |
     * <============================= *                                 |
(3)  |                                |        MMSMS_RELEASE_REQ        |
     |                                * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ERROR
	ti	TI_MO_TO_MS
	cp_cause	SMS_CP_CS_NETWORK_FAILURE
	}	
(2) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	SMS_RX_CS_NETWORK_FAILURE
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY
(3) MMSMS_RELEASE_REQ		
	ti	TI_MO

History:			
	6-Jan-98	SZ	Initial
	6-Apr-2001	FK	Error code for ACI changed
	28-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.10 SMS050: Lower Layer Failure

Description: A lower layer failure is detected. The SMS connection is not longer valid. Relay Layer and the user are informed about the loss of the connection.

Preamble: SMS043A

```
   MMI                              SMS                              SIM/MM
    |                                |                                 |
(1) |                                |         MMSMS_ERROR_IND         |
    |                                * <=============================== *
(2) |        MNSMS_SUBMIT_CNF        |                                 |
    * <=============================== *                               |
(3) |                                |        MMSMS_RELEASE_REQ        |
    |                                * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ERROR_IND		
	ti	TI_MO
	cause	RRCS_DATA_LINK_FAIL
(2) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	RRCS_DATA_LINK_FAIL
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY

(3) MMSMS_RELEASE_REQ		
	ti	TI_MO

History:	6-Jan-98	SZ	Initial
	28-Nov-2001	FK	Major rework
	07-Jan-2002	FK	MMSMS_RELEASE_REQ added
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.11 SMS051: Release of MM-Connection

Description: The lower layer have released the connection. Control Protocol is informed. This information is forwarded to relay layer. The user is informed about the error.

Preamble: SMS043A

```
   MMI                                      SMS                                   SIM/MM
    |                                        |                                       |
(1) |                                        |          MMSMS_RELEASE_IND            |
    |                                        * <==================================== *
(2) |          MNSMS_SUBMIT_CNF              |                                       |
    * <================================== *                                          |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                        |                                       |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_RELEASE_IND		
	ti	TI_MO
	cause	RRCS_ABNORM_UNSPEC
(2) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	RRCS_ABNORM_UNSPEC
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY

History:	6-Jan-98	SZ	Initial
	28-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.12 SMS052: Timeout TC1M (first time)

Description: The timer TC1M of the control protocol times-out the first time. So the CP-DATA message is retransmitted.

Preamble: SMS043A

```
   MMI                                      SMS                                   SIM/MM
    |                                        |                                       |
MUTE(12500)
(1) |                                        |            MMSMS_DATA_REQ             |
    |                                        |               (CP_DATA)               |
    |                                        * ====================================> *
    |                                        |                                       |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MO
	cp_user_data_ul	RP_DATA_SUBMIT_ABS
		}

History:	6-Jan-98	SZ	Initial
	13-Apr-2000	FK	Primitive change
	12-Dec-2000	LW	Adaption to new tap
	28-Nov-2001	FK	Major rework

3.3.13 SMS053: Timeout TC1M (max retransmissions)

Description: The timeout occurs a second time. So the maximum of retransmissions is reached and the error is forwarded to the Relay Layer. The user is informed about the error. The SMS connection is not longer necessary. Relay Layer requests the release of SMS connection. Control Protocol releases the SMS connection.

Preamble: SMS052

```
   MMI                               SMS                             SIM/MM
     |                                |                                 |
MUTE(12500)
(1)  |       MNSMS_SUBMIT_CNF         |                                 |
     * <============================= *                                 |
(2)  |                                |        MMSMS_RELEASE_REQ        |
     |                                * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	SMS_CAUSE_NET_TIMEOUT
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY
(2) MMSMS_RELEASE_REQ		
	ti	TI_MO

History:	6-Jan-98	SZ	Initial
	28-Nov-2001	FK	Major rework

3.3.14 SMS061: Acknowledgment of the Network (SUBMIT successful)

Description: A SMS connection is established and the response of the network receives. It is a CP-DATA message containing a RP-ACK message. The RP-ACK message is forwarded to the Relay Layer. The user is informed about the positive end of procedure. After reception of the CP-DATA message Control Protocol sends a CP-ACK message as response to the infrastructure. Relay Layer releases the SMS connection. The release of SMS connection is requested to MM.

Variants: <A>...<C>

Preamble:

<A>	SMS046A
<B>	SMS046B
<C>	SMS045C

```
   MMI                               SMS                              SIM/MM
     |                                 |                                 |
MUTE(1000)
(1)  |                                 |         MMSMS_DATA_IND          |
     |                                 |            (CP_DATA)            |
     |                                 * <============================== *
(2)  |                                 |         MMSMS_DATA_REQ          |
     |                                 |            (CP_ACK)             |
     |                                 * ==============================> *
(3)  |        MNSMS_SUBMIT_CNF         |                                 |
    * <=============================== *                                 |
(4)  |                                 |        MMSMS_RELEASE_REQ        |
     |                                 * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                 |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_ACK_DLNK
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MO
	}	
(3) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	SMS_NO_ERROR
	tp_mr	TP_MR_3N1
	sms_sdu	SMS_SDU_EMPTY
(4) MMSMS_RELEASE_REQ		
	ti	TI_MO

History:			
	6-Jan-98	SZ	Initial
	28-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.15 SMS062: Error signalled by the Network

Description: A SMS connection is established and the response of the network receives. It is a CP-DATA message containing a RP-ERROR message. The RP-ERROR message is forwarded to the relay layer. The user is informed about the negative end of procedure. Relay Layer releases the SMS connection. The release of SMS connection is requested to MM.

Preamble: SMS046A

```
   MMI                              SMS                             SIM/MM
     |                                |                                |
(1)  |                                |        MMSMS_DATA_IND          |
     |                                |           (CP_DATA)            |
     |                                * <============================= *
(2)  |                                |        MMSMS_DATA_REQ          |
     |                                |            (CP_ACK)            |
     |                                * =============================> *
(3)  |       MNSMS_SUBMIT_CNF         |                                |
     * <============================= *                                |
(4)  |                                |       MMSMS_RELEASE_REQ        |
     |                                * =============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_ERR_CONGESTION
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MO
	}	
(3) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	SMS_RX_CS_CONGESTION
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY
(4) MMSMS_RELEASE_REQ		
	ti	TI_MO

History:	6-Jan-98	SZ	Initial
	28-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.16 SMS063: Wrong Message signalled by the Network (CP layer, CP-DATA)

Description: A SMS connection is established and the response of the network receives. It is a CP-ACK. The error is forwarded to the relay layer. The user is informed about the negative end of procedure. Control protocol sends a CP-ERROR message as response to the infrastructure. The used cause is #98. Relay Layer releases the SMS connection. The release of SMS connection is requested to MM.

Preamble: SMS046A

```
   MMI                                SMS                               SIM/MM
     |                                 |                                  |
(1)  |                                 |          MMSMS_DATA_IND          |
     |                                 |             (CP_ACK)             |
     |                                 * <=============================== *
(2)  |                                 |          MMSMS_DATA_REQ          |
     |                                 |            (CP_ERROR)            |
     |                                 * ===============================> *
(3)  |         MNSMS_SUBMIT_CNF        |                                  |
     * <============================== *                                  |
(4)  |                                 |        MMSMS_RELEASE_REQ         |
     |                                 * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                 |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MO_TO_MS
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ERROR
	ti	TI_MO
	cp_cause	SMS_CP_CS_MSG_NOT_COMP
	}	
(3) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	SMS_TX_CS_MSG_NOT_COMP
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY

(4) MMSMS_RELEASE_REQ		
	ti	TI_MO

History:	6-Jan-98	SZ	Initial
	28-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.17 SMS064: Wrong Message signalled by the Infrastructure (CP layer, unknown)

Description: A SMS connection is established and the response of the network receives. It is an unknown message. The error is forwarded to the relay layer. The user is informed about the negative end of procedure. Control Protocol sends a CP-ERROR message as response to the infrastructure. The used cause is #97. Relay Layer releases the SMS connection. The release of SMS connection is requested to MM.

Preamble: SMS046A

```
   MMI                               SMS                            SIM/MM
     |                                |                                |
(1)  |                                |         MMSMS_DATA_IND         |
     |                                |            (unknown)           |
     |                                * <============================= *
(2)  |                                |         MMSMS_DATA_REQ         |
     |                                |           (CP_ERROR)           |
     |                                * =============================> *
(3)  |       MNSMS_SUBMIT_CNF         |                                |
     * <============================= *                                |
(4)  |                                |       MMSMS_RELEASE_REQ        |
     |                                * =============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	UNKN_SMS_MO_MSG
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ERROR
	ti	TI_MO
	cp_cause	SMS_CP_CS_INFO_NON_EXIST
	}	
(3) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	SMS_TX_CS_INFO_NON_EXIST
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY
(4) MMSMS_RELEASE_REQ		
	ti	TI_MO

History:	6-Jan-98	SZ	Initial
	28-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.18 SMS065: Wrong Message signalled by the Infrastructure (RL layer)

Description: A response of the infrastructure is expected. Control Protocol receives a CP-DATA message. This CP-DATA message contains a RP-DATA message instead of the expected RP-ACK message. Relay Layer builds an RP-ERROR message and forwards it to the control procotol. The RP-ERROR message is included into a CP-DATA message and sent to the infrastructure. The error is reported to the user. Relay Layer requests the release of the SMS connection. The SMS connection is released.

Preamble: SMS046A

```
   MMI                               SMS                             SIM/MM
     |                                |                                |
(1)  |                                |         MMSMS_DATA_IND         |
     |                                |            (CP_DATA)           |
     |                                * <============================= *
(2)  |                                |         MMSMS_DATA_REQ         |
     |                                |            (CP_ACK)            |
     |                                * =============================> *
(3)  |                                |         MMSMS_DATA_REQ         |
     |                                |            (CP_DATA)           |
     |                                * =============================> *
(4)  |       MNSMS_SUBMIT_CNF         |                                |
     * <============================= *                                |
(5)  |                                |         MMSMS_DATA_IND         |
     |                                |            (CP_ACK)            |
     |                                * <============================= *
(6)  |                                |       MMSMS_RELEASE_REQ        |
     |                                * =============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_DATA_DELIVER_7DEF
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MO
	}	

(3) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MO
cp_user_data_ul	RP_ERR_PROTOCOL
}	

(4) MNSMS_SUBMIT_CNF

mem_type	NOT_PRESENT_8BIT
rec_num	SMS_RECORD_NOT_EXIST
cause	SMS_TX_CS_PROTOCOL_ERROR
tp_mr	NOT_USED
sms_sdu	SMS_SDU_EMPTY

(5) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MO_TO_MS
}	

(6) MMSMS_RELEASE_REQ

ti	TI_MO

History:	6-Jan-98	SZ	Initial
	28-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.19 SMS066: Timeout TR1M while waiting for RP-ACK

Description: A SMS connection is established. A CP-ACK arrives, therefore the timer TC1M is reset. Then the timer TR1M of the relay layer expires. The control protocol is informed about the abort. Control Protocol sends a CP-ERROR message with the cause #111 to the infrastructure. The relay layer requests the release of the SMS connection. The SMS connection is released.

Preamble: SMS046A

```
   MMI                               SMS                             SIM/MM
    |                                 |                                 |
MUTE(37500)
(1) |        MNSMS_SUBMIT_CNF         |                                 |
    * <============================== *                                 |
(2) |                                 |         MMSMS_DATA_REQ          |
    |                                 |           (CP_ERROR)            |
    |                                 * ==============================> *
(3) |                                 |        MMSMS_RELEASE_REQ        |
    |                                 * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                 |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	SMS_CAUSE_NET_TIMEOUT
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ERROR
	ti	TI_MO
	cp_cause	SMS_CP_CS_PROTOCOL_ERROR
	}	
(3) MMSMS_RELEASE_REQ		
	ti	TI_MO

History:			
History:	6-Jan-98	SZ	Initial
	13-Apr-2000	FK	Primitive change
	7-Nov-2000	LW	Timeout, CP-ACK stops TC1M
	12-Dec-2000	LW	Adaption to new TAP
	30-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.20 SMS067: Lower Layer Failure

Description: SMS expects a response of the infrastructure. Instead of that a lower layer failure is indicated. The release indication is forwarded to the relay layer. The user is informed about the negative end of procedure.

Preamble: SMS046A

```
   MMI                                SMS                              SIM/MM
     |                                 |                                  |
MUTE(1000)
(1)  |                                 |         MMSMS_ERROR_IND          |
     |                                 * <=============================== *
(2)  |        MNSMS_SUBMIT_CNF         |                                  |
     * <============================== *                                  |
(3)  |                                 |        MMSMS_RELEASE_REQ         |
     |                                 * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                 |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ERROR_IND		
	ti	TI_MO
	cause	RRCS_DATA_LINK_FAIL

(2) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	RRCS_DATA_LINK_FAIL
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY
(3) MMSMS_RELEASE_REQ		
	ti	TI_MO

History:	6-Jan-98	SZ	Initial
	30-Nov-2001	FK	Major rework
	07-Jan-2002	FK	MMSMS_RELEASE_REQ added
	25-Oct-2002	FK	Adaption to Cause Concept

3.3.21 SMS068: Release of MM-Connection

Description: SMS expects the response of the infrastructure. Instead of that the lower layer indicates the release of connection. The release indication is forwarded to the relay layer. The user is informed about the negative end of procedure.

Preamble: SMS046A

```
   MMI                               SMS                             SIM/MM
    |                                 |                                |
(1) |                                 |       MMSMS_RELEASE_IND        |
    |                                 * <============================= *
(2) |        MNSMS_SUBMIT_CNF         |                                |
    * <============================== *                                |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                 |                                |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_RELEASE_IND		
	ti	TI_MO
	cause	RRCS_ABNORM_UNSPEC
(2) MNSMS_SUBMIT_CNF		
	mem_type	NOT_PRESENT_8BIT
	rec_num	SMS_RECORD_NOT_EXIST
	cause	RRCS_ABNORM_UNSPEC
	tp_mr	NOT_USED
	sms_sdu	SMS_SDU_EMPTY

History:	6-Jan-98	SZ	Initial
	30-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.4 Interworking MO-SM with MT-SM

3.4.1 SMS071: Sending of MO-SM parallel to Reception of Class 1 Message (stored in SIM-Card)

Description: The user starts sending of a short message. The RP DATA message is created and forwarded to the Control Protocol layer. The timer TR1M is started. The RP DATA message is stored and the establishment of the SMS-Connection is requested by MM. Then the Relay Layer receives a mobile terminated short message. The parameter data-coding scheme of the message indicates that it is a class 1 message. These messages are not displayed directly to the user. They are stored in the mobile station memory if available, else on the SIM card. There is no mobile station memory, so it is stored on the SIM card.

Preamble: SMS041A

```
   MMI                               SMS                             SIM/MM
     |                                |                                 |
MUTE(500)
(1)  |                                |      MMSMS_ESTABLISH_IND        |
     |                                |          (CP_DATA: MT)          |
     |                                * <============================== *
(2)  |                                |         MMSMS_DATA_REQ          |
     |                                |          (CP_ACK: MT)           |
     |                                * ==============================> *
(3)  |                                |     SIM_UPDATE_RECORD_REQ       |
     |                                * ==============================> *
MUTE(500)
(4)  |                                |         SIM_UPDATE_CNF          |
     |                                * <============================== *
MUTE(500)
(1)  |                                |      MMSMS_ESTABLISH_CNF        |
     |                                * <============================== *
(2)  |                                |         MMSMS_DATA_REQ          |
     |                                |          (CP_DATA: MO)          |
     |                                * ==============================> *
(4)  |                                |     SIM_UPDATE_RECORD_CNF       |
     |                                * <============================== *
(5)  |       MNSMS_MESSAGE_IND        |                                 |
     * <============================= *                                 |
(6)  |                                |         MMSMS_DATA_REQ          |
     |                                |          (CP_DATA: MT)          |
     |                                * ==============================> *
MUTE(500)
(1)  |                                |         MMSMS_DATA_IND          |
     |                                |          (CP_ACK: MO)           |
     |                                * <============================== *
MUTE(500)
(7)  |                                |         MMSMS_DATA_IND          |
     |                                |          (CP_ACK: MT)           |
     |                                * <============================== *
(8)  |                                |       MMSMS_RELEASE_REQ         |
     |                                * ==============================> *
MUTE(500)
(1)  |                                |         MMSMS_DATA_IND          |
     |                                |          (CP_DATA: MO)          |
     |                                * <============================== *
(2)  |                                |         MMSMS_DATA_REQ          |
     |                                |          (CP_ACK: MO)           |
     |                                * ==============================> *
(3)  |       MNSMS_SUBMIT_CNF         |                                 |
     * <============================= *                                 |
(4)  |                          |        MMSMS_RELEASE_REQ        |
     |                          * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                          |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL1
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_1
(4) SIM_UPDATE_CNF		
	datafield	SIM_SMSS
	cause	SIM_NO_ERROR
(5) MMSMS_ESTABLISH_CNF		
	ti	TI_MO
(6) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MO
	cp_user_data_ul	RP_DATA_SUBMIT_ABS
	}	
(7) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_NO_ERROR

(8) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SIM_RECORD_1
rec_max	SIM_RECORD_3
status	SMS_RECORD_REC_UNREAD
sms_sdu	SMS_SDU_DELIVER_7CL1

(9) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_FROM_MS
cp_user_data_ul	RP_ACK_ULNK
}	

(10) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MO_TO_MS
}	

(11) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(12) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

(13) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	D_CP_DATA
ti	TI_MO_TO_MS
cp_user_data_dl	RP_ACK_DLNK
}	

(14) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MO
}	

(15) MNSMS_SUBMIT_CNF

mem_type	NOT_PRESENT_8BIT
rec_num	SMS_RECORD_NOT_EXIST
cause	SMS_NO_ERROR
tp_mr	TP_MR_3N1
sms_sdu	SMS_SDU_EMPTY

(16) MMSMS_RELEASE_REQ

ti	TI_MO

History:	25-Nov-99	FK	Initial
	30-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.4.2 SMS081: Reception of Class 1 Message (stored in SIM-Card) parallel to Writing of MO-SM to SIM

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 1 message. This messages are not displayed directly to the user. They are stored in the mobile station memory if available, else on the SIM card. There is no mobile station memory, so it is stored on the SIM card. During waiting for response from the SIM Entity the higher layer initiates a Storing Request which has to be succesfull as well.

Preamble: SMS031C

```
   MMI                               SMS                              SIM/MM
     |                                 |                                 |
MUTE(500)
(1)  |                                 |      MMSMS_ESTABLISH_IND        |
     |                                 |            (CP_DATA)            |
     |                                 * <============================== *
(2)  |                                 |         MMSMS_DATA_REQ          |
     |                                 |            (CP_ACK)             |
     |                                 * ==============================> *
(3)  |                                 |     SIM_UPDATE_RECORD_REQ       |
     |                                 * ==============================> *
MUTE(500)
(1)  |        MNSMS_STORE_REQ          |                                 |
     * ==============================> *                                 |
(2)  |                                 |     SIM_UPDATE_RECORD_REQ       |
     |                                 * ==============================> *
MUTE(500)
(4)  |                                 |     SIM_UPDATE_RECORD_CNF       |
     |                                 * <============================== *
(5)  |       MNSMS_MESSAGE_IND         |                                 |
     * <============================== *                                 |
(6)  |                                 |         MMSMS_DATA_REQ          |
     |                                 |            (CP_DATA)            |
     |                                 * ==============================> *
(3)  |                                 |     SIM_UPDATE_RECORD_CNF       |
     |                                 * <============================== *
```

```
(4) |         MNSMS_STORE_CNF        |                                 |
    * <============================= *                                 |
(7) |                                |         MMSMS_DATA_IND          |
    |                                |             (CP_ACK)            |
    |                                * <============================== *
(8) |                                |        MMSMS_RELEASE_REQ        |
    |                                * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL1
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_1
(4) MNSMS_STORE_REQ		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_0
	condx	SMS_CONDX_OVR_NON
	status	SMS_RECORD_STO_UNSENT
	sms_sdu	SMS_SDU_MO
(5) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
	linear_data	SIM_SMS_MO
(6) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_NO_ERROR

(7) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SIM_RECORD_1
rec_max	SIM_RECORD_3
status	SMS_RECORD_REC_UNREAD
sms_sdu	SMS_SDU_DELIVER_7CL1

(8) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_FROM_MS
cp_user_data_ul	RP_ACK_ULNK
}	

(9) SIM_UPDATE_RECORD_CNF

datafield	SIM_SMS
record	SIM_RECORD_2
cause	SIM_NO_ERROR

(10) MNSMS_STORE_CNF

mem_type	MEM_SM
rec_num	SIM_RECORD_2
cause	SIM_NO_ERROR

(11) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(12) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	30-Nov-99	FK	Initial
	13-Apr-2000	FK	Primitive change
	30-Nov-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.4.3 SMS082: Reception of a Replaceable Message parallel to Writing of MO-SM to SIM (Case 1)

Description: The Relay Layer receives a mobile terminated short message. The protocol identifier indicates a replace short message, which is to replace the previous stored message of record 1. While waiting for the SMS record to be read the higher layer initiates a Storing Request which has to be succesfull as well.

Preamble: SMS200

Variants: <A>...<C>

```
   MMI                                SMS                           SIM/MM
     |                                 |                                 |
(1)  |                                 |       MMSMS_ESTABLISH_IND       |
     |                                 |           (CP_DATA)             |
     |                                 * <============================== *
```

```
(2)  |                                |         MMSMS_DATA_REQ          |
     |                                |            (CP_ACK)             |
     |                                * ==============================> *
(3)  |                                |       SIM_READ_RECORD_REQ       |
     |                                * ==============================> *
MUTE(500)
(1)  |        MNSMS_STORE_REQ         |                                 |
     * =============================> *                                 |
(2)  |                                |      SIM_UPDATE_RECORD_REQ      |
     |                                * ==============================> *
MUTE(500)
(4)  |                                |       SIM_READ_RECORD_CNF       |
     |                                * <============================== *
(5)  |                                |      SIM_UPDATE_RECORD_REQ      |
     |                                * ==============================> *
MUTE(500)
(3)  |                                |      SIM_UPDATE_RECORD_CNF      |
     |                                * <============================== *
(4)  |        MNSMS_STORE_CNF         |                                 |
     * <============================= *                                 |
MUTE(500)
(6)  |                                |      SIM_UPDATE_RECORD_CNF      |
     |                                * <============================== *
(7)  |       MNSMS_MESSAGE_IND        |                                 |
     * <============================= *                                 |
(8)  |                                |         MMSMS_DATA_REQ          |
     |                                |            (CP_DATA)            |
     |                                * ==============================> *
MUTE(500)
(9)  |                                |         MMSMS_DATA_IND          |
     |                                |            (CP_ACK)             |
     |                                * <============================== *
(10) |                                |        MMSMS_RELEASE_REQ        |
     |                                * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
<A>	cp_user_data_dl	RP_DATA_DELIVER_7CL0_43
<B>	cp_user_data_dl	RP_DATA_DELIVER_7CL1_43
<C>	cp_user_data_dl	RP_DATA_DELIVER_7CL3_43
	}	

(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
(4) MNSMS_STORE_REQ		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_0
	condx	SMS_CONDX_OVR_NON
	status	SMS_RECORD_STO_UNSENT
	sms_sdu	SMS_SDU_MO
(5) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
	linear_data	SIM_SMS_MO
(6) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_2_43
(7) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
<A>	linear_data	SIM_SMS_CLASS_0_43
<B>	linear_data	SIM_SMS_CLASS_1_43
<C>	linear_data	SIM_SMS_CLASS_3_43
(8) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_2
	cause	SIM_NO_ERROR
(9) MNSMS_STORE_CNF		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	cause	SIM_NO_ERROR
(10) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_NO_ERROR

(11) MNSMS_MESSAGE_IND

	mem_type	MEM_SM
	rec_num	SIM_RECORD_1
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
<A>	sms_sdu	SMS_SDU_DELIVER_7CL0_43
<B>	sms_sdu	SMS_SDU_DELIVER_7CL1_43
<C>	sms_sdu	SMS_SDU_DELIVER_7CL3_43

(12) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_FROM_MS
cp_user_data_ul	RP_ACK_ULNK
}	

(13) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(14) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	5-May-2000	FK	Initial
	20-Dec-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.4.4 SMS083: Reception of a Replaceable Message parallel to Writing of MO-SM to SIM (Case 2)

Description: The Relay Layer receives a mobile terminated short message. The protocol identifier indicates a replace short message, which is to replace the previous stored message of record 1. While waiting for the SMS record to be updated the higher layer initiates a Storing Request which has to be succesfull as well.

Variants: <A>...<C>

Preamble: SMS200

```
   MMI                                SMS                           SIM/MM
    |                                  |                               |
(1) |                                  |      MMSMS_ESTABLISH_IND      |
    |                                  |           (CP_DATA)           |
    |                                  * <============================ *
(2) |                                  |         MMSMS_DATA_REQ        |
    |                                  |           (CP_ACK)            |
    |                                  * ============================> *
(3) |                                  |      SIM_READ_RECORD_REQ      |
    |                                  * ============================> *
MUTE(500)
(4) |                                  |      SIM_READ_RECORD_CNF      |
    |                                  * <============================ *
```

```
(5)  |                               |      SIM_UPDATE_RECORD_REQ      |
     |                               * ==============================> *
MUTE(500)
(1)  |        MNSMS_STORE_REQ        |                                 |
     * ============================> *                                 |
(2)  |                               |      SIM_UPDATE_RECORD_REQ      |
     |                               * ==============================> *
MUTE(500)
(6)  |                               |      SIM_UPDATE_RECORD_CNF      |
     |                               * <============================== *
(7)  |       MNSMS_MESSAGE_IND       |                                 |
     * <============================ *                                 |
(8)  |                               |         MMSMS_DATA_REQ          |
     |                               |            (CP_DATA)            |
     |                               * ==============================> *
MUTE(500)
 (3) |                               |      SIM_UPDATE_RECORD_CNF      |
     |                               * <============================== *
 (4) |        MNSMS_STORE_CNF        |                                 |
     * <============================ *                                 |
(9)  |                               |         MMSMS_DATA_IND          |
     |                               |            (CP_ACK)             |
     |                               * <============================== *
(10) |                               |        MMSMS_RELEASE_REQ        |
     |                               * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                               |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
<A>	cp_user_data_dl	RP_DATA_DELIVER_7CL0_43
<B>	cp_user_data_dl	RP_DATA_DELIVER_7CL1_43
<C>	cp_user_data_dl	RP_DATA_DELIVER_7CL3_43
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS

(4) SIM_READ_RECORD_CNF

	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_2_43

(5) SIM_UPDATE_RECORD_REQ

	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
<A>	linear_data	SIM_SMS_CLASS_0_43
<B>	linear_data	SIM_SMS_CLASS_1_43
<C>	linear_data	SIM_SMS_CLASS_3_43

(6) MNSMS_STORE_REQ

	mem_type	MEM_SM
	rec_num	SIM_RECORD_0
	condx	SMS_CONDX_OVR_NON
	status	SMS_RECORD_STO_UNSENT
	sms_sdu	SMS_SDU_MO

(7) SIM_UPDATE_RECORD_REQ

	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
	linear_data	SIM_SMS_MO

(8) SIM_UPDATE_RECORD_CNF

	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_NO_ERROR

(9) MNSMS_MESSAGE_IND

	mem_type	MEM_SM
	rec_num	SIM_RECORD_1
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
<A>	sms_sdu	SMS_SDU_DELIVER_7CL0_43
<B>	sms_sdu	SMS_SDU_DELIVER_7CL1_43
<C>	sms_sdu	SMS_SDU_DELIVER_7CL3_43

(10) MMSMS_DATA_REQ

	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ACK_ULNK
	}	

(11) SIM_UPDATE_RECORD_CNF

	datafield	SIM_SMS
	record	SIM_RECORD_2
	cause	SIM_NO_ERROR

(12) MNSMS_STORE_CNF

mem_type	MEM_SM
rec_num	SIM_RECORD_2
cause	SIM_NO_ERROR

(13) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(14) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	5-May-2000	FK	Initial
	20-Dec-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.5 Mobile Originated Short Message Command

3.5.1 SMS091: Mobile Originated Short Message Command

Description: The mobile originated short message command procedure is used to send commands to the service center for previous sent short message. The procedure differs from a mobile originated short message service procedure only in the different initial primitive. The mobile station starts sending of a short message command. The following commands are available: 'Status request for a short message', 'Delete of status report request for a short message' and 'Delete of a short message'. The named short message is identified by the message reference used for the short message. Each short message command has its own message reference. This number is used if other short message commands are related to this short message command. The message reference is incremented by one. If a phase 2 SIM is available the message reference is stored on the SIM card. The relay layer builds a RP-DATA message containing the short message command. The message is forwarded to the control protocol. The timer TR1M is started to supervise the response of the infrastructure. Control Protocol requests establishment of the SMS connection by MM. MM confirms establishment of the SMS connection. The CP-DATA message containing the RP-DATA message of the relay layer is send to the infrastructure. The timer TC1M is started to supervise response of the infrastructure. Control Protocol receives the response of the infrastructure. It is a CP-ACK message. It now waits for the response for the relay layer. A CP-DATA message receives. The content for the relay layer is decoded. The content of the message is a RP-ACK message. The user is informed about the positive end of procedure. The connection release is requested. The SMS connection is released.

Variant A: request status report

Variant B: enquiry on previously submitted short message

Variant C: cancel status report request

Variant D: delete previously submitted status report request

Variants: <A>...<D>

Preamble:

<A> SMS021
<B> SMS021
<C> SMS021
<D> SMS061A

```
   MMI                               SMS                            SIM/MM
     |                                |                                |
 (1) |       MNSMS_COMMAND_REQ        |                                |
     * =============================> *                                |
 (2) |                                |      MMSMS_ESTABLISH_REQ       |
     |                                * =============================> *
 (3) |                                |         SIM_UPDATE_REQ         |
     |                                * =============================> *
MUTE(500)
 (4) |                                |      MMSMS_ESTABLISH_CNF       |
     |                                * <============================= *
 (5) |                                |         MMSMS_DATA_REQ         |
     |                                |           (CP_DATA)            |
     |                                * =============================> *
MUTE(500)
 (6) |                                |         MMSMS_DATA_IND         |
     |                                |            (CP_ACK)            |
     |                                * <============================= *
MUTE(500)
 (7) |                                |         MMSMS_DATA_IND         |
     |                                |           (CP_DATA)            |
     |                                * <============================= *
 (8) |                                |         MMSMS_DATA_REQ          |
     |                                |            (CP_ACK)             |
     |                                * ==============================> *
 (9) |       MNSMS_COMMAND_CNF        |                                 |
     * <============================= *                                 |
(10) |                                |       MMSMS_RELEASE_REQ         |
     |                                * ==============================> *
MUTE(500)
(11) |                                |         SIM_UPDATE_CNF          |
     |                                * <============================== *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_COMMAND_REQ		
<A>	sms_sdu	SMS_SDU_COMMAND_STAT_REQ
<B>	sms_sdu	SMS_SDU_COMMAND_ENQ
<C>	sms_sdu	SMS_SDU_COMMAND_CANCEL_REP
<D>	sms_sdu	SMS_SDU_COMMAND_DEL
(2) MMSMS_ESTABLISH_REQ		
	ti	TI_MO
(3) SIM_UPDATE_REQ		
	source	SRC_SMS
	offset	OFFSET_0
	datafield	SIM_SMSS
	length	SIMREC_SMSS_MSG_REF_LEN
<A>	trans_data	SIMREC_SMSS_MSG_REF
<B>	trans_data	SIMREC_SMSS_MSG_REF
<C>	trans_data	SIMREC_SMSS_MSG_REF
<D>	trans_data	SIMREC_SMSS_MSG_REF_N2
(4) MMSMS_ESTABLISH_CNF		
	ti	TI_MO
(5) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MO
<A>	cp_user_data_ul	RP_DATA_CMD_STAT_REQ
<B>	cp_user_data_ul	RP_DATA_CMD_ENQ
<C>	cp_user_data_ul	RP_DATA_CMD_CANCEL_REP
<D>	cp_user_data_ul	RP_DATA_CMD_DEL
	}	

(6) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MO_TO_MS
}	

(7) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	D_CP_DATA
ti	TI_MO_TO_MS
cp_user_data_dl	RP_ACK_DLNK
}	

(8) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MO
}	

(9) MNSMS_COMMAND_CNF

	cause	SMS_NO_ERROR
<A>	tp_mr	TP_MR_3N1
<B>	tp_mr	TP_MR_3N1
<C>	tp_mr	TP_MR_3N1
<D>	tp_mr	TP_MR_3N2
	sms_sdu	SMS_SDU_EMPTY

(10) MMSMS_RELEASE_REQ

ti	TI_MO

(11) SIM_UPDATE_CNF

datafield	SIM_SMSS
cause	SIM_NO_ERROR

History:	6-Jan-98	SZ	Initial
	31-Oct-2000	LW	Added variants B - D
	21-Dec-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.6 Mobile Terminated Short Message Service

3.6.1 SMS110: Reception of a Short Message (Preamble)

Description: MM indicates establishment of a SMS connection. This initial primitive contains a CP-DATA message. The content of the CP-DATA message is a RP-DATA message which is forwarded to the Relay Layer. Control Procotol sends a CP-ACK message as acknowlegement to the infrastructure. The following handling of the mobile terminated short message depends on the protocol identifier and the data coding scheme of the message. In this case it is a class 2 message which will be stored

Preamble: SMS031C

```
   MMI                                SMS                               SIM/MM
     |                                 |                                  |
(1)  |                                 |       MMSMS_ESTABLISH_IND        |
     |                                 |            (CP_DATA)             |
     |                                 * <============================== *
(2)  |                                 |          MMSMS_DATA_REQ          |
     |                                 |             (CP_ACK)             |
     |                                 * ==============================> *
(3)  |                                 |      SIM_UPDATE_RECORD_REQ       |
     |                                 * ==============================> *
MUTE(500)
     |                                 |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7DEF
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT_DELIVER_7DEF

History:	6-Jan-98	SZ	Initial
	03-Dec-2001	FK	Preamble adapted

3.6.2 SMS111: Reception of a CP-ERROR Message

Description: MM indicates a new SMS connection. The incoming message is a CP-ERROR message. The message is ignored. The SMS connection is released.

Preamble: SMS031C

```
   MMI                              SMS                             SIM/MM
    |                                |                                 |
(1) |                                |       MMSMS_ESTABLISH_IND       |
    |                                |            (CP_ERROR)           |
    |                                * <============================== *
(2) |                                |        MMSMS_RELEASE_REQ        |
    |                                * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                |                                 |
```

Parametrization

	Primitive	Parameter	Value
(1)	MMSMS_ESTABLISH_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ERROR
		ti	TI_MT
		cp_cause	SMS_CP_CS_NETWORK_FAILURE
		}	
(2)	MMSMS_RELEASE_REQ		
		ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	03-Dec-2001	FK	Preamble adapted

3.6.3 SMS112: Reception of a CP-ACK message

Description: A new SMS connection is indicated by MM. The incoming message is CP-ACK. Control Protocol builds a CP-ERROR message with cause #98. The SMS Connection is released.

Preamble: SMS031C

```
   MMI                              SMS                             SIM/MM
    |                                |                                 |
(1) |                                |       MMSMS_ESTABLISH_IND       |
    |                                |             (CP_ACK)            |
    |                                * <============================== *
(2) |                                |          MMSMS_DATA_REQ         |
    |                                |            (CP_ERROR)           |
    |                                * ==============================> *
(3) |                                |        MMSMS_RELEASE_REQ        |
    |                                * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MT
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ERROR
	ti	TI_MT_FROM_MS
	cp_cause	SMS_CP_CS_MSG_NOT_COMP
	}	
(3) MMSMS_RELEASE_REQ		
	ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	03-Dec-2001	FK	Preamble adapted

3.6.4 SMS113: Reception of an unknown message

Description: A new SMS connection is indicated by MM. The incoming message is unknown. Control Protocol builds a CP-ERROR message with cause #97. The SMS Connection is released.

Preamble: SMS031C

```
   MMI                                 SMS                               SIM/MM
    |                                   |                                   |
(1) |                                   |        MMSMS_ESTABLISH_IND        |
    |                                   |            (unknown)              |
    |                                   * <================================ *
(2) |                                   |           MMSMS_DATA_REQ          |
    |                                   |            (CP_ERROR)             |
    |                                   * ================================> *
(3) |                                   |         MMSMS_RELEASE_REQ         |
    |                                   * ================================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                   |                                   |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	UNKN_SMS_MT_MSG

Primitive	Parameter	Value
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ERROR
	ti	TI_MT_FROM_MS
	cp_cause	SMS_CP_CS_INFO_NON_EXIST
	}	
(3) MMSMS_RELEASE_REQ		
	ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	03-Dec-2001	FK	Preamble adapted

3.6.5 SMS121: Second Mobile Terminated Connection

Description: MM indicates a second mobile terminated short message connection with a CP-DATA message. The content of the CP-DATA message is forwarded to the Relay Layer. Only one terminated transaction is allowed in parallel. So a RP-ERROR message is build by the Relay Layer and forwarded to the Control Protocol. The message is included into a CP-DATA message and send to the infrastructure. The SMS connection is released by the Relay Layer. The release request is forwarded to MM.

Preamble: SMS110

```
   MMI                                  SMS                              SIM/MM
     |                                   |                                  |
(1)  |                                   |       MMSMS_ESTABLISH_IND        |
     |                                   |             (CP_DATA)            |
     |                                   * <=============================== *
(2)  |                                   |          MMSMS_DATA_REQ          |
     |                                   |             (CP_DATA)            |
     |                                   * ===============================> *
(3)  |                                   |        MMSMS_RELEASE_REQ         |
     |                                   * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                   |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT_2
	cp_user_data_dl	RP_DATA_DELIVER_2
	}	

(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_2_FROM_MS
	cp_user_data_ul	RP_ERR_PROTOCOL_SECOND
	}	
(3) MMSMS_RELEASE_REQ		
	ti	TI_MT_2_FROM_MS

History:	6-Jan-98	SZ	Initial
	03-Dec-2001	FK	Preamble adapted
	08-Jan-2002	FK	Changes in MSG SMS

3.6.6 SMS122: Additional Message for Mobile Terminated Connection

Description: Control Protocol receives additional CP-DATA messages for an existing transaction. The content of this CP-DATA message is a RP-DATA message. It is forwarded to the Relay Layer. The Relay Layer builds an RP-ERROR message and send it back to the Control Protocol. This message is part of a CP-DATA message which is forwarded to the infrastructure.

Preamble: SMS110

```
   MMI                                    SMS                                 SIM/MM
    |                                      |                                     |
(1) |                                      |          MMSMS_DATA_IND             |
    |                                      |            (CP_DATA)                |
    |                                      * <=============================== *
(2) |                                      |          MMSMS_DATA_REQ             |
    |                                      |             (CP_ACK)                |
    |                                      * ===============================> *
(3) |                                      |          MMSMS_DATA_REQ             |
    |                                      |            (CP_DATA)                |
    |                                      * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                      |                                     |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7DEF
	}	

(2) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MT_FROM_MS
}	

(3) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_FROM_MS
cp_user_data_ul	RP_ERR_PROTOCOL
}	

History:	6-Jan-98	SZ	Initial
	08-Jan-2002	FK	Changes in MSG SMS

3.6.7 SMS131: MT Message, Phase 2+ Handling, Normal Case

Description: In the preamble message handling is configured to be compatible with Pase 2+, so the entity has to wait for acknowledgment of the above layer. The parameter data coding scheme of the mobile terminated short message indicates that it is a class 0 message. This message is displayed immediately and not stored. The Relay Layer waits for MNSMS_ACK_RES to either build a RP-ACK or RP-ERROR message and send it to Control Protocol. The message is part of a CP-DATA message which is build by Control Protocol. The message is sent to the infrastructure. The SMS connection will be released by the Relay Layer. The release of SMS connection is forwarded to MM.

Variants: <A>...<D>

Preamble: SMS033D

```
   MMI                               SMS                             SIM/MM
    |                                 |                                 |
(1) |                                 |       MMSMS_ESTABLISH_IND       |
    |                                 |            (CP_DATA)            |
    |                                 * <============================== *
(2) |                                 |          MMSMS_DATA_REQ         |
    |                                 |            (CP_ACK)             |
    |                                 * ==============================> *
(3) |        MNSMS_MESSAGE_IND        |                                 |
    * <============================== *                                 |
MUTE(2000)
(4) |          MNSMS_ACK_RES          |                                 |
    * ==============================> *                                 |
(4) |                                 |          MMSMS_DATA_REQ         |
    |                                 |            (CP_DATA)            |
    |                                 * ==============================> *
(5) |                                 |          MMSMS_DATA_IND         |
    |                                 |            (CP_ACK)             |
    |                                 * <============================== *
```

```
(6)  |                          |        MMSMS_RELEASE_REQ        |
     |                          * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                          |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL0
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) MNSMS_MESSAGE_IND		
	mem_type	NOT_USED
	rec_num	SMS_RECORD_NOT_EXIST
	rec_max	NOT_USED
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_DELIVER_7CL0
(4) MNSMS_ACK_RES		
<A>	resp	SMS_RP_ACK
<B>	resp	SMS_RP_ACK
<C>	resp	SMS_RP_ERROR
<D>	resp	SMS_RP_ERROR
<A>	sms_sdu	SMS_SDU_EMPTY
<B>	sms_sdu	SMS_SDU_DLVR_REP_ACK
<C>	sms_sdu	SMS_SDU_EMPTY
<D>	sms_sdu	SMS_SDU_DLVR_REP_ERR

(5) MMSMS_DATA_REQ

	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
<A>	cp_user_data_ul	RP_ACK_RESP
<B>	cp_user_data_ul	RP_ACK_DLVR_REP
<C>	cp_user_data_ul	RP_ERR_RESP
<D>	cp_user_data_ul	RP_ERR_DLVR_REP
	}	

(6) MMSMS_DATA_IND

	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MT
	}	

(7) MMSMS_RELEASE_REQ

	ti	TI_MT_FROM_MS

History:	29-Nov-99	FK	Initial
	13-Dec-2000	LW	Adapted to new TAP
	23-Jan-2000	FK	Major rework, new TC number

3.6.8 SMS132: Class 0 Message, Phase 2+ Handling, no Response

Description: In the preamble message handling is configured to be compatible with Phase 2+, so the entity has to wait for acknowledgement of the above layer. The parameter data coding scheme of the mobile terminated short message indicates that it is a class 0 message. This message is displayed immediately and not stored. The Relay Layer waits for MNSMS_ACK_RES to either build a RP-ACK or RP-ERROR message. There is no response, so after Timeout a RP-ERROR message is sent it to Control Protocol. The message is part of a CP-DATA message, which is build by Control Protocol. The message is sent to the infrastructure. The SMS connection will be released by the Relay Layer. The release of SMS connection is forwarded to MM.

Preamble: SMS033D

```
   MMI                                SMS                            SIM/MM
    |                                  |                                |
(1) |                                  |       MMSMS_ESTABLISH_IND      |
    |                                  |            (CP_DATA)           |
    |                                  * <============================= *
(2) |                                  |          MMSMS_DATA_REQ        |
    |                                  |             (CP_ACK)           |
    |                                  * =============================> *
(3) |        MNSMS_MESSAGE_IND         |                                |
    * <=============================== *                                |
MUTE (12500)
(4) |          MNSMS_ERROR_IND         |                                |
    * <=============================== *                                |
(4) |                                  |          MMSMS_DATA_REQ        |
    |                                  |            (CP_DATA)           |
    |                                  * =============================> *
```

```
(5)  |                                |         MMSMS_DATA_IND          |
     |                                |            (CP_ACK)             |
     |                                * <============================== *
(6)  |                                |       MMSMS_RELEASE_REQ         |
     |                                * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL0
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) MNSMS_MESSAGE_IND		
	mem_type	NOT_USED
	rec_num	SMS_RECORD_NOT_EXIST
	rec_max	NOT_USED
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_DELIVER_7CL0
(4) MNSMS_ERROR_IND		
	cause	SMS_CAUSE_NET_TIMEOUT
(5) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ERR_PROTOCOL_AA
	}	

(6) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MT
	}	
(7) MMSMS_RELEASE_REQ		
	ti	TI_MT_FROM_MS

History:	29-Nov-99	FK	Initial
	7-Nov-2000	LW	Timeout
	12-Dec-2000	LW	Adaption to new TAP
	23-Jan-2000	FK	Major rework, new TC number
	25-Oct-2002	FK	Adaption to Cause Concept

3.6.9 SMS133: Class 0 Message, Phase 2+ Handling, Failure case

Description: SMS receives an unsolicited primitive MNSMS_ACK_RES. The failure cases are: Phase 2+ not supported (A), no outstanding response (B) and response timed out (C).

Variants: <A>...<C>

Preamble:

<A>	SMS031C
<B>	SMS033D
<C>	SMS132

```
   MMI                                    SMS                                     SIM/MM
     |                                     |                                         |
(1)  |          MNSMS_ACK_RES              |                                         |
     * ============================> *                                               |
(2)  |         MNSMS_ERROR_IND             |                                         |
     * <============================== *                                             |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                     |                                         |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_ACK_RES		
	resp	SMS_RP_ACK
	sms_sdu	SMS_SDU_EMPTY
(2) MNSMS_ERROR_IND		
<A>	cause	SMS_CAUSE_OPER_NOT_ALLW
<B>	cause	SMS_CAUSE_UNEXP_CNMA
<C>	cause	SMS_CAUSE_UNEXP_CNMA

History:	23-Jan-2002	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.6.10 SMS151: Class 0 Message

Description: The incoming mobile terminated short message receives in the Relay Layer. The parameter data coding scheme indicates that it is a class 0 message. This message is displayed immediately and not stored. Relay Layer builds a RP-ACK message and sends it to Control Protocol. The message is part of a CP-DATA message which is build by Control Protocol. The message is sent to the infrastructure. The SMS connection will be released by the Relay Layer. The release of SMS connection is forwarded to MM.

Variants: <A>...<F>

Preamble:

<A>	SMS031D
<B>	SMS031D
<C>	SMS522
<D>	SMS523
<E>	SMS524B
<F>	SMS525B

```
   MMI                                SMS                              SIM/MM
     |                                  |                                   |
(1)  |                                  |       MMSMS_ESTABLISH_IND         |
     |                                  |            (CP_DATA)              |
     |                                  * <============================== *
(2)  |                                  |          MMSMS_DATA_REQ           |
     |                                  |             (CP_ACK)              |
     |                                  * ==============================> *
(3)  |        MNSMS_MESSAGE_IND         |                                   |
     * <============================== *                                   |
(4)  |                                  |          MMSMS_DATA_REQ           |
     |                                  |            (CP_DATA)              |
     |                                  * ==============================> *
(5)  |                                  |          MMSMS_DATA_IND           |
     |                                  |             (CP_ACK)              |
     |                                  * <============================== *
(6)  |                                  |        MMSMS_RELEASE_REQ          |
     |                                  * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                  |                                   |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
<A>	cp_user_data_dl	RP_DATA_DELIVER_7CL0
<B>	cp_user_data_dl	RP_DATA_DELIVER_7CL0_DEF
<C>	cp_user_data_dl	RP_DATA_DELIVER_7CL0_GDC
<D>	cp_user_data_dl	RP_DATA_DELIVER_8CL0_DEF
<E>	cp_user_data_dl	RP_DATA_DELIVER_8CL0_GDC
<F>	cp_user_data_dl	RP_DATA_DELIVER_16CL0_GDC
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	

(3) MNSMS_MESSAGE_IND

	mem_type	NOT_USED
	rec_num	SMS_RECORD_NOT_EXIST
	rec_max	NOT_USED
	status	SMS_RECORD_REC_UNREAD
<A>	sms_sdu	SMS_SDU_DELIVER_7CL0
<B>	sms_sdu	SMS_SDU_DELIVER_7CL0_DEF
<C>	sms_sdu	SMS_SDU_DELIVER_7CL0_GDC
<D>	sms_sdu	SMS_SDU_DELIVER_8CL0_DEF
<E>	sms_sdu	SMS_SDU_DELIVER_8CL0_GDC
<F>	sms_sdu	SMS_SDU_DELIVER_16CL0_GDC

(4) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_FROM_MS
cp_user_data_ul	RP_ACK_RESP
}	

(5) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(6) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	29-Feb-2000	FK	Various DCS
	13-Dec-2000	LW	Adapted to new TAP
	08-Jan-2002	FK	Major rework

3.6.11 SMS152: Class 0 Message (long message)

Description: The incoming mobile terminated short message receives in the Relay Layer. The parameter data coding scheme indicates that it is a class 0 message. This message is displayed immediately and not stored, because the preamble sets parameter <mt>=2.. The message is displayed to the user. Relay Layer builds a RP-ACK message and sends it to Control Protocol. The message is part of a CP-DATA message which is build by Control Protocol. The message is sent to the infrastructure. The SMS connection will be released by the Relay Layer. The release of SMS connection is forwarded to MM.

Preamble: SMS031D

```
  MMI                                   SMS                        SIM/MM
    |                                     |                             |
(1) |                                     |    MMSMS_ESTABLISH_IND      |
    |                                     |        (CP_DATA)            |
    |                                     * <========================== *
(2) |                                     |       MMSMS_DATA_REQ        |
    |                                     |         (CP_ACK)            |
    |                                     * ==========================> *
```

```
 (3) |        MNSMS_MESSAGE_IND         |                                 |
     * <============================== *                                 |
 (4) |                                  |         MMSMS_DATA_REQ          |
     |                                  |           (CP_DATA)             |
     |                                  * ==============================> *
 (5) |                                  |         MMSMS_DATA_IND          |
     |                                  |            (CP_ACK)             |
     |                                  * <============================== *
 (6) |                                  |        MMSMS_RELEASE_REQ        |
     |                                  * ==============================> *
     |                                  |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL0L
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) MNSMS_MESSAGE_IND		
	mem_type	NOT_USED
	rec_num	SMS_RECORD_NOT_EXIST
	rec_max	NOT_USED
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_DELIVER_7CL0L
(4) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ACK_RESP
	}	

(5) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MT
	}	
(6) MMSMS_RELEASE_REQ		
	ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	08-Jan-2002	FK	Major rework

3.6.12 SMS153: Class 0 Message (long message), stored on SIM

Description: The incoming mobile terminated short message receives in the Relay Layer. The parameter data coding scheme indicates that it is a class 0 message. This message is stored on <mem3>, because the preamble sets parameter <mt>=1 or 3. The message is displayed to the user. Relay Layer builds a RP-ACK message and sends it to Control Protocol. The message is part of a CP-DATA message which is build by Control Protocol. The message is sent to the infrastructure. The SMS connection will be released by the Relay Layer. The release of SMS connection is forwarded to MM.

Variants: <A>...<B>

Preamble:

<A>	SMS031C
<B>	SMS031E

```
   MMI                                SMS                              SIM/MM
    |                                  |                                  |
(1) |                                  |      MMSMS_ESTABLISH_IND         |
    |                                  |           (CP_DATA)              |
    |                                  * <=============================== *
(2) |                                  |         MMSMS_DATA_REQ           |
    |                                  |            (CP_ACK)              |
    |                                  * ===============================> *
(3) |                                  |      SIM_UPDATE_RECORD_REQ       |
    |                                  * ===============================> *
MUTE(500)
(4) |                                  |      SIM_UPDATE_RECORD_CNF       |
    |                                  * <=============================== *
(5) |        MNSMS_MESSAGE_IND         |                                  |
    * <=============================== *                                  |
(6) |                                  |         MMSMS_DATA_REQ           |
    |                                  |           (CP_DATA)              |
    |                                  * ===============================> *
(7) |                                  |         MMSMS_DATA_IND           |
    |                                  |            (CP_ACK)              |
    |                                  * <=============================== *
(8) |                                  |       MMSMS_RELEASE_REQ          |
    |                                  * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                  |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL0L
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_0L
(4) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_NO_ERROR
(5) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_1
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_DELIVER_7CL0L
(6) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ACK_RESP
	}	

(7) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(8) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	10-Jan-2002	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.6.13 SMS161: Class 1 - Reception of Message (stored in SIM-Card)

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 1 message. According to the setting of parameter -<mt> in the preamble these messages are not displayed directly to the user. They are stored in the ME memory if available, else on the SIM card. There is no ME memory existent or available, so it is stored on the SIM card.

Variants: <A>...<D>

Preamble:

<A>	SMS031C
<B>	SMS031E
<C>	SMS034C
<D>	SMS034F

```
   MMI                               SMS                             SIM/MM
     |                                |                                |
(1)  |                                |      MMSMS_ESTABLISH_IND       |
     |                                |           (CP_DATA)            |
     |                                * <============================= *
(2)  |                                |         MMSMS_DATA_REQ         |
     |                                |            (CP_ACK)            |
     |                                * =============================> *
(3)  |                                |     SIM_UPDATE_RECORD_REQ      |
     |                                * =============================> *
MUTE(500)
(4)  |                                |     SIM_UPDATE_RECORD_CNF      |
     |                                * <============================= *
(5)  |       MNSMS_MESSAGE_IND        |                                |
     * <============================= *                                |
(6)  |                                |         MMSMS_DATA_REQ         |
     |                                |           (CP_DATA)            |
     |                                * =============================> *
(7)  |                                |         MMSMS_DATA_IND         |
     |                                |            (CP_ACK)            |
     |                                * <============================= *
(8)  |                                |       MMSMS_RELEASE_REQ        |
     |                                * =============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL1
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_1
(4) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_NO_ERROR
(5) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_1
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_DELIVER_7CL1
(6) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ACK_ULNK
	}	

(7) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(8) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:			
	6-Jan-98	SZ	Initial
	09-Jan-2002	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.6.14 SMS162: Class 1 - Reception of Message (stored in ME Memory)

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 1 message. According to the setting of parameter -<mt> in the preamble these messages are are stored in the mobile station memory (Variant A, B) or routed to the TE (Variant C, D).

Variants: <A>...<D>

Preamble:

<A>	SMS034D
<B>	SMS034E
<C>	SMS034J
<D>	SMS034K

```
   MMI                               SMS                             SIM/MM
    |                                 |                                |
(1) |                                 |      MMSMS_ESTABLISH_IND       |
    |                                 |            (CP_DATA)           |
    |                                 * <============================= *
(2) |                                 |         MMSMS_DATA_REQ         |
    |                                 |            (CP_ACK)            |
    |                                 * =============================> *
(5) |        MNSMS_MESSAGE_IND        |                                |
    * <============================== *                                |
(6) |                                 |         MMSMS_DATA_REQ         |
    |                                 |            (CP_DATA)           |
    |                                 * =============================> *
(7) |                                 |         MMSMS_DATA_IND         |
    |                                 |            (CP_ACK)            |
    |                                 * <============================= *
(8) |                                 |       MMSMS_RELEASE_REQ        |
    |                                 * =============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                 |                                |
```

Parametrization

	Primitive	Parameter	Value
(1)	MMSMS_ESTABLISH_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	D_CP_DATA
		ti	TI_MT
		cp_user_data_dl	RP_DATA_DELIVER_7CL1
		}	
(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	B_CP_ACK
		ti	TI_MT_FROM_MS
		}	
(3)	MNSMS_MESSAGE_IND		
	<A>	mem_type	MEM_ME
	<B>	mem_type	MEM_ME
	<C>	mem_type	NOT_USED
	<D>	mem_type	NOT_USED
	<A>	rec_num	SIM_RECORD_1
	<B>	rec_num	SIM_RECORD_3
	<C>	rec_num	SMS_RECORD_NOT_EXIST
	<D>	rec_num	SMS_RECORD_NOT_EXIST
	<A>	rec_max	SIM_RECORD_3
	<B>	rec_max	SIM_RECORD_3
	<C>	rec_max	NOT_USED
	<D>	rec_max	NOT_USED
		status	SMS_RECORD_REC_UNREAD
		sms_sdu	SMS_SDU_DELIVER_7CL1
(4)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	U_CP_DATA
		ti	TI_MT_FROM_MS
		cp_user_data_ul	RP_ACK_ULNK
		}	

(5) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(6) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History: 18-Jun-2002 FK Initial

3.6.15 SMS163: Class 1 - Reception of Message (routed to TE)

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 1 message. According to the setting of parameter -<mt> in the preamble these messages are routed to the TE, without regarding the state of the memory.

Variants: <A>...<F>

Preamble:

<A>	SMS034G
<B>	SMS034H
<C>	SMS034J
<D>	SMS034K
<E>	SMS034I
<F>	SMS034L

```
   MMI                                  SMS                                 SIM/MM
     |                                   |                                    |
(1)  |                                   |        MMSMS_ESTABLISH_IND         |
     |                                   |             (CP_DATA)              |
     |                                   * <================================= *
(2)  |                                   |           MMSMS_DATA_REQ           |
     |                                   |             (CP_ACK)               |
     |                                   * =================================> *
(5)  |         MNSMS_MESSAGE_IND         |                                    |
     * <================================ *                                    |
(6)  |                                   |           MMSMS_DATA_REQ           |
     |                                   |             (CP_DATA)              |
     |                                   * =================================> *
(7)  |                                   |           MMSMS_DATA_IND           |
     |                                   |             (CP_ACK)               |
     |                                   * <================================= *
(8)  |                                   |         MMSMS_RELEASE_REQ          |
     |                                   * =================================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                   |                                    |
```

Parametrization

Primitive	Parameter	Value
(7) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL1
	}	
(8) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(9) MNSMS_MESSAGE_IND		
	mem_type	NOT_USED
	rec_num	SMS_RECORD_NOT_EXIST
	rec_max	NOT_USED
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_DELIVER_7CL1
(10) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ACK_ULNK
	}	
(11) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MT
	}	
(12) MMSMS_RELEASE_REQ		
	ti	TI_MT_FROM_MS

History:	18-Jun-2002	FK	Initial

3.6.16 SMS164: Class 1 - Storage available on SIM, writing is NOT successfull

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 1 message. This messages are not displayed directly to the user. They are stored in the mobile station memory if available, else on the SIM card. There is no mobile station memory, so it is stored on the SIM card. Storing on the SIM card fails. A RP-Error message is send to the infrastructure.

Variants: <A>...<B>

Preamble:

<A> SMS031C
<B> SMS031E

```
   MMI                                SMS                              SIM/MM
     |                                 |                                  |
(1)  |                                 |       MMSMS_ESTABLISH_IND        |
     |                                 |            (CP_DATA)             |
     |                                 * <=============================== *
(2)  |                                 |          MMSMS_DATA_REQ          |
     |                                 |             (CP_ACK)             |
     |                                 * ===============================> *
(3)  |                                 |      SIM_UPDATE_RECORD_REQ       |
     |                                 * ===============================> *
MUTE(500)
(4)  |                                 |      SIM_UPDATE_RECORD_CNF       |
     |                                 * <=============================== *
(5)  |                                 |          SIM_UPDATE_REQ          |
     |                                 * ===============================> *
(6)  |                                 |          MMSMS_DATA_REQ          |
     |                                 |            (CP_DATA)             |
     |                                 * ===============================> *
MUTE(500)
(7)  |                                 |          MMSMS_DATA_IND          |
     |                                 |             (CP_ACK)             |
     |                                 * <=============================== *
(8)  |                                 |        MMSMS_RELEASE_REQ         |
     |                                 * ===============================> *
MUTE(500)
(9)  |                                 |          SIM_UPDATE_CNF          |
     |                                 * <=============================== *
MUTE(1000)
COMMAND (SMS STATUS PARTITION)
     |                                 |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL1
	}	

(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	B_CP_ACK
		ti	TI_MT_FROM_MS
		}	
(3)	SIM_UPDATE_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
		record	SIM_RECORD_1
		length	LENGTH_SMS
		linear_data	SIM_SMS_CLASS_1
(4)	SIM_UPDATE_RECORD_CNF		
		datafield	SIM_SMS
		record	SIM_RECORD_1
		cause	SIM_CAUSE_OTHER_ERROR
(5)	SIM_UPDATE_REQ		
		source	SRC_SMS
		offset	OFFSET_1
		datafield	SIM_SMSS
		length	SIM_SMSS_MEM_CAP_ONLY_LEN
		trans_data	SIM_SMSS_NO_MEM_CAP
(6)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	U_CP_DATA
		ti	TI_MT_FROM_MS
		cp_user_data_ul	RP_ERR_MEM_CAP_EXC
		}	
(7)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(8)	MMSMS_RELEASE_REQ		
		ti	TI_MT_FROM_MS
(9)	SIM_UPDATE_CNF		
		datafield	SIM_SMSS
		cause	SIM_NO_ERROR

History:	6-Jan-98	SZ	Initial
	09-Jan-2002	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.6.17 SMS171: Class 2 - Storage available on SIM, writing is successfull

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 2 message. This messages are not displayed directly to the user. They are stored on the SIM card irrespectively to the configuration setting and an indication is sent to the user.
Variant B, C check the normal handling of One2One SIM-specific message, when no One2One SIM is present.
Variant E, F check that configuring ME memory does not affect the routing of class 2 messages.
Variant G fills the last record on SIM; ME memory is not existent.
Variant H checks the repeatability of the TC.

Variants: <A>...<H>

Preamble:

<A>	SMS031C
<B>	SMS031D
<C>	SMS031E
<D>	SMS151A
<E>	SMS034D
<F>	SMS034K
<G>	SMS031I
<H>	SMS171B

```
   MMI                              SMS                             SIM/MM
    |                                |                                |
(1) |                                |       MMSMS_ESTABLISH_IND      |
    |                                |            (CP_DATA)           |
    |                                * <============================= *
(2) |                                |          MMSMS_DATA_REQ        |
    |                                |             (CP_ACK)           |
    |                                * =============================> *
(3) |                                |      SIM_UPDATE_RECORD_REQ     |
    |                                * =============================> *
MUTE(500)
(4) |                                |      SIM_UPDATE_RECORD_CNF     |
    |                                * <============================= *
(5) |        MNSMS_MESSAGE_IND       |                                |
    * <============================= *                                |
(6) |                                |          MMSMS_DATA_REQ        |
    |                                |            (CP_DATA)           |
    |                                * =============================> *
MUTE(500)
(7) |                                |          MMSMS_DATA_IND        |
    |                                |             (CP_ACK)           |
    |                                * <============================= *
(8) |                                |         MMSMS_RELEASE_REQ      |
    |                                * =============================> *
MUTE(1000)
COMMAND (SMS STATUS PARTITION)
    |                                |                                |
```

Parametrization

	Primitive	Parameter	Value
(1)	MMSMS_ESTABLISH_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	D_CP_DATA
		ti	TI_MT
	<A>	cp_user_data_dl	RP_DATA_DELIVER_7CL2
	<B>	cp_user_data_dl	RP_DATA_DELIVER_121_B
	<C>	cp_user_data_dl	RP_DATA_DELIVER_121_C
	<D>	cp_user_data_dl	RP_DATA_DELIVER_7CL2
	<E>	cp_user_data_dl	RP_DATA_DELIVER_7CL2
	<F>	cp_user_data_dl	RP_DATA_DELIVER_7CL2
	<G>	cp_user_data_dl	RP_DATA_DELIVER_7CL2
	<H>	cp_user_data_dl	RP_DATA_DELIVER_7CL2
		}	
(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	B_CP_ACK
		ti	TI_MT_FROM_MS
		}	
(3)	SIM_UPDATE_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_1
	<B>	record	SIM_RECORD_1
	<C>	record	SIM_RECORD_1
	<D>	record	SIM_RECORD_1
	<E>	record	SIM_RECORD_1
	<F>	record	SIM_RECORD_1
	<G>	record	SIM_RECORD_2
	<H>	record	SIM_RECORD_2
		length	LENGTH_SMS
	<A>	linear_data	SIM_SMS_CLASS_2
	<B>	linear_data	SIM_SMS_DELIVER_121_B
	<C>	linear_data	SIM_SMS_DELIVER_121_C
	<D>	linear_data	SIM_SMS_CLASS_2
	<E>	linear_data	SIM_SMS_CLASS_2
	<F>	linear_data	SIM_SMS_CLASS_2
	<G>	linear_data	SIM_SMS_CLASS_2
	<H>	linear_data	SIM_SMS_CLASS_2

(4)	SIM_UPDATE_RECORD_CNF		
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_1
	<B>	record	SIM_RECORD_1
	<C>	record	SIM_RECORD_1
	<D>	record	SIM_RECORD_1
	<E>	record	SIM_RECORD_1
	<F>	record	SIM_RECORD_1
	<G>	record	SIM_RECORD_2
	<H>	record	SIM_RECORD_2
		cause	SIM_NO_ERROR
(5)	MNSMS_MESSAGE_IND		
		mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_1
	<B>	rec_num	SIM_RECORD_1
	<C>	rec_num	SIM_RECORD_1
	<D>	rec_num	SIM_RECORD_1
	<E>	rec_num	SIM_RECORD_1
	<F>	rec_num	SIM_RECORD_1
	<G>	rec_num	SIM_RECORD_2
	<H>	rec_num	SIM_RECORD_2
		rec_max	SIM_RECORD_3
		status	SMS_RECORD_REC_UNREAD
	<A>	sms_sdu	SMS_SDU_DELIVER_7CL2
	<B>	sms_sdu	SMS_SDU_DELIVER_121_B
	<C>	sms_sdu	SMS_SDU_DELIVER_121_C
	<D>	sms_sdu	SMS_SDU_DELIVER_7CL2
	<E>	sms_sdu	SMS_SDU_DELIVER_7CL2
	<F>	sms_sdu	SMS_SDU_DELIVER_7CL2
	<G>	sms_sdu	SMS_SDU_DELIVER_7CL2
	<H>	sms_sdu	SMS_SDU_DELIVER_7CL2
(6)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	U_CP_DATA
		ti	TI_MT_FROM_MS
		cp_user_data_ul	RP_ACK_ULNK
		}	
(7)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(8)	MMSMS_RELEASE_REQ		
		ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	09-Jan-2002	FK	Major rework
	25-Apr-2002	FK	Variant B, C with One2One SIM-specific message

18-Jun-2002	FK	Handling of ME memory added
25-Oct-2002	FK	Adaption to Cause Concept

3.6.18 SMS172: Class 2 - Storage available on SIM, writing not successfull, MCEF set

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 2 message. This messages are not displayed directly to the user. They are stored on the SIM card. Storing on the SIM card fails. A RP-Error message (Memory Capacity Exceeded) is sent to the network, because ME memory is not existent or not available.

Variants: <A>...<G>

Preamble:

<A>	SMS031C
<B>	SMS031D
<C>	SMS031E
<D>	SMS034C
<E>	SMS034F
<F>	SMS034I
<G>	SMS034L

```
   MMI                               SMS                            SIM/MM
     |                                |                                |
(1)  |                                |      MMSMS_ESTABLISH_IND       |
     |                                |           (CP_DATA)            |
     |                                * <============================= *
(2)  |                                |         MMSMS_DATA_REQ         |
     |                                |            (CP_ACK)            |
     |                                * =============================> *
(3)  |                                |     SIM_UPDATE_RECORD_REQ      |
     |                                * =============================> *
MUTE(500)
(4)  |                                |     SIM_UPDATE_RECORD_CNF      |
     |                                * <============================= *
(5)  |                                |         SIM_UPDATE_REQ         |
     |                                * =============================> *
(6)  |                                |         MMSMS_DATA_REQ         |
     |                                |           (CP_DATA)            |
     |                                * =============================> *
MUTE(500)
(7)  |                                |         SIM_UPDATE_CNF         |
     |                                * <============================= *
MUTE(500)
(7)  |                                |         MMSMS_DATA_IND         |
     |                                |            (CP_ACK)            |
     |                                * <============================= *
(8)  |                                |       MMSMS_RELEASE_REQ        |
     |                                * =============================> *
MUTE (1000)
     |                                |                                |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL2
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_2
(4) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_CAUSE_OTHER_ERROR
(5) SIM_UPDATE_REQ		
	source	SRC_SMS
	offset	OFFSET_1
	datafield	SIM_SMSS
	length	SIM_SMSS_MEM_CAP_ONLY_LEN
	trans_data	SIM_SMSS_NO_MEM_CAP
(6) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ERR_MEM_CAP_EXC
	}	
(7) SIM_UPDATE_CNF		
	datafield	SIM_SMSS
	cause	SIM_NO_ERROR

(8) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(9) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	10-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.6.19 SMS173: Class 2 - Storage not available on SIM/ME, MCEF to be set

Description: The preamble indicates, that the last free SIM SMS records was written. The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 2 message. This messages are not displayed directly to the user. They are stored on the SIM card. Storing on the SIM is not possible, because all records are used. The Memory Capacity Exceeding Flag on the SIM is set and a RP-Error message is sent to the network.

Variants: <A>...<D>

Preamble:

<A>	SMS031K
<B>	SMS031P
<C>	SMS034P
<D>	SMS171G

```
   MMI                               SMS                             SIM/MM
     |                                |                                 |
(1)  |                                |       MMSMS_ESTABLISH_IND       |
     |                                |            (CP_DATA)            |
     |                                * <============================== *
(2)  |                                |         MMSMS_DATA_REQ          |
     |                                |             (CP_ACK)            |
     |                                * ==============================> *
(3)  |                                |         SIM_UPDATE_REQ          |
     |                                * ==============================> *
(4)  |                                |         MMSMS_DATA_REQ          |
     |                                |            (CP_DATA)            |
     |                                * ==============================> *
MUTE(500)
(6)  |                                |         SIM_UPDATE_CNF          |
     |                                * <============================== *
MUTE(500)
(7)  |                                |         MMSMS_DATA_IND          |
     |                                |             (CP_ACK)            |
     |                                * <============================== *
(8)  |                                |        MMSMS_RELEASE_REQ        |
     |                                * ==============================> *
MUTE (1000)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL2
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_UPDATE_REQ		
	source	SRC_SMS
	offset	OFFSET_1
	datafield	SIM_SMSS
	length	SIMREC_SMSS_MEM_FLAG_LEN
	trans_data	SIMREC_SMSS_MCEF
(4) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ERR_MEM_CAP_EXC
	}	
(5) SIM_UPDATE_CNF		
	datafield	SIM_SMSS
	cause	SIM_NO_ERROR
(6) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MT
	}	
(7) MMSMS_RELEASE_REQ		
	ti	TI_MT_FROM_MS

History:			
	18-Jan-2000	FK	Initial
	10-Jan-2001	FK	Major rework
	18-Jun-2002	FK	Handling of ME memory added
	25-Oct-2002	FK	Adaption to Cause Concept

3.6.20 SMS174: Class 2 - Storage not available on SIM/ME, MCEF already set.

Description: The preamble indicates, that all SIM SMS records are used. The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 2 message. This messages are not displayed directly to the user. They are stored on the SIM card. Storing on the SIM is not possible, because all records are used. A RP-Error message is sent to the network. MCEF is already set, therefore it need not to be set again.

Variants: <A>...<H>

Preamble:

<A>	SMS031J
<B>	SMS034M
<C>	SMS173A
<D>	SMS173B
<E>	SMS173C
<F>	SMS174A
<G>	SMS174B
<H>	SMS174C

```
   MMI                                  SMS                              SIM/MM
    |                                    |                                  |
(1) |                                    |       MMSMS_ESTABLISH_IND        |
    |                                    |            (CP_DATA)             |
    |                                    * <=============================== *
(2) |                                    |          MMSMS_DATA_REQ          |
    |                                    |             (CP_ACK)             |
    |                                    * ===============================> *
(6) |                                    |          MMSMS_DATA_REQ          |
    |                                    |            (CP_DATA)             |
    |                                    * ===============================> *
(7) |                                    |          MMSMS_DATA_IND          |
    |                                    |             (CP_ACK)             |
    |                                    * <=============================== *
(8) |                                    |        MMSMS_RELEASE_REQ         |
    |                                    * ===============================> *
MUTE (1000)
    |                                    |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	MMSMS_ESTABLISH_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	D_CP_DATA
		ti	TI_MT
		cp_user_data_dl	RP_DATA_DELIVER_7CL2
		}	

(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	B_CP_ACK
		ti	TI_MT_FROM_MS
		}	
(3)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	U_CP_DATA
		ti	TI_MT_FROM_MS
		cp_user_data_ul	RP_ERR_MEM_CAP_EXC
		}	
(4)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(5)	MMSMS_RELEASE_REQ		
		ti	TI_MT_FROM_MS

History:	26-Nov-99	FK	Initial
	10-Jan-2001	FK	Major rework
	18-Jun-2002	FK	Handling of ME memory added

3.6.21 SMS175: Class 2 - Storage not available on SIM, but on ME, MCEF not to be set

Description: The preamble indicates, that all SIM SMS records are used, but ME memory is available. The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 2 message. This messages are not displayed directly to the user. They are to be stored on the SIM card. Storing on the SIM is not possible, because all records are used. ME Memory is not occupied, so the Memory Capacity Exceeding Flag on the SIM must not be set and a RP-Error message (Protocol Error) is sent to the network.

Variants: <A>...<D>

Preamble:

<A>	SMS034N
<B>	SMS034O
<C>	SMS175A
<D>	SMS175B

```
   MMI                                        SMS                                    SIM/MM
     |                                          |                                          |
(1)  |                                          |       MMSMS_ESTABLISH_IND                |
     |                                          |            (CP_DATA)                     |
     |                                          * <============================== *
```

```
(2)  |                                |         MMSMS_DATA_REQ          |
     |                                |            (CP_ACK)             |
     |                                * ==============================> *
(6)  |                                |         MMSMS_DATA_REQ          |
     |                                |           (CP_DATA)             |
     |                                * ==============================> *
(7)  |                                |         MMSMS_DATA_IND          |
     |                                |            (CP_ACK)             |
     |                                * <============================== *
(8)  |                                |       MMSMS_RELEASE_REQ         |
     |                                * ==============================> *
MUTE (1000)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL2
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ERR_PROTOCOL
	}	
(4) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MT
	}	

```
(5)    MMSMS_RELEASE_REQ
                                   ti                  TI_MT_FROM_MS
```

History: 18-Jun-2002 FK Initial

3.6.22 SMS176: Class 2 - Storage available on SIM/ME, writing not successfull, MCEF not set

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 2 message. This messages are not displayed directly to the user. They are stored on the SIM card. Storing on the SIM card fails. A RP-Error message ('Protocol Error, unspecified' or 'Memory Capacity execeeded', depending on the availabililty of ME memory) is sent to the network.

Variants: <A>...<J>

Preamble:

<A>	SMS034A
<B>	SMS034B
<C>	SMS034D
<D>	SMS034E
<E>	SMS034H
<F>	SMS034K
<G>	SMS172A
<H>	SMS172D
<I>	SMS176A
<J>	SMS176C

```
   MMI                                 SMS                              SIM/MM
     |                                  |                                 |
(1)  |                                  |       MMSMS_ESTABLISH_IND       |
     |                                  |            (CP_DATA)            |
     |                                  * <=============================== *
(2)  |                                  |          MMSMS_DATA_REQ         |
     |                                  |             (CP_ACK)            |
     |                                  * ===============================> *
(3)  |                                  |      SIM_UPDATE_RECORD_REQ      |
     |                                  * ===============================> *
MUTE(500)
(4)  |                                  |      SIM_UPDATE_RECORD_CNF      |
     |                                  * <=============================== *
(6)  |                                  |          MMSMS_DATA_REQ         |
     |                                  |            (CP_DATA)            |
     |                                  * ===============================> *
MUTE(500)
(7)  |                                  |          MMSMS_DATA_IND         |
     |                                  |             (CP_ACK)            |
     |                                  * <=============================== *
(8)  |                                  |        MMSMS_RELEASE_REQ        |
     |                                  * ===============================> *
MUTE (1000)
     |                                  |                                 |
```

Parametrization

Primitive	Parameter	Value
（1） MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL2
	}	
（2） MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
（3） SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_2
（4） SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_CAUSE_OTHER_ERROR
（5） MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
<A>	cp_user_data_ul	RP_ERR_PROTOCOL
<B>	cp_user_data_ul	RP_ERR_PROTOCOL
<C>	cp_user_data_ul	RP_ERR_PROTOCOL
<D>	cp_user_data_ul	RP_ERR_PROTOCOL
<E>	cp_user_data_ul	RP_ERR_PROTOCOL
<F>	cp_user_data_ul	RP_ERR_PROTOCOL
<G>	cp_user_data_ul	RP_ERR_MEM_CAP_EXC
<H>	cp_user_data_ul	RP_ERR_MEM_CAP_EXC
<I>	cp_user_data_ul	RP_ERR_PROTOCOL
<J>	cp_user_data_ul	RP_ERR_PROTOCOL
	}	

Primitive	Parameter	Value
(6) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MT
	}	
(7) MMSMS_RELEASE_REQ		
	ti	TI_MT_FROM_MS

History:			
History:	8-Jul-2001	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.6.23 SMS181: Class 3 Message

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 3 message. This messages are not displayed directly to the user. They are stored on the SIM card.

Preamble: SMS031C

```
   MMI                                 SMS                               SIM/MM
    |                                   |                                   |
(1) |                                   |       MMSMS_ESTABLISH_IND         |
    |                                   |            (CP_DATA)              |
    |                                   * <=============================== *
(2) |                                   |          MMSMS_DATA_REQ           |
    |                                   |             (CP_ACK)              |
    |                                   * ===============================> *
(3) |                                   |      SIM_UPDATE_RECORD_REQ        |
    |                                   * ===============================> *
(4) |                                   |      SIM_UPDATE_RECORD_CNF        |
    |                                   * <=============================== *
(5) |         MNSMS_MESSAGE_IND         |                                   |
    * <=============================== *                                   |
(6) |                                   |          MMSMS_DATA_REQ           |
    |                                   |            (CP_DATA)              |
    |                                   * ===============================> *
(7) |                                   |          MMSMS_DATA_IND           |
    |                                   |             (CP_ACK)              |
    |                                   * <=============================== *
(8) |                                   |        MMSMS_RELEASE_REQ          |
    |                                   * ===============================> *
    |                                   |                                   |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL3
	}	

(2) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MT_FROM_MS
}	

(3) SIM_UPDATE_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_1
length	LENGTH_SMS
linear_data	SIM_SMS_CLASS_3

(4) SIM_UPDATE_RECORD_CNF

datafield	SIM_SMS
record	SIM_RECORD_1
cause	SIM_NO_ERROR

(5) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SIM_RECORD_1
rec_max	SIM_RECORD_3
status	SMS_RECORD_REC_UNREAD
sms_sdu	SMS_SDU_DELIVER_7CL3

(6) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_FROM_MS
cp_user_data_ul	RP_ACK_ULNK
}	

(7) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(8) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	10-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.6.24 SMS182: Class 3 - Storage available on SIM, writing is not successfull

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 3 message. This messages are not displayed directly to the user. They are stored in the mobile station memory if available, else on the SIM card. There is no mobile station memory, so it is stored on the SIM card. Storing on the SIM card fails. A RP-Error message is send to the infrastructure.

Preamble: SMS031C

```
   MMI                                       SMS                                    SIM/MM
    |                                         |                                       |
(1) |                                         |       MMSMS_ESTABLISH_IND            |
    |                                         |            (CP_DATA)                  |
    |                                         * <================================ *
(2) |                                         |          MMSMS_DATA_REQ               |
    |                                         |             (CP_ACK)                  |
    |                                         * ================================> *
(3) |                                         |       SIM_UPDATE_RECORD_REQ           |
    |                                         * ================================> *
(4) |                                         |       SIM_UPDATE_RECORD_CNF           |
    |                                         * <================================ *
(5) |                                         |          SIM_UPDATE_REQ               |
    |                                         * ================================> *
(6) |                                         |          MMSMS_DATA_REQ               |
    |                                         |            (CP_DATA)                  |
    |                                         * ================================> *
(7) |                                         |          MMSMS_DATA_IND               |
    |                                         |             (CP_ACK)                  |
    |                                         * <================================ *
(8) |                                         |         MMSMS_RELEASE_REQ             |
    |                                         * ================================> *
    |                                         |                                       |
```

Parametrization

	Primitive	Parameter	Value
(1)	MMSMS_ESTABLISH_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	D_CP_DATA
		ti	TI_MT
		cp_user_data_dl	RP_DATA_DELIVER_7CL3
		}	
(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	B_CP_ACK
		ti	TI_MT_FROM_MS
		}	
(3)	SIM_UPDATE_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
		record	SIM_RECORD_1
		length	LENGTH_SMS
		linear_data	SIM_SMS_CLASS_3

(4) SIM_UPDATE_RECORD_CNF

datafield	SIM_SMS
record	SIM_RECORD_1
cause	SIM_CAUSE_OTHER_ERROR

(5) SIM_UPDATE_REQ

source	SRC_SMS
offset	OFFSET_1
datafield	SIM_SMSS
length	SIM_SMSS_MEM_CAP_ONLY_LEN
trans_data	SIM_SMSS_NO_MEM_CAP

(6) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_FROM_MS
cp_user_data_ul	RP_ERR_MEM_CAP_EXC
}	

(7) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(8) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	10-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.6.25 SMS191: No Class - Storage available on SIM, writing is successfull

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a default class message. This messages are not displayed directly to the user. They are stored on the SIM card.
Variant B assures normal handling of One2One SIM-specific message A, when no One2One SIM is recognized

Variants: <A>...<B>

Preamble: SMS031C

```
   MMI                                     SMS                           SIM/MM
    |                                       |                               |
(1) |                                       |       MMSMS_ESTABLISH_IND      |
    |                                       |           (CP_DATA)            |
    |                                       * <============================= *
(2) |                                       |          MMSMS_DATA_REQ        |
    |                                       |            (CP_ACK)            |
    |                                       * =============================> *
(3) |                                       |      SIM_UPDATE_RECORD_REQ     |
    |                                       * =============================> *
MUTE (500)
```

```
(4)  |                                  |     SIM_UPDATE_RECORD_CNF        |
     |                                  * <=============================== *
(5)  |       MNSMS_MESSAGE_IND          |                                  |
    * <================================ *                                  |
(6)  |                                  |         MMSMS_DATA_REQ           |
     |                                  |            (CP_DATA)             |
     |                                  * ===============================> *
(7)  |                                  |         MMSMS_DATA_IND           |
     |                                  |             (CP_ACK)             |
     |                                  * <=============================== *
(8)  |                                  |       MMSMS_RELEASE_REQ          |
     |                                  * ===============================> *
     |                                  |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	MMSMS_ESTABLISH_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	D_CP_DATA
		ti	TI_MT
	<A>	cp_user_data_dl	RP_DATA_DELIVER_7DEF
	<B>	cp_user_data_dl	RP_DATA_DELIVER_121_A
		}	
(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	B_CP_ACK
		ti	TI_MT_FROM_MS
		}	
(3)	SIM_UPDATE_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
		record	SIM_RECORD_1
		length	LENGTH_SMS
	<A>	linear_data	SIM_SMS_MT_DELIVER_7DEF
	<B>	linear_data	SIM_SMS_DELIVER_121_A
(4)	SIM_UPDATE_RECORD_CNF		
		datafield	SIM_SMS
		record	SIM_RECORD_1
		cause	SIM_NO_ERROR
(5)	MNSMS_MESSAGE_IND		
		mem_type	MEM_SM
		rec_num	SIM_RECORD_1
		rec_max	SIM_RECORD_3
		status	SMS_RECORD_REC_UNREAD
	<A>	sms_sdu	SMS_SDU_DELIVER_7DEF
	<B>	sms_sdu	SMS_SDU_DELIVER_121_A

(6) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_FROM_MS
cp_user_data_ul	RP_ACK_ULNK
}	

(7) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(8) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	10-Jan-2001	FK	Major rework
	25-Apr-2002	FK	New Variant B with One2One SIM-specific message A
	25-Oct-2002	FK	Adaption to Cause Concept

3.6.26 SMS192: No Class - Storage available on SIM, writing is not successfull

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a default class message. This messages are not displayed directly to the user. They are stored in the mobile station memory if available, else on the SIM card. There is no mobile station memory, so it is stored on the SIM card. Storing on the SIM card fails. A RP-Error message is send to the infrastructure.

Preamble: SMS031C

```
   MMI                                   SMS                              SIM/MM
    |                                     |                                 |
(1) |                                     |       MMSMS_ESTABLISH_IND       |
    |                                     |            (CP_DATA)            |
    |                                     * <============================== *
(2) |                                     |         MMSMS_DATA_REQ          |
    |                                     |            (CP_ACK)             |
    |                                     * ==============================> *
(3) |                                     |      SIM_UPDATE_RECORD_REQ      |
    |                                     * ==============================> *
(4) |                                     |      SIM_UPDATE_RECORD_CNF      |
    |                                     * <============================== *
(5) |                                     |         SIM_UPDATE_REQ          |
    |                                     * ==============================> *
(6) |                                     |         MMSMS_DATA_REQ          |
    |                                     |            (CP_DATA)            |
    |                                     * ==============================> *
(7) |                                     |         MMSMS_DATA_IND          |
    |                                     |            (CP_ACK)             |
    |                                     * <============================== *
(8) |                                     |        MMSMS_RELEASE_REQ        |
    |                                     * ==============================> *
    |                                     |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7DEF
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT_DELIVER_7DEF
(4) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_CAUSE_OTHER_ERROR
(5) SIM_UPDATE_REQ		
	source	SRC_SMS
	offset	OFFSET_1
	datafield	SIM_SMSS
	length	SIM_SMSS_MEM_CAP_ONLY_LEN
	trans_data	SIM_SMSS_NO_MEM_CAP
(6) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ERR_MEM_CAP_EXC
	}	

(7)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(8)	MMSMS_RELEASE_REQ		
		ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	10-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.7 Replace MT Short Message

3.7.1 SMS200: Storage of a Replaceable Short Message

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 2 message. The message has the protocol identifier 0x43 and is stored on the SIM card.

Preamble: SMS031C

```
   MMI                                 SMS                             SIM/MM
     |                                  |                                  |
(1)  |                                  |       MMSMS_ESTABLISH_IND        |
     |                                  |            (CP_DATA)             |
     |                                  * <=============================== *
(2)  |                                  |          MMSMS_DATA_REQ          |
     |                                  |             (CP_ACK)             |
     |                                  * ===============================> *
(3)  |                                  |       SIM_UPDATE_RECORD_REQ      |
     |                                  * ===============================> *
MUTE(500)
(4)  |                                  |       SIM_UPDATE_RECORD_CNF      |
     |                                  * <=============================== *
(5)  |        MNSMS_MESSAGE_IND         |                                  |
     * <=============================== *                                  |
(6)  |                                  |          MMSMS_DATA_REQ          |
     |                                  |            (CP_DATA)             |
     |                                  * ===============================> *
(7)  |                                  |          MMSMS_DATA_IND          |
     |                                  |             (CP_ACK)             |
     |                                  * <=============================== *
(8)  |                                  |        MMSMS_RELEASE_REQ         |
     |                                  * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                  |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	MMSMS_ESTABLISH_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	D_CP_DATA
		ti	TI_MT
		cp_user_data_dl	RP_DATA_DELIVER_7CL2_43
		}	
(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	B_CP_ACK
		ti	TI_MT_FROM_MS
		}	
(3)	SIM_UPDATE_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
		record	SIM_RECORD_1
		length	LENGTH_SMS
		linear_data	SIM_SMS_CLASS_2_43
(4)	SIM_UPDATE_RECORD_CNF		
		datafield	SIM_SMS
		record	SIM_RECORD_1
		cause	SIM_NO_ERROR
(5)	MNSMS_MESSAGE_IND		
		mem_type	MEM_SM
		rec_num	SIM_RECORD_1
		rec_max	SIM_RECORD_3
		status	SMS_RECORD_REC_UNREAD
		sms_sdu	SMS_SDU_DELIVER_7CL2_43
(6)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	U_CP_DATA
		ti	TI_MT_FROM_MS
		cp_user_data_ul	RP_ACK_ULNK
		}	

(7) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(8) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	20-Dec-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.7.2 SMS201: PID, SC and OA okay, replace message

Description: The Relay Layer receives a mobile terminated short message. The protocol identifier indicates a replace short message. The message replaces the previous stored message for record 1 of the SIM card.

Preamble: SMS200

Variants: <A>...<C>

```
   MMI                                 SMS                               SIM/MM
     |                                  |                                  |
(1)  |                                  |       MMSMS_ESTABLISH_IND        |
     |                                  |            (CP_DATA)             |
     |                                  * <=============================== *
(2)  |                                  |          MMSMS_DATA_REQ          |
     |                                  |             (CP_ACK)             |
     |                                  * ===============================> *
(3)  |                                  |       SIM_READ_RECORD_REQ        |
     |                                  * ===============================> *
MUTE(500)
(4)  |                                  |       SIM_READ_RECORD_CNF        |
     |                                  * <=============================== *
(5)  |                                  |      SIM_UPDATE_RECORD_REQ       |
     |                                  * ===============================> *
MUTE(500)
(6)  |                                  |      SIM_UPDATE_RECORD_CNF       |
     |                                  * <=============================== *
(7)  |        MNSMS_MESSAGE_IND         |                                  |
     * <=============================== *                                  |
(8)  |                                  |          MMSMS_DATA_REQ          |
     |                                  |            (CP_DATA)             |
     |                                  * ===============================> *
(9)  |                                  |          MMSMS_DATA_IND          |
     |                                  |             (CP_ACK)             |
     |                                  * <=============================== *
(10) |                                  |        MMSMS_RELEASE_REQ         |
     |                                  * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                  |                                  |
```

Parametrization

Primitive		Parameter	Value
(1)	MMSMS_ESTABLISH_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	D_CP_DATA
		ti	TI_MT
	<A>	cp_user_data_dl	RP_DATA_DELIVER_7CL0_43
	<B>	cp_user_data_dl	RP_DATA_DELIVER_7CL1_43
	<C>	cp_user_data_dl	RP_DATA_DELIVER_7CL3_43
		}	
(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	B_CP_ACK
		ti	TI_MT_FROM_MS
		}	
(3)	SIM_READ_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
		record	SIM_RECORD_1
		length	LENGTH_SMS
(4)	SIM_READ_RECORD_CNF		
		datafield	SIM_SMS
		cause	SIM_NO_ERROR
		record	SIM_RECORD_1
		max_record	SIM_RECORD_3
		length	LENGTH_SMS
		linear_data	SIM_SMS_CLASS_2_43
(5)	SIM_UPDATE_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
		record	SIM_RECORD_1
		length	LENGTH_SMS
	<A>	linear_data	SIM_SMS_CLASS_0_43
	<B>	linear_data	SIM_SMS_CLASS_1_43
	<C>	linear_data	SIM_SMS_CLASS_3_43
(6)	SIM_UPDATE_RECORD_CNF		
		datafield	SIM_SMS
		record	SIM_RECORD_1
		cause	SIM_NO_ERROR

(7)	MNSMS_MESSAGE_IND		
		mem_type	MEM_SM
		rec_num	SIM_RECORD_1
		rec_max	SIM_RECORD_3
		status	SMS_RECORD_REC_UNREAD
	<A>	sms_sdu	SMS_SDU_DELIVER_7CL0_43
	<B>	sms_sdu	SMS_SDU_DELIVER_7CL1_43
	<C>	sms_sdu	SMS_SDU_DELIVER_7CL3_43
(8)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	U_CP_DATA
		ti	TI_MT_FROM_MS
		cp_user_data_ul	RP_ACK_ULNK
		}	
(9)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(10)		MMSMS_RELEASE_REQ	
		ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	10-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.7.3 SMS202: PID different, SC and OA okay, don´t replace message

Description: The Relay Layer receives a mobile terminated short message. The protocol identifier indicates a replace short message. The message will not replace the previous stored message for record 1 of the SIM card, because the protocol identifier is different.

Preamble: SMS200

```
   MMI                                   SMS                                  SIM/MM
    |                                     |                                     |
(1) |                                     |        MMSMS_ESTABLISH_IND          |
    |                                     |             (CP_DATA)               |
    |                                     * <================================== *
(2) |                                     |           MMSMS_DATA_REQ            |
    |                                     |              (CP_ACK)               |
    |                                     * ==================================> *
(3) |                                     |        SIM_UPDATE_RECORD_REQ        |
    |                                     * ==================================> *
(4) |                                     |        SIM_UPDATE_RECORD_CNF        |
    |                                     * <================================== *
(5) |          MNSMS_MESSAGE_IND          |                                     |
    * <================================== *                                     |
(6) |                                     |           MMSMS_DATA_REQ            |
    |                                     |             (CP_DATA)               |
    |                                     * ==================================> *
```

```
(7)  |                              |         MMSMS_DATA_IND          |
     |                              |            (CP_ACK)             |
     |                              * <=============================== *
(8)  |                              |        MMSMS_RELEASE_REQ        |
     |                              * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                              |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL1_42
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_1_42
(4) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_2
	cause	SIM_NO_ERROR
(5) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_DELIVER_7CL1_42

```
(6)   MMSMS_DATA_REQ
                                    d1                  NOT_USED
                                    d2                  NOT_USED
                                    sdu
                                    {
                                    component           SMS
                                    direction           UPLINK
                                    pd                  U_CP_DATA
                                    ti                  TI_MT_FROM_MS
                                    cp_user_data_ul     RP_ACK_ULNK
                                    }
(7)   MMSMS_DATA_IND
                                    d1                  NOT_USED
                                    d2                  NOT_USED
                                    sdu
                                    {
                                    component           SMS
                                    direction           DOWNLINK
                                    pd                  B_CP_ACK
                                    ti                  TI_MT
                                    }
(8)   MMSMS_RELEASE_REQ
                                    ti                  TI_MT_FROM_MS
```

History:	6-Jan-98	SZ	Initial
	10-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.7.4 SMS203: PID and OA okay, SC is different, replace message

Description: The Relay Layer receives a mobile terminated short message. The Protocol identifier indicates a replace short message. The message will replace the previous stored message for record 1 of the SIM card, because the service centre address is not to affect the replacement check.

Preamble: SMS200

```
   MMI                                SMS                              SIM/MM
    |                                  |                                  |
(1) |                                  |      MMSMS_ESTABLISH_IND         |
    |                                  |           (CP_DATA)              |
    |                                  * <=============================== *
(2) |                                  |         MMSMS_DATA_REQ           |
    |                                  |            (CP_ACK)              |
    |                                  * ===============================> *
(3) |                                  |      SIM_READ_RECORD_REQ         |
    |                                  * ===============================> *
(4) |                                  |      SIM_READ_RECORD_CNF         |
    |                                  * <=============================== *
(5) |                                  |     SIM_UPDATE_RECORD_REQ        |
    |                                  * ===============================> *
(6) |                                  |     SIM_UPDATE_RECORD_CNF        |
    |                                  * <=============================== *
(7) |       MNSMS_MESSAGE_IND          |                                  |
    * <=============================== *                                  |
(8) |                                  |         MMSMS_DATA_REQ           |
    |                                  |           (CP_DATA)              |
    |                                  * ===============================> *
(9) |                                  |         MMSMS_DATA_IND           |
    |                                  |            (CP_ACK)              |
    |                                  * <=============================== *
```

```
(10) |                                    |         MMSMS_RELEASE_REQ        |
     |                                    * ==============================> *
     |                                    |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL1_43S
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
(4) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_2_43
(5) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_1_43S
(6) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_NO_ERROR
(7) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_1
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_DELIVER_7CL1_43S

(8) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_FROM_MS
cp_user_data_ul	RP_ACK_ULNK
}	

(9) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(10) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	10-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.7.5 SMS204: PID and SC okay, OA is different, don´t replace message

Description: The Relay Layer receives a mobile terminated short message. The Protocol identifier indicates a replace short message. The message will not replace the previous stored message for record 1 of the SIM card, because the originator address is different.

Preamble: SMS200

```
   MMI                              SMS                             SIM/MM
     |                                |                                 |
(1)  |                                |      MMSMS_ESTABLISH_IND        |
     |                                |           (CP_DATA)             |
     |                                * <============================== *
(2)  |                                |        MMSMS_DATA_REQ           |
     |                                |            (CP_ACK)             |
     |                                * ==============================> *
(3)  |                                |      SIM_READ_RECORD_REQ        |
     |                                * ==============================> *
(4)  |                                |      SIM_READ_RECORD_CNF        |
     |                                * <============================== *
(5)  |                                |     SIM_UPDATE_RECORD_REQ       |
     |                                * ==============================> *
(6)  |                                |     SIM_UPDATE_RECORD_CNF       |
     |                                * <============================== *
(7)  |       MNSMS_MESSAGE_IND        |                                 |
     * <============================= *                                 |
(8)  |                                |        MMSMS_DATA_REQ           |
     |                                |           (CP_DATA)             |
     |                                * ==============================> *
(9)  |                                |        MMSMS_DATA_IND           |
     |                                |            (CP_ACK)             |
     |                                * <============================== *
```

```
(10) |                                    |         MMSMS_RELEASE_REQ         |
     |                                    * ================================> *
     |                                    |                                   |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL1_43O
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
(4) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_2_43
(5) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_1_43O
(6) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_2
	cause	SIM_NO_ERROR
(7) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_DELIVER_7CL1_43O

(8) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_FROM_MS
cp_user_data_ul	RP_ACK_ULNK
}	

(9) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(10) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	10-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.8 Mobile Terminated Short Message Status

3.8.1 SMS261: Mobile Terminated Short Message Status

Description: A status message has received. The message will be decoded by SMS and the parameters are forwarded to MMI.

Variants: <A>...<F>

Preamble:

<A>	SMS032A
<B>	SMS032B
<C>	SMS032C
<D>	SMS032D
<E>	SMS032E
<F>	SMS032F

```
   MMI                               SMS                              SIM/MM
    |                                 |                                  |
(1) |                                 |       MMSMS_ESTABLISH_IND        |
    |                                 |            (CP_DATA)             |
    |                                 * <=============================== *
(2) |                                 |          MMSMS_DATA_REQ          |
    |                                 |             (CP_ACK)             |
    |                                 * ===============================> *
(3) |         MNSMS_STATUS_IND        |                                  |
    * <=============================== *                                 |
(4) |                                 |          MMSMS_DATA_REQ          |
    |                                 |            (CP_DATA)             |
    |                                 * ===============================> *
(5) |                                 |          MMSMS_DATA_IND          |
    |                                 |             (CP_ACK)             |
    |                                 * <=============================== *
```

```
(6)  |                                   |         MMSMS_RELEASE_REQ         |
     |                                   * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                   |                                   |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_STATUS_REP
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) MNSMS_STATUS_IND		
	sms_sdu	SMS_SDU_STATUS_REP
(4) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ACK_ULNK
	}	
(5) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MT
	}	
(6) MMSMS_RELEASE_REQ		
	ti	TI_MT_FROM_MS

History:	6-Jan-98	SZ	Initial
	13-Dec-2000	LW	Adapted to new TAP
	11-Jan-2002	FK	Major rework

3.8.2 SMS262: Mobile Terminated Short Message Status, Phase 2+ handling

Description: A status message has received. The message will be decoded by SMS and the parameters are forwarded to ACI, which has to respond with a primitive MNSMS_ACK_RES.

Variants: <A>...<B>

Preamble: SMS033G

```
   MMI                               SMS                             SIM/MM
     |                                 |                                  |
(1)  |                                 |       MMSMS_ESTABLISH_IND        |
     |                                 |           (CP_DATA)              |
     |                                 * <=============================== *
(2)  |                                 |         MMSMS_DATA_REQ           |
     |                                 |            (CP_ACK)              |
     |                                 * ===============================> *
(3)  |       MNSMS_STATUS_IND          |                                  |
     * <============================== *                                  |
MUTE(2000)
(4)  |         MNSMS_ACK_RES           |                                  |
     * ==============================> *                                  |
(5)  |                                 |         MMSMS_DATA_REQ           |
     |                                 |           (CP_DATA)              |
     |                                 * ===============================> *
(6)  |                                 |         MMSMS_DATA_IND           |
     |                                 |            (CP_ACK)              |
     |                                 * <=============================== *
(7)  |                                 |       MMSMS_RELEASE_REQ          |
     |                                 * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                 |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_STATUS_REP
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	

(3)	MNSMS_STATUS_IND		
		sms_sdu	SMS_SDU_STATUS_REP
(4)	MNSMS_ACK_RES		
	<A>	resp	SMS_RP_ACK
	<B>	resp	SMS_RP_ERROR
	<A>	sms_sdu	SMS_SDU_EMPTY
	<B>	sms_sdu	SMS_SDU_EMPTY
(5)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	U_CP_DATA
		ti	TI_MT_FROM_MS
	<A>	cp_user_data_ul	RP_ACK_ULNK
	<B>	cp_user_data_ul	RP_ERR_ULNK_RESP
		}	
(6)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(7)	MMSMS_RELEASE_REQ		
		ti	TI_MT_FROM_MS

History:	23-Jan-2002	FK	Initial

3.9 One2One Operator-specific Short Message Handling

3.9.1 SMS311: Receiving SIM-specific Message

Description: According to operator specification a SIM-specific message shall be stored on SIM and subsequentedly be deleted without notification.
Pre-amble D checks the repeatability of the feature, Pre-amble E the error recovery.

Variants: <A>...<E>

Preamble:

<A>	SMS031N
<B>	SMS031N
<C>	SMS031N
<D>	SMS311A
<E>	SMS313A

```
   MMI                                      SMS                          SIM/MM
     |                                       |                              |
(1)  |                                       |     MMSMS_ESTABLISH_IND      |
     |                                       |          (CP_DATA)           |
     |                                       * <=========================== *
(2)  |                                       |        MMSMS_DATA_REQ        |
     |                                       |          (CP_ACK)            |
     |                                       * ===========================> *
```

```
(3)  |                                |      SIM_UPDATE_RECORD_REQ       |
     |                                * ===============================> *
MUTE(500)
(4)  |                                |      SIM_UPDATE_RECORD_CNF       |
     |                                * <=============================== *
(3)  |                                |      SIM_UPDATE_RECORD_REQ       |
     |                                * ===============================> *
MUTE(500)
(4)  |                                |      SIM_UPDATE_RECORD_CNF       |
     |                                * <=============================== *
(6)  |                                |         MMSMS_DATA_REQ           |
     |                                |            (CP_DATA)             |
     |                                * ===============================> *
(7)  |                                |         MMSMS_DATA_IND           |
     |                                |            (CP_ACK)              |
     |                                * <=============================== *
(8)  |                                |        MMSMS_RELEASE_REQ         |
     |                                * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
<A>	cp_user_data_dl	RP_DATA_DELIVER_121_A
<B>	cp_user_data_dl	RP_DATA_DELIVER_121_B
<C>	cp_user_data_dl	RP_DATA_DELIVER_121_C
<D>	cp_user_data_dl	RP_DATA_DELIVER_121_B
<E>	cp_user_data_dl	RP_DATA_DELIVER_121_C
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	

(3)	SIM_UPDATE_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
		record	SIM_RECORD_1
		length	LENGTH_SMS
	<A>	linear_data	SIM_SMS_DELIVER_121_A
	<B>	linear_data	SIM_SMS_DELIVER_121_B
	<C>	linear_data	SIM_SMS_DELIVER_121_C
	<D>	linear_data	SIM_SMS_DELIVER_121_B
	<E>	linear_data	SIM_SMS_DELIVER_121_C
(4)	SIM_UPDATE_RECORD_CNF		
		datafield	SIM_SMS
		record	SIM_RECORD_1
		cause	SIM_NO_ERROR
(5)	SIM_UPDATE_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
		record	SIM_RECORD_1
		length	LENGTH_SMS
		linear_data	SIM_SMS_EMPTY
(6)	SIM_UPDATE_RECORD_CNF		
		datafield	SIM_SMS
		record	SIM_RECORD_1
		cause	SIM_NO_ERROR
(7)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	U_CP_DATA
		ti	TI_MT_FROM_MS
		cp_user_data_ul	RP_ACK_ULNK
		}	
(8)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(9)	MMSMS_RELEASE_REQ		
		ti	TI_MT_FROM_MS

History:	23-Apr-2002	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.9.2 SMS312: Receiving SIM-specific Message, Memory Full Condition

Description: According to operator specification a SIM-specific message shall be stored on SIM and subsequentedly be deleted without notification. However, since SIM memory is full a 'Memory Capacity Exceeding' error is returned to the network

Variants: <A>...<C>

Preamble: SMS031O

```
   MMI                               SMS                             SIM/MM
    |                                 |                                |
(1) |                                 |      MMSMS_ESTABLISH_IND       |
    |                                 |           (CP_DATA)            |
    |                                 * <============================= *
(2) |                                 |         MMSMS_DATA_REQ         |
    |                                 |            (CP_ACK)            |
    |                                 * =============================> *
(6) |                                 |         MMSMS_DATA_REQ         |
    |                                 |           (CP_DATA)            |
    |                                 * =============================> *
(7) |                                 |         MMSMS_DATA_IND         |
    |                                 |            (CP_ACK)            |
    |                                 * <============================= *
(8) |                                 |       MMSMS_RELEASE_REQ        |
    |                                 * =============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                 |                                |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
<A>	cp_user_data_dl	RP_DATA_DELIVER_121_A
<B>	cp_user_data_dl	RP_DATA_DELIVER_121_B
<C>	cp_user_data_dl	RP_DATA_DELIVER_121_C
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ERR_MEM_CAP_EXC
	}	

(4) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(5) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History: 25-Apr-2002 FK Initial

3.9.3 SMS313: Receiving SIM-specific Message, Write Failure

Description: According to operator specification a SIM-specific message shall be stored on SIM and subsequentedly be deleted without notification.

Variants: <A>...<D>

Preamble:

<A>	SMS031N
<B>	SMS031N
<C>	SMS031N
<D>	SMS311A

```
   MMI                               SMS                             SIM/MM
     |                                |                                 |
(1)  |                                |      MMSMS_ESTABLISH_IND        |
     |                                |           (CP_DATA)             |
     |                                * <============================== *
(2)  |                                |         MMSMS_DATA_REQ          |
     |                                |            (CP_ACK)             |
     |                                * ==============================> *
(3)  |                                |     SIM_UPDATE_RECORD_REQ       |
     |                                * ==============================> *
MUTE(500)
(4)  |                                |     SIM_UPDATE_RECORD_CNF       |
     |                                * <============================== *
(5)  |                                |         SIM_UPDATE_REQ          |
     |                                * ==============================> *
(6)  |                                |         MMSMS_DATA_REQ          |
     |                                |           (CP_DATA)             |
     |                                * ==============================> *
(7)  |                                |         MMSMS_DATA_IND          |
     |                                |            (CP_ACK)             |
     |                                * <============================== *
(8)  |                                |       MMSMS_RELEASE_REQ         |
     |                                * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
（1） MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
<A>	cp_user_data_dl	RP_DATA_DELIVER_121_A
<B>	cp_user_data_dl	RP_DATA_DELIVER_121_B
<C>	cp_user_data_dl	RP_DATA_DELIVER_121_C
<D>	cp_user_data_dl	RP_DATA_DELIVER_121_B
	}	
（2） MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
（3） SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
<A>	linear_data	SIM_SMS_DELIVER_121_A
<B>	linear_data	SIM_SMS_DELIVER_121_B
<C>	linear_data	SIM_SMS_DELIVER_121_C
<D>	linear_data	SIM_SMS_DELIVER_121_B
（4） SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
<A>	cause	SIM_CAUSE_OTHER_ERROR
<B>	cause	SIM_CAUSE_ACCESS_PROHIBIT
<C>	cause	SIM_CAUSE_PUK1_BLOCKED
<D>	cause	SIM_CAUSE_ACCESS_PROHIBIT
（5） SIM_UPDATE_REQ		
	source	SRC_SMS
	offset	OFFSET_1
	datafield	SIM_SMSS
	length	SIM_SMSS_MEM_CAP_ONLY_LEN
	trans_data	SIM_SMSS_NO_MEM_CAP

(6) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_FROM_MS
cp_user_data_ul	RP_ERR_MEM_CAP_EXC
}	

(7) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(8) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	25-Apr-2002	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept
	10-Feb-2003	FK	SIM_UPDATE_REQ added

3.9.4 SMS314: Receiving Empty Message

Description: According to operator specification the Empty Message is used by the network to determine the SIM memory state. The message is neither stored nor displayed, but is acknowledged to the network, when memory is available (Variant A), or rejected with cause "Memory capacity exceeded, when memory is full (Variant B).

Variants: <A>...<B>

Preamble:

<A> SMS031N

<B> SMS031O

```
   MMI                                 SMS                              SIM/MM
     |                                  |                                  |
(1)  |                                  |       MMSMS_ESTABLISH_IND        |
     |                                  |            (CP_DATA)             |
     |                                  * <=============================== *
(2)  |                                  |          MMSMS_DATA_REQ          |
     |                                  |             (CP_ACK)             |
     |                                  * ===============================> *
(6)  |                                  |          MMSMS_DATA_REQ          |
     |                                  |            (CP_DATA)             |
     |                                  * ===============================> *
(7)  |                                  |          MMSMS_DATA_IND          |
     |                                  |             (CP_ACK)             |
     |                                  * <=============================== *
(8)  |                                  |        MMSMS_RELEASE_REQ         |
     |                                  * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                  |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	MMSMS_ESTABLISH_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	D_CP_DATA
		ti	TI_MT
		cp_user_data_dl	RP_DATA_DELIVER_EMPTY
		}	
(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	B_CP_ACK
		ti	TI_MT_FROM_MS
		}	
(3)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	U_CP_DATA
		ti	TI_MT_FROM_MS
	<A>	cp_user_data_ul	RP_ACK_ULNK
	<B>	cp_user_data_ul	RP_ERR_MEM_CAP_EXC
		}	
(4)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(5)	MMSMS_RELEASE_REQ		
		ti	TI_MT_FROM_MS

History: 26-Apr-2002 FK Initial

3.10 Short Message Service Procedures

3.10.1 SMS401: Storing of Short Messages (any record)

Description: It is possible to store mobile originated messages on the SIM card. Storing of new messages is indicated with the record number zero. The positive report indication contains the used record number.

Variants: <A>...<J>

Preamble:

<A>	SMS031A
<B>	SMS031B
<C>	SMS031C
<D>	SMS031D
<E>	SMS031E
<F>	SMS031F
<G>	SMS031I
<H>	SMS031I
<I>	SMS031I
<J>	SMS031I

```
   MMI                               SMS                               SIM/MM
    |                                 |                                 |
(1) |         MNSMS_STORE_REQ         |                                 |
    * ==============================> *                                 |
(2) |                                 |      SIM_UPDATE_RECORD_REQ      |
    |                                 * ==============================> *
MUTE(500)
(3) |                                 |      SIM_UPDATE_RECORD_CNF      |
    |                                 * <============================== *
(4) |         MNSMS_STORE_CNF         |                                 |
    * <============================== *                                 |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                 |                                 |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_STORE_REQ		
		mem_type	MEM_SM
		rec_num	SIM_RECORD_0
	<A>	condx	SMS_CONDX_OVR_NON
	<B>	condx	SMS_CONDX_OVR_NON
	<C>	condx	SMS_CONDX_OVR_NON
	<D>	condx	SMS_CONDX_OVR_NON
	<E>	condx	SMS_CONDX_OVR_NON
	<F>	condx	SMS_CONDX_OVR_NON
	<G>	condx	SMS_CONDX_OVR_NON
	<H>	condx	SMS_CONDX_OVR_MO
	<I>	condx	SMS_CONDX_OVR_ANY
	<J>	condx	NOT_PRESENT_8BIT
		status	SMS_RECORD_STO_UNSENT
	<A>	sms_sdu	SMS_SDU_SBM_DEF
	<B>	sms_sdu	SMS_SDU_SBM_DEF_X
	<C>	sms_sdu	SMS_SDU_SBM_DEF_X
	<D>	sms_sdu	SMS_SDU_SBM_DEF
	<E>	sms_sdu	SMS_SDU_SBM_DEF
	<F>	sms_sdu	SMS_SDU_SBM_DEF
	<G>	sms_sdu	SMS_SDU_SBM_DEF
	<H>	sms_sdu	SMS_SDU_SBM_DEF
	<I>	sms_sdu	SMS_SDU_SBM_DEF
	<J>	sms_sdu	SMS_SDU_SBM_DEF

(2)	SIM_UPDATE_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_1
	<B>	record	SIM_RECORD_1
	<C>	record	SIM_RECORD_1
	<D>	record	SIM_RECORD_1
	<E>	record	SIM_RECORD_1
	<F>	record	SIM_RECORD_1
	<G>	record	SIM_RECORD_2
	<H>	record	SIM_RECORD_2
	<I>	record	SIM_RECORD_2
	<J>	record	SIM_RECORD_2
		length	LENGTH_SMS
		linear_data	SIM_SMS_SBM_DEF
(3)	SIM_UPDATE_RECORD_CNF		
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_1
	<B>	record	SIM_RECORD_1
	<C>	record	SIM_RECORD_1
	<D>	record	SIM_RECORD_1
	<E>	record	SIM_RECORD_1
	<F>	record	SIM_RECORD_1
	<G>	record	SIM_RECORD_2
	<H>	record	SIM_RECORD_2
	<I>	record	SIM_RECORD_2
	<J>	record	SIM_RECORD_2
		cause	SIM_NO_ERROR
(4)	MNSMS_STORE_CNF		
		mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_1
	<B>	rec_num	SIM_RECORD_1
	<C>	rec_num	SIM_RECORD_1
	<D>	rec_num	SIM_RECORD_1
	<E>	rec_num	SIM_RECORD_1
	<F>	rec_num	SIM_RECORD_1
	<G>	rec_num	SIM_RECORD_2
	<H>	rec_num	SIM_RECORD_2
	<I>	rec_num	SIM_RECORD_2
	<J>	rec_num	SIM_RECORD_2
		cause	SIM_NO_ERROR

History:	6-Jan-98	SZ	Initial
	13-Apr-2000	FK	Primitive change
	11-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.2 SMS402: Storing of Short Messages (given record)

Description: It is possible to store mobile originated messages on the SIM card. The record for storing the new message is given by the request, therefore the parameters 'condx' has to be observed.

Variants: <A>...<J>

Preamble:

<A>	SMS031C
<B>	SMS031C
<C>	SMS031C
<D>	SMS401C
<E>	SMS401C
<F>	SMS401C

<G>	SMS031I
<H>	SMS031I
<I>	SMS031I
<J>	SMS031I

```
   MMI                                SMS                                SIM/MM
    |                                  |                                  |
(1) |         MNSMS_STORE_REQ          |                                  |
    * ===============================> *                                  |
(2) |                                  |      SIM_UPDATE_RECORD_REQ       |
    |                                  * ===============================> *
MUTE(500)
(3) |                                  |      SIM_UPDATE_RECORD_CNF       |
    |                                  * <=============================== *
(4) |         MNSMS_STORE_CNF          |                                  |
    * <=============================== *                                  |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                  |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_STORE_REQ		
		mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_1
	<B>	rec_num	SIM_RECORD_2
	<C>	rec_num	SIM_RECORD_3
	<D>	rec_num	SIM_RECORD_1
	<E>	rec_num	SIM_RECORD_1
	<F>	rec_num	SIM_RECORD_1
	<G>	rec_num	SIM_RECORD_2
	<H>	rec_num	SIM_RECORD_1
	<I>	rec_num	SIM_RECORD_3
	<J>	rec_num	SIM_RECORD_1
	<A>	condx	SMS_CONDX_OVR_NON
	<B>	condx	SMS_CONDX_OVR_NON
	<C>	condx	SMS_CONDX_OVR_NON
	<D>	condx	SMS_CONDX_OVR_MO
	<E>	condx	SMS_CONDX_OVR_ANY
	<F>	condx	SMS_CONDX_OVR_UNDEF
	<G>	condx	SMS_CONDX_OVR_NON
	<H>	condx	SMS_CONDX_OVR_MO
	<I>	condx	SMS_CONDX_OVR_ANY
	<J>	condx	NOT_PRESENT_8BIT
		status	SMS_RECORD_STO_UNSENT
		sms_sdu	SMS_SDU_SBM_DEF

(2) SIM_UPDATE_RECORD_REQ

	source	SRC_SMS
	datafield	SIM_SMS
<A>	record	SIM_RECORD_1
<B>	record	SIM_RECORD_2
<C>	record	SIM_RECORD_3
<D>	record	SIM_RECORD_1
<E>	record	SIM_RECORD_1
<F>	record	SIM_RECORD_1
<G>	record	SIM_RECORD_2
<H>	record	SIM_RECORD_1
<I>	record	SIM_RECORD_3
<J>	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_SBM_DEF

(3) SIM_UPDATE_RECORD_CNF

	datafield	SIM_SMS
<A>	record	SIM_RECORD_1
<B>	record	SIM_RECORD_2
<C>	record	SIM_RECORD_3
<D>	record	SIM_RECORD_1
<E>	record	SIM_RECORD_1
<F>	record	SIM_RECORD_1
<G>	record	SIM_RECORD_2
<H>	record	SIM_RECORD_1
<I>	record	SIM_RECORD_3
<J>	record	SIM_RECORD_1
	cause	SIM_NO_ERROR

(4) MNSMS_STORE_CNF

	mem_type	MEM_SM
<A>	rec_num	SIM_RECORD_1
<B>	rec_num	SIM_RECORD_2
<C>	rec_num	SIM_RECORD_3
<D>	rec_num	SIM_RECORD_1
<E>	rec_num	SIM_RECORD_1
<F>	rec_num	SIM_RECORD_1
<G>	rec_num	SIM_RECORD_2
<H>	rec_num	SIM_RECORD_1
<I>	rec_num	SIM_RECORD_3
<J>	rec_num	SIM_RECORD_1
	cause	SIM_NO_ERROR

History:	11-Jan-2001	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.3 SMS403: Storing of Short Messages, failure cases

Description: The MMI requests storing of a short message. This is not possible for different reasons. A failure cause shall be returned by SMS.
Variant A: record number too high
Variant B: ME memory is not available.
Variant C: SR memory is not available.
Variant D: invalid parameter 'mem_type'.
Variant E, F: overwriting of message not set for MO-SM.
Variant G,H: overwriting of message not set for MT-SM.
Variant I: overwriting of message not set, memory is full.

Variants: <A>...<I>

Preamble:

<A> SMS012A

<B>	SMS012A
<C>	SMS012A
<D>	SMS012A
<E>	SMS013
<F>	SMS013
<G>	SMS014
<H>	SMS014
<I>	SMS016A

```
   MMI                               SMS                             SIM/MM
     |                                |                                 |
(1)  |        MNSMS_STORE_REQ         |                                 |
     * =============================> *                                 |
(2)  |        MNSMS_STORE_CNF         |                                 |
     * <============================= *                                 |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                 |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_STORE_REQ		
	<A>	mem_type	MEM_SM
	<B>	mem_type	MEM_ME
	<C>	mem_type	MEM_SR
	<D>	mem_type	BYTE_00
	<E>	mem_type	MEM_SM
	<F>	mem_type	MEM_SM
	<G>	mem_type	MEM_SM
	<H>	mem_type	MEM_SM
	<I>	mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_5
	<B>	rec_num	SIM_RECORD_2
	<C>	rec_num	SIM_RECORD_1
	<D>	rec_num	SIM_RECORD_1
	<E>	rec_num	SIM_RECORD_2
	<F>	rec_num	SIM_RECORD_3
	<G>	rec_num	SIM_RECORD_2
	<H>	rec_num	SIM_RECORD_2
	<I>	rec_num	SIM_RECORD_0
	<A>	condx	SMS_CONDX_OVR_NON
	<B>	condx	SMS_CONDX_OVR_NON
	<C>	condx	SMS_CONDX_OVR_NON
	<D>	condx	SMS_CONDX_OVR_NON
	<E>	condx	SMS_CONDX_OVR_NON
	<F>	condx	SMS_CONDX_OVR_NON
	<G>	condx	SMS_CONDX_OVR_NON
	<H>	condx	SMS_CONDX_OVR_MO
	<I>	condx	SMS_CONDX_OVR_NON
		status	SMS_RECORD_STO_UNSENT
		sms_sdu	SMS_SDU_SBM_DEF

(2) MNSMS_STORE_CNF

<A>	mem_type	MEM_SM
<B>	mem_type	MEM_ME
<C>	mem_type	MEM_SR
<D>	mem_type	BYTE_00
<E>	mem_type	MEM_SM
<F>	mem_type	MEM_SM
<G>	mem_type	MEM_SM
<H>	mem_type	MEM_SM
<I>	mem_type	MEM_SM
<A>	rec_num	SIM_RECORD_5
<B>	rec_num	SIM_RECORD_2
<C>	rec_num	SIM_RECORD_1
<D>	rec_num	SIM_RECORD_1
<E>	rec_num	SIM_RECORD_2
<F>	rec_num	SIM_RECORD_3
<G>	rec_num	SIM_RECORD_2
<H>	rec_num	SIM_RECORD_2
<I>	rec_num	SIM_RECORD_0
<A>	cause	SMS_CAUSE_INV_INDEX
<B>	cause	SMS_CAUSE_MEM_FAIL
<C>	cause	SMS_CAUSE_MEM_FAIL
<D>	cause	SMS_CAUSE_PARAM_WRONG
<E>	cause	SMS_CAUSE_INV_INDEX
<F>	cause	SMS_CAUSE_INV_INDEX
<G>	cause	SMS_CAUSE_INV_INDEX
<H>	cause	SMS_CAUSE_INV_INDEX
<I>	cause	SMS_CAUSE_MEM_FULL

History:	14-Oct-98	LE	Initial
	13-Apr-2000	FK	Primitive change
	25-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.4 SMS411: Changing of Short Messages

Description: It is possible to change stored messages on the SIM card. Changing of new messages is indicated with the previous record number.

Variants: <A>...<J>

Preamble:

<A>	SMS031I
<B>	SMS031I
<C>	SMS031I
<D>	SMS031I
<E>	SMS031I
<F>	SMS031J
<G>	SMS031J
<H>	SMS401G
<I>	SMS401G
<J>	SMS401G

```
   MMI                                SMS                                SIM/MM
    |                                  |                                   |
(1) |          MNSMS_STORE_REQ         |                                   |
    * ===============================> *                                   |
(2) |                                  |       SIM_UPDATE_RECORD_REQ       |
    |                                  * ===============================>  *
MUTE(500)
(3) |                                  |       SIM_UPDATE_RECORD_CNF       |
    |                                  * <===============================  *
```

```
(4)  |          MNSMS_STORE_CNF           |                                       |
     * <============================== *                                       |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                    |                                       |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_STORE_REQ		
	mem_type	MEM_SM
<A>	rec_num	SIM_RECORD_1
<B>	rec_num	SIM_RECORD_1
<C>	rec_num	SIM_RECORD_1
<D>	rec_num	SIM_RECORD_3
<E>	rec_num	SIM_RECORD_3
<F>	rec_num	SIM_RECORD_2
<G>	rec_num	SIM_RECORD_2
<H>	rec_num	SIM_RECORD_2
<I>	rec_num	SIM_RECORD_2
<J>	rec_num	SIM_RECORD_2
<A>	condx	SMS_CONDX_OVR_MO
<B>	condx	SMS_CONDX_OVR_ANY
<C>	condx	NOT_PRESENT_8BIT
<D>	condx	SMS_CONDX_OVR_ANY
<E>	condx	NOT_PRESENT_8BIT
<F>	condx	SMS_CONDX_OVR_ANY
<G>	condx	NOT_PRESENT_8BIT
<H>	condx	SMS_CONDX_OVR_MO
<I>	condx	SMS_CONDX_OVR_ANY
<J>	condx	NOT_PRESENT_8BIT
	status	SMS_RECORD_STO_UNSENT
	sms_sdu	SMS_SDU_SBM_DEF
(2) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
<A>	record	SIM_RECORD_1
<B>	record	SIM_RECORD_1
<C>	record	SIM_RECORD_1
<D>	record	SIM_RECORD_3
<E>	record	SIM_RECORD_3
<F>	record	SIM_RECORD_2
<G>	record	SIM_RECORD_2
<H>	record	SIM_RECORD_2
<I>	record	SIM_RECORD_2
<J>	record	SIM_RECORD_2
	length	LENGTH_SMS
	linear_data	SIM_SMS_SBM_DEF

(3)	SIM_UPDATE_RECORD_CNF		
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_1
	<B>	record	SIM_RECORD_1
	<C>	record	SIM_RECORD_1
	<D>	record	SIM_RECORD_3
	<E>	record	SIM_RECORD_3
	<F>	record	SIM_RECORD_2
	<G>	record	SIM_RECORD_2
	<H>	record	SIM_RECORD_2
	<I>	record	SIM_RECORD_2
	<J>	record	SIM_RECORD_2
		cause	SIM_NO_ERROR

(4)	MNSMS_STORE_CNF		
		mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_1
	<B>	rec_num	SIM_RECORD_1
	<C>	rec_num	SIM_RECORD_1
	<D>	rec_num	SIM_RECORD_3
	<E>	rec_num	SIM_RECORD_3
	<F>	rec_num	SIM_RECORD_2
	<G>	rec_num	SIM_RECORD_2
	<H>	rec_num	SIM_RECORD_2
	<I>	rec_num	SIM_RECORD_2
	<J>	rec_num	SIM_RECORD_2
		cause	SIM_NO_ERROR

History:	6-Jan-98	SZ	Initial
	13-Apr-2000	FK	Primitive change
	14-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.5 SMS421: Reading of Short Messages, Mobile originated, not read

Description: In the preamble a mobile originated short message was indicated by SMS. This message is read from the MMI.

Variants: <A>...<D>

Preamble: SMS013

```
   MMI                               SMS                               SIM/MM
    |                                 |                                 |
(1) |          MNSMS_READ_REQ         |                                 |
    * ==============================> *                                 |
(2) |                                 |       SIM_READ_RECORD_REQ       |
    |                                 * ==============================> *
MUTE(500)
(3) |                                 |       SIM_READ_RECORD_CNF       |
    |                                 * <============================== *
(4) |          MNSMS_READ_CNF         |                                 |
    * <============================== *                                 |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                 |                                 |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_READ_REQ	mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_2
	<B>	rec_num	SIM_RECORD_2
	<C>	rec_num	SIM_RECORD_3
	<D>	rec_num	SIM_RECORD_3
		read_mode	READ_NORMAL
	<A>	status	BYTE_00
	<B>	status	NOT_PRESENT_8BIT
	<C>	status	BYTE_00
	<D>	status	NOT_PRESENT_8BIT
(2)	SIM_READ_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_2
	<B>	record	SIM_RECORD_2
	<C>	record	SIM_RECORD_3
	<D>	record	SIM_RECORD_3
		length	LENGTH_SMS
(3)	SIM_READ_RECORD_CNF		
		datafield	SIM_SMS
		cause	SIM_NO_ERROR
	<A>	record	SIM_RECORD_2
	<B>	record	SIM_RECORD_2
	<C>	record	SIM_RECORD_3
	<D>	record	SIM_RECORD_3
		max_record	SIM_RECORD_3
		length	LENGTH_SMS
	<A>	linear_data	SIM_SMS_MO
	<B>	linear_data	SIM_SMS_MO
	<C>	linear_data	SIM_SMS_MO_ABS
	<D>	linear_data	SIM_SMS_MO_ABS
(4)	MNSMS_READ_CNF		
		mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_2
	<B>	rec_num	SIM_RECORD_2
	<C>	rec_num	SIM_RECORD_3
	<D>	rec_num	SIM_RECORD_3
	<A>	rec_next	SIM_RECORD_3
	<B>	rec_next	SIM_RECORD_3
	<C>	rec_next	SMS_RECORD_NOT_EXIST
	<D>	rec_next	SMS_RECORD_NOT_EXIST
		rec_max	SIM_RECORD_3
		cause	SIM_NO_ERROR
		rec_status	SMS_RECORD_NOT_EXIST
	<A>	status	SMS_RECORD_STO_UNSENT
	<B>	status	SMS_RECORD_STO_UNSENT
	<C>	status	SMS_RECORD_STO_SENT
	<D>	status	SMS_RECORD_STO_SENT
	<A>	sms_sdu	SMS_SDU_MO
	<B>	sms_sdu	SMS_SDU_MO
	<C>	sms_sdu	SMS_SDU_MO_ABS
	<D>	sms_sdu	SMS_SDU_MO_ABS

History:	13-Oct-98	LE	Initial
	13-Apr-2000	FK	Primitive change
	14-Apr-2000	FK	No status change on MO-SM

15-Jan-2001	FK	Major rework
25-Oct-2002	FK	Adaption to Cause Concept

3.10.6 SMS422: Reading of Short Messages, Mobile originated, already read

Description: In the preamble a mobile originated short message was read. This message is reread that it was not modified by the previous read command.

Variants: <A>...<B>

Preamble:

<A>	SMS421A
<B>	SMS421D

```
   MMI                                 SMS                               SIM/MM
     |                                  |                                  |
(1)  |          MNSMS_READ_REQ          |                                  |
     * ===============================> *                                  |
(2)  |                                  |       SIM_READ_RECORD_REQ        |
     |                                  * ===============================> *
MUTE(500)
(3)  |                                  |       SIM_READ_RECORD_CNF        |
     |                                  * <=============================== *
(4)  |          MNSMS_READ_CNF          |                                  |
     * <=============================== *                                  |
MUTE(500)
     |                                  |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_READ_REQ	mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_2
	<B>	rec_num	SIM_RECORD_3
		read_mode	READ_NORMAL
	<A>	status	BYTE_00
	<B>	status	NOT_PRESENT_8BIT
(2)	SIM_READ_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_2
	<B>	record	SIM_RECORD_3
		length	LENGTH_SMS
(3)	SIM_READ_RECORD_CNF		
		datafield	SIM_SMS
		cause	SIM_NO_ERROR
	<A>	record	SIM_RECORD_2
	<B>	record	SIM_RECORD_3
		max_record	SIM_RECORD_3
		length	LENGTH_SMS
	<A>	linear_data	SIM_SMS_MO
	<B>	linear_data	SIM_SMS_MO_ABS

		Parameter	Value
(4)	MNSMS_READ_CNF		
		mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_2
	<B>	rec_num	SIM_RECORD_3
	<A>	rec_next	SIM_RECORD_3
	<B>	rec_next	SMS_RECORD_NOT_EXIST
		rec_max	SIM_RECORD_3
		cause	SIM_NO_ERROR
		rec_status	SMS_RECORD_NOT_EXIST
	<A>	status	SMS_RECORD_STO_UNSENT
	<B>	status	SMS_RECORD_STO_SENT
	<A>	sms_sdu	SMS_SDU_MO
	<B>	sms_sdu	SMS_SDU_MO_ABS

History:	13-Oct-98	LE	Initial
	13-Apr-2000	FK	Primitive change
	15-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.7 SMS423: Reading of Short Messages, Mobile terminated, not read

Description: In the preamble a mobile terminated short message was indicated by SMS. This message is read from the MMI.

Preamble: SMS014

```
   MMI                                 SMS                               SIM/MM
     |                                  |                                   |
(1)  |         MNSMS_READ_REQ           |                                   |
     * ===============================> *                                   |
(2)  |                                  |       SIM_READ_RECORD_REQ         |
     |                                  * ================================> *
MUTE(500)
(3)  |                                  |       SIM_READ_RECORD_CNF         |
     |                                  * <================================ *
(4)  |                                  |      SIM_UPDATE_RECORD_REQ        |
     |                                  * ================================> *
MUTE(500)
(5)  |                                  |      SIM_UPDATE_RECORD_CNF        |
     |                                  * <================================ *
(6)  |         MNSMS_READ_CNF           |                                   |
     * <=============================== *                                   |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                  |                                   |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_READ_REQ	mem_type	MEM_SM
		rec_num	SIM_RECORD_2
		read_mode	READ_NORMAL
		status	BYTE_00
(2)	SIM_READ_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
		record	SIM_RECORD_2
		length	LENGTH_SMS

(3) SIM_READ_RECORD_CNF

datafield	SIM_SMS
cause	SIM_NO_ERROR
record	SIM_RECORD_2
max_record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_MT

(4) SIM_UPDATE_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_2
length	LENGTH_SMS
linear_data	SIM_SMS_MT_READ

(5) SIM_UPDATE_RECORD_CNF

datafield	SIM_SMS
record	SIM_RECORD_2
cause	SIM_NO_ERROR

(6) MNSMS_READ_CNF

mem_type	MEM_SM
rec_num	SIM_RECORD_2
rec_next	SMS_RECORD_NOT_EXIST
rec_max	SIM_RECORD_3
cause	SIM_NO_ERROR
rec_status	SMS_RECORD_NOT_EXIST
status	SMS_RECORD_REC_UNREAD
sms_sdu	SMS_SDU_MT

History:	13-Oct-98	LE	Initial
	11-Jan-2001	LW	MT IND now after UPDATE CNF (patch by RIE 12/19/00)
	15-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.8 SMS424: Reading of Short Messages, Mobile terminated, already read

Description: In the preamble a mobile terminated short message was indicated by SMS. This message is read from the MMI.

Variants: <A>...<B>

Preamble:

<A> SMS016A
<B> SMS423

```
   MMI                                SMS                               SIM/MM
     |                                  |                                  |
(1)  |         MNSMS_READ_REQ           |                                  |
     * ===============================> *                                  |
(2)  |                                  |      SIM_READ_RECORD_REQ         |
     |                                  * ===============================> *
MUTE(500)
(3)  |                                  |      SIM_READ_RECORD_CNF         |
     |                                  * <=============================== *
(4)  |         MNSMS_READ_CNF           |                                  |
     * <=============================== *                                  |
MUTE(500)
     |                                  |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_READ_REQ	mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_3
	<B>	rec_num	SIM_RECORD_2
		read_mode	READ_NORMAL
		status	BYTE_00
(2)	SIM_READ_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_3
	<B>	record	SIM_RECORD_2
		length	LENGTH_SMS
(3)	SIM_READ_RECORD_CNF		
		datafield	SIM_SMS
		cause	SIM_NO_ERROR
	<A>	record	SIM_RECORD_3
	<B>	record	SIM_RECORD_2
		max_record	SIM_RECORD_3
		length	LENGTH_SMS
		linear_data	SIM_SMS_MT_READ
(4)	MNSMS_READ_CNF		
		mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_3
	<B>	rec_num	SIM_RECORD_2
		rec_next	SMS_RECORD_NOT_EXIST
		rec_max	SIM_RECORD_3
		cause	SIM_NO_ERROR
		rec_status	SMS_RECORD_NOT_EXIST
		status	SMS_RECORD_REC_READ
		sms_sdu	SMS_SDU_MT

History:	13-Oct-98	LE	Initial
	15-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.9 SMS425: Reading of Short Messages, failure cases

Description: The MMI requests reading of a short message. This is not possible for different reasons. A failure cause shall be returned by SMS.
Variant A: record number 0, memory is empty
Variant B: record number too high
Variant C: record is unused
Variant D: ME memory is not available.
Variant E: SR memory is not available
Variant F: invalid parameter 'mem_type'
Variant G: Reading of record 1 holding an unsupported TPDU (SMS-Command)
Variant H: Reading of record 3 holding an unsupported TPDU (SMS-Status-Report)

Variants: <A>...<H>

Preamble:

<A>	SMS012A
<B>	SMS015
<C>	SMS015
<D>	SMS015
<E>	SMS015
<F>	SMS015
<G>	SMS021
<H>	SMS021

```
   MMI                            SMS                            SIM/MM
    |                              |                               |
(1) |         MNSMS_READ_REQ       |                               |
    * ============================> *                               |
(2) |         MNSMS_READ_CNF       |                               |
    * <============================ *                               |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                              |                               |
```

Parametrization

Primitive		Parameter	Value
(1)	MNSMS_READ_REQ		
	<A>	mem_type	MEM_SM
	<B>	mem_type	MEM_SM
	<C>	mem_type	MEM_SM
	<D>	mem_type	MEM_ME
	<E>	mem_type	MEM_SR
	<F>	mem_type	BYTE_00
	<G>	mem_type	MEM_SM
	<H>	mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_0
	<B>	rec_num	SIM_RECORD_5
	<C>	rec_num	SIM_RECORD_2
	<D>	rec_num	SIM_RECORD_1
	<E>	rec_num	SIM_RECORD_1
	<F>	rec_num	SIM_RECORD_1
	<G>	rec_num	SIM_RECORD_1
	<H>	rec_num	SIM_RECORD_3
		read_mode	READ_NORMAL
		status	BYTE_00

（2） MNSMS_READ_CNF

<A>	mem_type	MEM_SM
<B>	mem_type	MEM_SM
<C>	mem_type	MEM_SM
<D>	mem_type	MEM_ME
<E>	mem_type	MEM_SR
<F>	mem_type	BYTE_00
<G>	mem_type	MEM_SM
<H>	mem_type	MEM_SM
<A>	rec_num	SIM_RECORD_0
<B>	rec_num	SIM_RECORD_5
<C>	rec_num	SIM_RECORD_2
<D>	rec_num	SIM_RECORD_1
<E>	rec_num	SIM_RECORD_1
<F>	rec_num	SIM_RECORD_1
<G>	rec_num	SIM_RECORD_1
<H>	rec_num	SIM_RECORD_3
<A>	rec_next	SMS_RECORD_NOT_EXIST
<B>	rec_next	SMS_RECORD_NOT_EXIST
<C>	rec_next	SIM_RECORD_3
<D>	rec_next	SMS_RECORD_NOT_EXIST
<E>	rec_next	SMS_RECORD_NOT_EXIST
<F>	rec_next	SMS_RECORD_NOT_EXIST
<G>	rec_next	SMS_RECORD_NOT_EXIST
<H>	rec_next	SMS_RECORD_NOT_EXIST
<A>	rec_max	SIM_RECORD_3
<B>	rec_max	SIM_RECORD_3
<C>	rec_max	SIM_RECORD_3
<D>	rec_max	SMS_RECORD_NOT_EXIST
<E>	rec_max	SMS_RECORD_NOT_EXIST
<F>	rec_max	SMS_RECORD_NOT_EXIST
<G>	rec_max	SIM_RECORD_3
<H>	rec_max	SIM_RECORD_3
<A>	cause	SMS_CAUSE_INV_INDEX
<B>	cause	SMS_CAUSE_INV_INDEX
<C>	cause	SMS_CAUSE_INV_INDEX
<D>	cause	SMS_CAUSE_MEM_FAIL
<E>	cause	SMS_CAUSE_MEM_FAIL
<F>	cause	SMS_CAUSE_PARAM_WRONG
<G>	cause	SMS_CAUSE_INV_INDEX
<H>	cause	SMS_CAUSE_INV_INDEX
	rec_status	SMS_RECORD_NOT_EXIST
	status	SMS_RECORD_FREE
	sms_sdu	SMS_SDU_EMPTY

History:	14-Oct-98	LE	Initial
	30-May-2000	FK	Unrecognized messages added
	25-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.10 SMS426: Searching for Short Messages, any Message

Description: The first Short Message with the property given by the parameter 'status' will be retrieved from SIM.

Variants: <A>...<E>

Preamble:

<A>	SMS013
<B>	SMS014
<C>	SMS015
<D>	SMS161A
<E>	SMS401G

```
   MMI                                SMS                                SIM/MM
    |                                  |                                  |
(1) |         MNSMS_READ_REQ           |                                  |
    * ==============================> *                                  |
(2) |                                  |       SIM_READ_RECORD_REQ        |
    |                                  * ==============================> *
MUTE(500)
(3) |                                  |       SIM_READ_RECORD_CNF        |
    |                                  * <============================== *
(4) |         MNSMS_READ_CNF           |                                  |
    * <============================== *                                  |
MUTE(500)
    |                                  |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_READ_REQ	mem_type	MEM_SM
		rec_num	SIM_RECORD_0
		read_mode	READ_PREVIEW
		status	BYTE_00
(2)	SIM_READ_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_2
	<B>	record	SIM_RECORD_2
	<C>	record	SIM_RECORD_1
	<D>	record	SIM_RECORD_1
	<E>	record	SIM_RECORD_1
		length	LENGTH_SMS
(3)	SIM_READ_RECORD_CNF		
		datafield	SIM_SMS
		cause	SIM_NO_ERROR
	<A>	record	SIM_RECORD_2
	<B>	record	SIM_RECORD_2
	<C>	record	SIM_RECORD_1
	<D>	record	SIM_RECORD_1
	<E>	record	SIM_RECORD_1
		max_record	SIM_RECORD_3
		length	LENGTH_SMS
	<A>	linear_data	SIM_SMS_MO
	<B>	linear_data	SIM_SMS_MT
	<C>	linear_data	SIM_SMS_MO
	<D>	linear_data	SIM_SMS_CLASS_1
	<E>	linear_data	SIM_SMS_MO

(4) MNSMS_READ_CNF

	mem_type	MEM_SM
<A>	rec_num	SIM_RECORD_2
<B>	rec_num	SIM_RECORD_2
<C>	rec_num	SIM_RECORD_1
<D>	rec_num	SIM_RECORD_1
<E>	rec_num	SIM_RECORD_1
<A>	rec_next	SIM_RECORD_3
<B>	rec_next	SMS_RECORD_NOT_EXIST
<C>	rec_next	SIM_RECORD_3
<D>	rec_next	SMS_RECORD_NOT_EXIST
<E>	rec_next	SIM_RECORD_2
	rec_max	SIM_RECORD_3
	cause	SIM_NO_ERROR
	rec_status	SMS_RECORD_NOT_EXIST
<A>	status	SMS_RECORD_STO_UNSENT
<B>	status	SMS_RECORD_REC_UNREAD
<C>	status	SMS_RECORD_STO_UNSENT
<D>	status	SMS_RECORD_REC_UNREAD
<E>	status	SMS_RECORD_STO_UNSENT
<A>	sms_sdu	SMS_SDU_MO
<B>	sms_sdu	SMS_SDU_MT
<C>	sms_sdu	SMS_SDU_MO
<D>	sms_sdu	SMS_SDU_MT_7CL1
<E>	sms_sdu	SMS_SDU_MO

History:	29-Feb-2000	FK	Initial
	12-Dec-2000	LW	Adaption to new TAP
	15-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.11 SMS427: Searching for Short Messages, Mobile Originated, not Sent

Description: The first Short Message with the property given by the parameter 'status' will be retrieved from SIM.

Variants: <A>...<E>

Preamble:

<A>	SMS013
<B>	SMS015
<C>	SMS016A
<D>	SMS401C
<E>	SMS401G

```
   MMI                              SMS                             SIM/MM
     |                                |                                |
(1)  |        MNSMS_READ_REQ          |                                |
     * =============================> *                                |
(2)  |                                |      SIM_READ_RECORD_REQ       |
     |                                * =============================> *
MUTE(500)
(3)  |                                |      SIM_READ_RECORD_CNF       |
     |                                * <============================= *
(4)  |        MNSMS_READ_CNF          |                                |
     * <============================= *                                |
MUTE(500)
     |                                |                                |
```

Parametrization

Primitive		Parameter	Value
(1)	MNSMS_READ_REQ	mem_type	MEM_SM
		rec_num	SIM_RECORD_0
		read_mode	READ_PREVIEW
		status	SMS_RECORD_STO_UNSENT
(2)	SIM_READ_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_2
	<B>	record	SIM_RECORD_1
	<C>	record	SIM_RECORD_1
	<D>	record	SIM_RECORD_1
	<E>	record	SIM_RECORD_1
		length	LENGTH_SMS
(3)	SIM_READ_RECORD_CNF		
		datafield	SIM_SMS
		cause	SIM_NO_ERROR
	<A>	record	SIM_RECORD_2
	<B>	record	SIM_RECORD_1
	<C>	record	SIM_RECORD_1
	<D>	record	SIM_RECORD_1
	<E>	record	SIM_RECORD_1
		max_record	SIM_RECORD_3
		length	LENGTH_SMS
	<A>	linear_data	SIM_SMS_MO
	<B>	linear_data	SIM_SMS_MO
	<C>	linear_data	SIM_SMS_MO
	<D>	linear_data	SIM_SMS_SBM_DEF
	<E>	linear_data	SIM_SMS_MO
(4)	MNSMS_READ_CNF		
		mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_2
	<B>	rec_num	SIM_RECORD_1
	<C>	rec_num	SIM_RECORD_1
	<D>	rec_num	SIM_RECORD_1
	<E>	rec_num	SIM_RECORD_1
	<A>	rec_next	SMS_RECORD_NOT_EXIST
	<B>	rec_next	SMS_RECORD_NOT_EXIST
	<C>	rec_next	SMS_RECORD_NOT_EXIST
	<D>	rec_next	SMS_RECORD_NOT_EXIST
	<E>	rec_next	SIM_RECORD_2
		rec_max	SIM_RECORD_3
		cause	SIM_NO_ERROR
		rec_status	SMS_RECORD_NOT_EXIST
		status	SMS_RECORD_STO_UNSENT
	<A>	sms_sdu	SMS_SDU_MO
	<B>	sms_sdu	SMS_SDU_MO
	<C>	sms_sdu	SMS_SDU_MO
	<D>	sms_sdu	SMS_SDU_SBM_DEF
	<E>	sms_sdu	SMS_SDU_MO

History:	15-Jan-2001	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept
	11-Feb-2003	FK	SDU bit length of variant D corrected

3.10.12 SMS429: Searching for Short Messages, Mobile Terminated, not Read

Description: The first Short Message with the property given by the parameter 'status' will be retrieved from SIM.

Variants: <A>...<C>

Preamble:

<A>	SMS014
<B>	SMS015
<C>	SMS161A

```
    MMI                               SMS                             SIM/MM
     |                                 |                                 |
(1)  |         MNSMS_READ_REQ          |                                 |
     * ==============================> *                                 |
(2)  |                                 |      SIM_READ_RECORD_REQ        |
     |                                 * ==============================> *
MUTE(500)
(3)  |                                 |      SIM_READ_RECORD_CNF        |
     |                                 * <============================== *
(4)  |         MNSMS_READ_CNF          |                                 |
     * <============================== *                                 |
MUTE(500)
     |                                 |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_READ_REQ	mem_type	MEM_SM
	rec_num	SIM_RECORD_0
	read_mode	READ_PREVIEW
	status	SMS_RECORD_REC_UNREAD
(2) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
<A>	record	SIM_RECORD_2
<B>	record	SIM_RECORD_3
<C>	record	SIM_RECORD_1
	length	LENGTH_SMS
(3) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
<A>	record	SIM_RECORD_2
<B>	record	SIM_RECORD_3
<C>	record	SIM_RECORD_1
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
<A>	linear_data	SIM_SMS_MT
<B>	linear_data	SIM_SMS_MT
<C>	linear_data	SIM_SMS_CLASS_1

(4) MNSMS_READ_CNF		
	mem_type	MEM_SM
<A>	rec_num	SIM_RECORD_2
<B>	rec_num	SIM_RECORD_3
<C>	rec_num	SIM_RECORD_1
<A>	rec_next	SMS_RECORD_NOT_EXIST
<B>	rec_next	SMS_RECORD_NOT_EXIST
<C>	rec_next	SMS_RECORD_NOT_EXIST
	rec_max	SIM_RECORD_3
	cause	SIM_NO_ERROR
	rec_status	SMS_RECORD_NOT_EXIST
	status	SMS_RECORD_REC_UNREAD
<A>	sms_sdu	SMS_SDU_MT
<B>	sms_sdu	SMS_SDU_MT
<C>	sms_sdu	SMS_SDU_MT_7CL1

History:	15-Jan-2001	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.13 SMS431: Preview Reading of Short Messages, Mobile originated, not read

Description: In the preamble a mobile originated short message was indicated by SMS. This message is read from the MMI without changing the status.

Preamble: SMS014

```
   MMI                                SMS                              SIM/MM
     |                                 |                                 |
(1)  |          MNSMS_READ_REQ         |                                 |
     * ==============================> *                                 |
(2)  |                                 |        SIM_READ_RECORD_REQ      |
     |                                 * ==============================> *
MUTE(500)
(3)  |                                 |        SIM_READ_RECORD_CNF      |
     |                                 * <============================== *
(4)  |          MNSMS_READ_CNF         |                                 |
     * <============================== *                                 |
MUTE(500)
     |                                 |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_READ_REQ	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	read_mode	READ_PREVIEW
	status	BYTE_00
(2) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
(3) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_2
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT

(4) MNSMS_READ_CNF		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	rec_next	SMS_RECORD_NOT_EXIST
	rec_max	SIM_RECORD_3
	cause	SIM_NO_ERROR
	rec_status	SMS_RECORD_NOT_EXIST
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_MT

History:	29-Sep-99	LE	Initial
	13-Apr-2000	FK	Primitive change
	13-Dec-2000	LW	Adapted to new TAP
	15-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.14 SMS432: Preview-Reading of Short Messages, Mobile originated, already read

Description: In the preamble a mobile originated short message was indicated by SMS. This message is read from the MMI without changing the status.

Preamble: SMS016A

```
   MMI                                SMS                                SIM/MM
     |                                 |                                  |
(1)  |         MNSMS_READ_REQ          |                                  |
     * ==============================> *                                  |
(2)  |                                 |       SIM_READ_RECORD_REQ        |
     |                                 * ===============================> *
MUTE(500)
(3)  |                                 |       SIM_READ_RECORD_CNF        |
     |                                 * <=============================== *
(4)  |         MNSMS_READ_CNF          |                                  |
     * <============================== *                                  |
MUTE(500)
     |                                 |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_READ_REQ	mem_type	MEM_SM
	rec_num	SIM_RECORD_3
	read_mode	READ_PREVIEW
	status	BYTE_00
(2) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_3
	length	LENGTH_SMS
(3) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_3
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT_READ

		Parameter	Value
(4)	MNSMS_READ_CNF		
		mem_type	MEM_SM
		rec_num	SIM_RECORD_3
		rec_next	SMS_RECORD_NOT_EXIST
		rec_max	SIM_RECORD_3
		cause	SIM_NO_ERROR
		rec_status	SMS_RECORD_NOT_EXIST
		status	SMS_RECORD_REC_READ
		sms_sdu	SMS_SDU_MT

History:	13-Oct-98	LE	Initial
	13-Apr-2000	FK	Primitive change
	13-Dec-2000	LW	Adapted to new TAP
	15-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.15 SMS435: Preview-Reading of Short Messages, failure cases

Description: The MMI requests reading of a short message. This is not possible for different reasons. A failure cause shall be returned by SMS.
Variant A: record number 0, memory is empty
Variant B: record number too high
Variant C: record is unused
Variant D: ME memory is not available.
Variant E: SR memory is not available.
Variant F: invalid parameter 'mem_type'

Variants: <A>...<F>

Preamble:

<A>	SMS012A
<B>	SMS015
<C>	SMS015
<D>	SMS015
<E>	SMS015
<F>	SMS015

```
   MMI                             SMS                             SIM/MM
    |                               |                                |
(1) |          MNSMS_READ_REQ       |                                |
    * =============================> *                               |
(2) |          MNSMS_READ_CNF       |                                |
    * <============================== *                              |
    |                               |                                |
```

Parametrization

Primitive		Parameter	Value
(1)	MNSMS_READ_REQ		
	<A>	mem_type	MEM_SM
	<B>	mem_type	MEM_SM
	<C>	mem_type	MEM_SM
	<D>	mem_type	MEM_ME
	<E>	mem_type	MEM_SR
	<F>	mem_type	BYTE_00
	<A>	rec_num	SIM_RECORD_0
	<B>	rec_num	SIM_RECORD_5
	<C>	rec_num	SIM_RECORD_2
	<D>	rec_num	SIM_RECORD_1
	<E>	rec_num	SIM_RECORD_1
	<F>	rec_num	SIM_RECORD_1
		read_mode	READ_PREVIEW
		status	BYTE_00

(2) MNSMS_READ_CNF		
<A>	mem_type	MEM_SM
<B>	mem_type	MEM_SM
<C>	mem_type	MEM_SM
<D>	mem_type	MEM_ME
<E>	mem_type	MEM_SR
<F>	mem_type	BYTE_00
<A>	rec_num	SIM_RECORD_0
<B>	rec_num	SIM_RECORD_5
<C>	rec_num	SIM_RECORD_2
<D>	rec_num	SIM_RECORD_1
<E>	rec_num	SIM_RECORD_1
<F>	rec_num	SIM_RECORD_1
<A>	rec_next	SMS_RECORD_NOT_EXIST
<B>	rec_next	SMS_RECORD_NOT_EXIST
<C>	rec_next	SIM_RECORD_3
<D>	rec_next	SMS_RECORD_NOT_EXIST
<E>	rec_next	SMS_RECORD_NOT_EXIST
<F>	rec_next	SMS_RECORD_NOT_EXIST
<A>	rec_max	SIM_RECORD_3
<B>	rec_max	SIM_RECORD_3
<C>	rec_max	SIM_RECORD_3
<D>	rec_max	SMS_RECORD_NOT_EXIST
<E>	rec_max	SMS_RECORD_NOT_EXIST
<F>	rec_max	SMS_RECORD_NOT_EXIST
<A>	cause	SMS_CAUSE_INV_INDEX
<B>	cause	SMS_CAUSE_INV_INDEX
<C>	cause	SMS_CAUSE_INV_INDEX
<D>	cause	SMS_CAUSE_MEM_FAIL
<E>	cause	SMS_CAUSE_MEM_FAIL
<F>	cause	SMS_CAUSE_PARAM_WRONG
	rec_status	SMS_RECORD_NOT_EXIST
	status	SMS_RECORD_FREE
	sms_sdu	SMS_SDU_EMPTY

History:	14-Oct-98	LE	Initial
	16-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.16 SMS441: Status Change Reading of Short Messages, Mobile Originated, not Sent

Description: In the preamble a mobile originated short message was indicated by SMS. Changing the status is not applicable for unsent MO-SM, therefore an error is returned.

Preamble: SMS013

```
   MMI                                    SMS                                    SIM/MM
    |                                      |                                        |
(1) |          MNSMS_READ_REQ              |                                        |
    * ==============================> *                                        |
(2) |          MNSMS_READ_CNF              |                                        |
    * <============================== *                                        |
    |                                      |                                        |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_READ_REQ	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	read_mode	READ_STATUS_CHANGE
	status	BYTE_00

(2) MNSMS_READ_CNF		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	rec_next	SIM_RECORD_3
	rec_max	SIM_RECORD_3
	cause	SMS_CAUSE_OPER_NOT_ALLW
	rec_status	SMS_RECORD_NOT_EXIST
	status	SMS_RECORD_STO_UNSENT
	sms_sdu	SMS_SDU_EMPTY

History:	29-Sep-99	LE	Initial
	14-Apr-2000	FK	No status change of MO-SM
	16-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.17 SMS442: Status Change-Reading of SMS, Mobile Originated, already Sent

Description: In the preamble a mobile originated short message was indicated by SMS. This message is read from the MMI only for changing the status.

Preamble: SMS421A

```
   MMI                                SMS                                  SIM/MM
    |                                  |                                    |
(1) |          MNSMS_READ_REQ          |                                    |
    * ==============================> *                                    |
(2) |          MNSMS_READ_CNF          |                                    |
    * <============================== *                                    |
    |                                  |                                    |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_READ_REQ	mem_type	MEM_SM
	rec_num	SIM_RECORD_3
	read_mode	READ_STATUS_CHANGE
	status	BYTE_00
(2) MNSMS_READ_CNF		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_3
	rec_next	SMS_RECORD_NOT_EXIST
	rec_max	SIM_RECORD_3
	cause	SMS_CAUSE_OPER_NOT_ALLW
	rec_status	SMS_RECORD_NOT_EXIST
	status	SMS_RECORD_STO_SENT
	sms_sdu	SMS_SDU_EMPTY

History:	13-Oct-98	LE	Initial
	31-May-2000	FK	Modification due to +CMSS
	16-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.18 SMS443: Status Change Reading of Short Messages, Mobile terminated, not read

Description: In the preamble a mobile terminated short message was indicated by SMS. This message is read from the MMI only for changing the status.

Preamble: SMS014

```
   MMI                                SMS                                  SIM/MM
    |                                  |                                    |
(1) |          MNSMS_READ_REQ          |                                    |
    * ==============================> *                                    |
(2)  |                                |       SIM_READ_RECORD_REQ       |
     |                                * ==============================> *
MUTE(500)
(3)  |                                |       SIM_READ_RECORD_CNF       |
     |                                * <============================== *
(4)  |                                |      SIM_UPDATE_RECORD_REQ      |
     |                                * ==============================> *
MUTE(500)
(5)  |                                |      SIM_UPDATE_RECORD_CNF      |
     |                                * <============================== *
(6)  |        MNSMS_READ_CNF          |                                 |
     * <============================= *                                 |
MUTE(500)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_READ_REQ	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	read_mode	READ_STATUS_CHANGE
	status	BYTE_00
(2) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
(3) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_2
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT
(4) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT_READ
(5) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_2
	cause	SIM_NO_ERROR
(6) MNSMS_READ_CNF		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	rec_next	SMS_RECORD_NOT_EXIST
	rec_max	SIM_RECORD_3
	cause	SIM_NO_ERROR
	rec_status	SMS_RECORD_NOT_EXIST
	status	SMS_RECORD_REC_READ
	sms_sdu	SMS_SDU_EMPTY

History:	13-Oct-98	LE	Initial
	16-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.19 SMS444: Reread after Status Change Reading of Short Message, Mobile terminated, previously not read

Description: In the preamble the status of a mobile terminated short message previously indicated by SMS was changed by MMI. This message is read again by MMI.

Preamble: SMS443

```
   MMI                                SMS                               SIM/MM
    |                                  |                                  |
(1) |          MNSMS_READ_REQ          |                                  |
    * ===============================> *                                  |
(2) |                                  |       SIM_READ_RECORD_REQ        |
    |                                  * ===============================> *
MUTE(500)
(3) |                                  |       SIM_READ_RECORD_CNF        |
    |                                  * <=============================== *
(4) |          MNSMS_READ_CNF          |                                  |
    * <=============================== *                                  |
MUTE(500)
    |                                  |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_READ_REQ	mem_type	MEM_SM
		rec_num	SIM_RECORD_2
		read_mode	READ_NORMAL
		status	BYTE_00
(2)	SIM_READ_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
		record	SIM_RECORD_2
		length	LENGTH_SMS
(3)	SIM_READ_RECORD_CNF		
		datafield	SIM_SMS
		cause	SIM_NO_ERROR
		record	SIM_RECORD_2
		max_record	SIM_RECORD_3
		length	LENGTH_SMS
		linear_data	SIM_SMS_MT_READ
(4)	MNSMS_READ_CNF		
		mem_type	MEM_SM
		rec_num	SIM_RECORD_2
		rec_next	SMS_RECORD_NOT_EXIST
		rec_max	SIM_RECORD_3
		cause	SIM_NO_ERROR
		rec_status	SMS_RECORD_NOT_EXIST
		status	SMS_RECORD_REC_READ
		sms_sdu	SMS_SDU_MT

History:	03-Nov-99	FK	Initial
	13-Dec-00	LW	Adapted to new TAP
	16-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.20 SMS445: Status Change Reading of SMS, Mobile Terminated, already Read

Description: In the preamble a mobile terminated short message was indicated by SMS. This message is read from the MMI only for changing the status.

Preamble: SMS014

```
   MMI                                SMS                                SIM/MM
     |                                  |                                  |
(1)  |          MNSMS_READ_REQ          |                                  |
     * ===============================> *                                  |
(2)  |                                  |       SIM_READ_RECORD_REQ        |
     |                                  * ===============================> *
MUTE(500)
(3)  |                                  |       SIM_READ_RECORD_CNF        |
     |                                  * <=============================== *
(4)  |          MNSMS_READ_CNF          |                                  |
     * <=============================== *                                  |
     |                                  |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_READ_REQ	mem_type	MEM_SM
		rec_num	SIM_RECORD_2
		read_mode	READ_STATUS_CHANGE
		status	BYTE_00
(2)	SIM_READ_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
		record	SIM_RECORD_2
		length	LENGTH_SMS
(3)	SIM_READ_RECORD_CNF		
		datafield	SIM_SMS
		cause	SIM_NO_ERROR
		record	SIM_RECORD_2
		max_record	SIM_RECORD_3
		length	LENGTH_SMS
		linear_data	SIM_SMS_MT_READ
(4)	MNSMS_READ_CNF		
		mem_type	MEM_SM
		rec_num	SIM_RECORD_2
		rec_next	SMS_RECORD_NOT_EXIST
		rec_max	SIM_RECORD_3
		cause	SIM_NO_ERROR
		rec_status	SMS_RECORD_NOT_EXIST
		status	SMS_RECORD_REC_READ
		sms_sdu	SMS_SDU_EMPTY

History:			
	13-Oct-98	LE	Initial
	16-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.21 SMS446: Reread after Status Change Reading of Short Message, Mobile terminated, previously read

Description: In the preamble the status of a mobile terminated short message previously read by MMI was changed by MMI (which was obsolet). This message is read again by MMI.

Preamble: SMS445

```
   MMI                                SMS                                SIM/MM
     |                                  |                                  |
(1)  |          MNSMS_READ_REQ          |                                  |
     * ===============================> *                                  |
(2)  |                               |       SIM_READ_RECORD_REQ       |
     |                               * ===============================> *
MUTE(500)
(3)  |                               |       SIM_READ_RECORD_CNF       |
     |                               * <=============================== *
(4)  |         MNSMS_READ_CNF        |                                 |
     * <============================ *                                 |
MUTE(500)
     |                               |                                 |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_READ_REQ	mem_type	MEM_SM
		rec_num	SIM_RECORD_2
		read_mode	READ_NORMAL
		status	BYTE_00
(2)	SIM_READ_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
		record	SIM_RECORD_2
		length	LENGTH_SMS
(3)	SIM_READ_RECORD_CNF		
		datafield	SIM_SMS
		cause	SIM_NO_ERROR
		record	SIM_RECORD_2
		max_record	SIM_RECORD_3
		length	LENGTH_SMS
		linear_data	SIM_SMS_MT_READ
(4)	MNSMS_READ_CNF		
		mem_type	MEM_SM
		rec_num	SIM_RECORD_2
		rec_next	SMS_RECORD_NOT_EXIST
		rec_max	SIM_RECORD_3
		cause	SIM_NO_ERROR
		rec_status	SMS_RECORD_NOT_EXIST
		status	SMS_RECORD_REC_READ
		sms_sdu	SMS_SDU_MT

History:	03-Nov-99	FK	Initial
	13-Dec-00	LW	Adapted to new TAP
	16-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.22 SMS447: Status Change-Reading of Short Messages, failure cases

Description: The MMI requests reading of a short message. This is not possible for different reasons. A failure cause shall be returned by SMS.
Variant A: record number 0
Variant B: record number too high
Variant C: record is unused
Variant D: ME memory is not available.
Variant E: SR memory is not available.
Variant F: invalid parameter 'mem_type'

Variants: <A>...<F>

Preamble:

<A>	SMS012A
<B>	SMS015
<C>	SMS015

<D>	SMS015
<E>	SMS015
<F>	SMS015

```
   MMI                               SMS                              SIM/MM
    |                                 |                                 |
(1) |          MNSMS_READ_REQ         |                                 |
    * ==============================> *                                 |
(2) |          MNSMS_READ_CNF         |                                 |
    * <============================== *                                 |
    |                                 |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_READ_REQ		
<A>	mem_type	MEM_SM
<B>	mem_type	MEM_SM
<C>	mem_type	MEM_SM
<D>	mem_type	MEM_ME
<E>	mem_type	MEM_SR
<F>	mem_type	BYTE_00
<A>	rec_num	SIM_RECORD_0
<B>	rec_num	SIM_RECORD_5
<C>	rec_num	SIM_RECORD_1
<D>	rec_num	SIM_RECORD_2
<E>	rec_num	SIM_RECORD_1
<F>	rec_num	SIM_RECORD_1
	read_mode	READ_STATUS_CHANGE
	status	BYTE_00

(2) MNSMS_READ_CNF

<A>	mem_type	MEM_SM
<B>	mem_type	MEM_SM
<C>	mem_type	MEM_SM
<D>	mem_type	MEM_ME
<E>	mem_type	MEM_SR
<F>	mem_type	BYTE_00
<A>	rec_num	SIM_RECORD_0
<B>	rec_num	SIM_RECORD_5
<C>	rec_num	SIM_RECORD_1
<D>	rec_num	SIM_RECORD_2
<E>	rec_num	SIM_RECORD_1
<F>	rec_num	SIM_RECORD_1
<A>	rec_next	SMS_RECORD_NOT_EXIST
<B>	rec_next	SMS_RECORD_NOT_EXIST
<C>	rec_next	SIM_RECORD_3
<D>	rec_next	SMS_RECORD_NOT_EXIST
<E>	rec_next	SMS_RECORD_NOT_EXIST
<F>	rec_next	SMS_RECORD_NOT_EXIST
<A>	rec_max	SIM_RECORD_3
<B>	rec_max	SIM_RECORD_3
<C>	rec_max	SIM_RECORD_3
<D>	rec_max	SMS_RECORD_NOT_EXIST
<E>	rec_max	SMS_RECORD_NOT_EXIST
<F>	rec_max	SMS_RECORD_NOT_EXIST
<A>	cause	SMS_CAUSE_INV_INDEX
<B>	cause	SMS_CAUSE_INV_INDEX
<C>	cause	SMS_CAUSE_OPER_NOT_ALLW
<D>	cause	SMS_CAUSE_MEM_FAIL
<E>	cause	SMS_CAUSE_MEM_FAIL
<F>	cause	SMS_CAUSE_PARAM_WRONG
	rec_status	SMS_RECORD_NOT_EXIST
<A>	status	SMS_RECORD_FREE
<B>	status	SMS_RECORD_FREE
<C>	status	SMS_RECORD_STO_UNSENT
<D>	status	SMS_RECORD_FREE
<E>	status	SMS_RECORD_FREE
<F>	status	SMS_RECORD_FREE
	sms_sdu	SMS_SDU_EMPTY

History:	14-Oct-98	LE	Initial
	16-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.23 SMS471: Deletion of a Short Messages from SIM, successful case

Description: In preamble A, B a Short Message was read. It is now deleted by the MMI.
The other preamble indicate full memory (SIM/ME) with MCSF not set. A deletion must not lead to sending of RP-SMMA

Variants: <A>...<D>

Preamble:

<A>	SMS421A
<B>	SMS424B
<C>	SMS017A
<D>	SMS017D

```
   MMI                                    SMS                                     SIM/MM
     |                                     |                                         |
(1)  |         MNSMS_DELETE_REQ            |                                         |
     * ============================> *                                         |
(2)  |                                  |      SIM_UPDATE_RECORD_REQ      |
     |                                  * ===============================> *
MUTE(500)
(3)  |                                  |      SIM_UPDATE_RECORD_CNF      |
     |                                  * <=============================== *
(4)  |         MNSMS_DELETE_CNF         |                                 |
     * <=============================== *                                 |
     |                                  |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_DELETE_REQ		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
(2) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
	linear_data	SIM_SMS_EMPTY
(3) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_2
	cause	SIM_NO_ERROR
(4) MNSMS_DELETE_CNF		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	cause	SIM_NO_ERROR

History:	6-Jan-98	SZ	Initial
	16-Oct-98	LE	updated
	17-Jan-2001	FK	Major rework
	20-Jun-2002	FK	TC 472 included
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.24 SMS472: Deletion of a Short Messages from ME, successful case

Description: In the preamble a mobile terminated short message was read. It is now deleted by the MMI.

Variants: <A>...<D>

Preamble:

<A>	SMS012E
<B>	SMS012F
<C>	SMS017C
<D>	SMS017D

```
   MMI                              SMS                              SIM/MM
     |                                  |                                 |
(1)  |         MNSMS_DELETE_REQ         |                                 |
     * ===============================> *                                 |
(2)  |         MNSMS_DELETE_CNF         |                                 |
     * <=============================== *                                 |
     |                                  |                                 |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_DELETE_REQ		
		mem_type	MEM_ME
		rec_num	SIM_RECORD_1
(2)	MNSMS_DELETE_CNF		
		mem_type	MEM_ME
		rec_num	SIM_RECORD_1
		cause	SIM_NO_ERROR

History:	20-Jan-2002	FK	New to cover ME memory
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.25 SMS473: Deleting of Short Messages, failure cases

Description: The MMI requests deleting of a short message. This is not possible for different reasons. A failure cause shall be returned by SMS.
Variant A: record number 0
Variant B: record number too high
Variant C: record is unused
Variant D: ME memory is not available.
Variant E: SR memory is not available.
Variant F: invalid parameter 'mem_type'

Variants: <A>...<F>

Preamble: SMS013

```
   MMI                               SMS                              SIM/MM
     |                                 |                                 |
(1)  |         MNSMS_DELETE_REQ        |                                 |
     * ==============================> *                                 |
(2)  |         MNSMS_DELETE_CNF        |                                 |
     * <============================== *                                 |
     |                                 |                                 |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_DELETE_REQ		
	<A>	mem_type	MEM_SM
	<B>	mem_type	MEM_SM
	<C>	mem_type	MEM_SM
	<D>	mem_type	MEM_ME
	<E>	mem_type	MEM_SR
	<F>	mem_type	BYTE_00
	<A>	rec_num	SIM_RECORD_0
	<B>	rec_num	SIM_RECORD_5
	<C>	rec_num	SIM_RECORD_1
	<D>	rec_num	SIM_RECORD_2
	<E>	rec_num	SIM_RECORD_1
	<F>	rec_num	SIM_RECORD_1

(2) MNSMS_DELETE_CNF

<A>	mem_type	MEM_SM
<B>	mem_type	MEM_SM
<C>	mem_type	MEM_SM
<D>	mem_type	MEM_ME
<E>	mem_type	MEM_SR
<F>	mem_type	BYTE_00
<A>	rec_num	SIM_RECORD_0
<B>	rec_num	SIM_RECORD_5
<C>	rec_num	SIM_RECORD_1
<D>	rec_num	SIM_RECORD_2
<E>	rec_num	SIM_RECORD_1
<F>	rec_num	SIM_RECORD_1
<A>	cause	SMS_CAUSE_INV_INDEX
<B>	cause	SMS_CAUSE_INV_INDEX
<C>	cause	SMS_CAUSE_INV_INDEX
<D>	cause	SMS_CAUSE_MEM_FAIL
<E>	cause	SMS_CAUSE_MEM_FAIL
<F>	cause	SMS_CAUSE_PARAM_WRONG

History:	14-Oct-98	LE	Initial
	17-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.26 SMS474: Deleting of Mobile Originated Short Message from full SIM memory

Description: The preamble indicates, that all SMS records on SIM/ME memory are used. One SIM record is now deleted by the MMI. A RP-SMMA message is sent to the network to indicate that memory is available again for storage of SMs.

Variants: <A>...<B>

Preamble:

<A>	SMS016A
<B>	SMS016D

```
   MMI                               SMS                                SIM/MM
    |                                 |                                   |
(1) |        MNSMS_DELETE_REQ         |                                   |
    * ==============================> *                                   |
(2) |                                 |      SIM_UPDATE_RECORD_REQ        |
    |                                 * ==============================> *
MUTE(500)
(3) |                                 |      SIM_UPDATE_RECORD_CNF        |
    |                                 * <============================== *
(3) |                                 |       MMSMS_ESTABLISH_REQ         |
    |                                 * ==============================> *
(1) |                                 |       MMSMS_ESTABLISH_CNF         |
    |                                 * <============================== *
(2) |                                 |          MMSMS_DATA_REQ           |
    |                                 |             (CP_DATA)             |
    |                                 * ==============================> *
(3) |                                 |          MMSMS_DATA_IND           |
    |                                 |             (CP_ACK)              |
    |                                 * <============================== *
(7) |                                 |          MMSMS_DATA_IND           |
    |                                 |             (CP_DATA)             |
    |                                 * <============================== *
(8) |                                 |          MMSMS_DATA_REQ           |
    |                                 |             (CP_ACK)              |
    |                                 * ==============================> *
(4) |                                 |         MMSMS_RELEASE_REQ         |
    |                                 * ==============================> *
```

```
(5) |                                |          SIM_UPDATE_REQ         |
    |                                * ==============================> *
MUTE(500)
(5) |                                |          SIM_UPDATE_CNF         |
    |                                * <============================== *
(4) |        MNSMS_DELETE_CNF        |                                 |
    * <============================= *                                 |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_DELETE_REQ		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_1
(2) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_EMPTY
(3) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_NO_ERROR
(4) MMSMS_ESTABLISH_REQ		
	ti	TI_MO
(5) MMSMS_ESTABLISH_CNF		
	ti	TI_MO
(6) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MO
	cp_user_data_ul	RP_SMMA
	}	
(7) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MO_TO_MS
	}	

(8) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	D_CP_DATA
ti	TI_MO_TO_MS
cp_user_data_dl	RP_ACK_SMMA
}	

(9) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MO
}	

(10) MMSMS_RELEASE_REQ

ti	TI_MO

(11) SIM_UPDATE_REQ

source	SRC_SMS
offset	OFFSET_1
datafield	SIM_SMSS
length	SIMREC_SMSS_MEM_FLAG_LEN
trans_data	SIMREC_SMSS_MEM_FLAG_AVAIL

(12) SIM_UPDATE_CNF

datafield	SIM_SMSS
cause	SIM_NO_ERROR

(13) MNSMS_DELETE_CNF

mem_type	MEM_SM
rec_num	SIM_RECORD_1
cause	SIM_NO_ERROR

History:	26-Nov-99	FK	Initial
	14-Jan-2000	FK	Update of EF(SMSS) added
	13-Dec-2000	LW	Adapted to new TAP
	17-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.27 SMS475: Deleting of Mobile Originated Short Message from full ME memory

Description: The preamble indicates, that all SMS records on SIM/ME memory are used. One ME record is now deleted by the MMI. A RP-SMMA message is sent to the network to indicate that memory is available again for storage of SMs.

Preamble: SMS016D

```
   MMI                            SMS                           SIM/MM
    |                              |                              |
(1) |        MNSMS_DELETE_REQ      |                              |
    * ===========================> *                              |
(2) |                              |       MMSMS_ESTABLISH_REQ    |
    |                              * ===========================> *
(1) |                              |       MMSMS_ESTABLISH_CNF    |
    |                              * <=========================== *
```

```
(2)  |                          |         MMSMS_DATA_REQ           |
     |                          |            (CP_DATA)             |
     |                          * ===============================> *
(3)  |                          |         MMSMS_DATA_IND           |
     |                          |            (CP_ACK)              |
     |                          * <=============================== *
(7)  |                          |         MMSMS_DATA_IND           |
     |                          |            (CP_DATA)             |
     |                          * <=============================== *
(8)  |                          |         MMSMS_DATA_REQ           |
     |                          |            (CP_ACK)              |
     |                          * ===============================> *
(4)  |                          |        MMSMS_RELEASE_REQ         |
     |                          * ===============================> *
(5)  |                          |         SIM_UPDATE_REQ           |
     |                          * ===============================> *
MUTE(500)
(5)  |                          |         SIM_UPDATE_CNF           |
     |                          * <=============================== *
(4)  |      MNSMS_DELETE_CNF    |                                  |
     * <======================= *                                  |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                          |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_DELETE_REQ		
	mem_type	MEM_ME
	rec_num	SIM_RECORD_1
(2) MMSMS_ESTABLISH_REQ		
	ti	TI_MO
(3) MMSMS_ESTABLISH_CNF		
	ti	TI_MO
(4) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MO
	cp_user_data_ul	RP_SMMA
	}	
(5) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MO_TO_MS
	}	

(6) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	D_CP_DATA
ti	TI_MO_TO_MS
cp_user_data_dl	RP_ACK_SMMA
}	

(7) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MO
}	

(8) MMSMS_RELEASE_REQ

ti	TI_MO

(9) SIM_UPDATE_REQ

source	SRC_SMS
offset	OFFSET_1
datafield	SIM_SMSS
length	SIMREC_SMSS_MEM_FLAG_LEN
trans_data	SIMREC_SMSS_MEM_FLAG_AVAIL

(10) SIM_UPDATE_CNF

datafield	SIM_SMSS
cause	SIM_NO_ERROR

(11) MNSMS_DELETE_CNF

mem_type	MEM_ME
rec_num	SIM_RECORD_1
cause	SIM_NO_ERROR

History:	20-Jun-2002	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.28 SMS476: Deleting of Short Message from SIM, no RP-SMMA

Description: In preamble A a RP-SMMA was sent during the intialisation. The following deletion of an SM entry must not trigger the repetition of sending RP-SMMA.
In preambles B, C a mobile originated short message was deleted from full memory. Therefore a RP-SMMA message was sent to the network. After the second delete a RP-SMMA message must not be sent.

Variants: <A>...<C>

Preamble:

<A>	SMS481A
<B>	SMS474A
<C>	SMS475

```
   MMI                                  SMS                                  SIM/MM
     |                                    |                                     |
(1)  |         MNSMS_DELETE_REQ           |                                     |
     * =================================> *                                     |
(2)  |                               |      SIM_UPDATE_RECORD_REQ      |
     |                               * ===============================> *
(3)  |                               |      SIM_UPDATE_RECORD_CNF      |
     |                               * <=============================== *
(4)  |        MNSMS_DELETE_CNF       |                                 |
     * <============================ *                                 |
MUTE(500)
     |                               |                                 |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_DELETE_REQ		
		mem_type	MEM_SM
		rec_num	SIM_RECORD_3
(2)	SIM_UPDATE_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
		record	SIM_RECORD_3
		length	LENGTH_SMS
		linear_data	SIM_SMS_EMPTY
(3)	SIM_UPDATE_RECORD_CNF		
		datafield	SIM_SMS
		record	SIM_RECORD_3
		cause	SIM_NO_ERROR
(4)	MNSMS_DELETE_CNF		
		mem_type	MEM_SM
		rec_num	SIM_RECORD_3
		cause	SIM_NO_ERROR

History:	26-Nov-99	FK	Initial
	17-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.29 SMS477: Deleting of Short Message from ME Memory, no RP-SMMA

Description: In preamble A a RP-SMMA was sent during the intialisation. The following deletion of an SM entry must not trigger the repetition of sending RP-SMMA.
In preambles B, C a mobile originated short message was deleted from full memory. Therefore a RP-SMMA message was sent to the network. After the second delete a RP-SMMA message must not be sent.

Variants: <A>...<C>

Preamble:

<A>	SMS481C
<B>	SMS474B
<C>	SMS475

```
   MMI                           SMS                               SIM/MM
     |                               |                                 |
(1)  |        MNSMS_DELETE_REQ       |                                 |
     * ============================> *                                 |
(2)  |        MNSMS_DELETE_CNF       |                                 |
     * <============================ *                                 |
MUTE(500)
     |                               |                                 |
```

Parametrization

	Primitive	Parameter	Value
(1)	MNSMS_DELETE_REQ		
		mem_type	MEM_ME
		rec_num	SIM_RECORD_2
(2)	MNSMS_DELETE_CNF		
		mem_type	MEM_ME
		rec_num	SIM_RECORD_2
		cause	SIM_NO_ERROR

History:	20-Jun-2002	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.30 SMS480: Memory Available Notification

Description: The SIM application jas send the initial parameters read from the SIM card, which indicate 'Memory Full'. SMS has found that empty records for storing Short Messages are available in the mobile equipment or on the SIM.. The MS sends a RP-SMMA to the network and waits for the RP-Response.

Variants: <A>...<F>

Preamble:

<A>	SMS018A
<B>	SMS018B
<C>	SMS018C
<D>	SMS018D
<E>	SMS019A
<F>	SMS019B

```
   MMI                                    SMS                              SIM/MM
     |                                     |                                  |
(9)  |                                     |       MMSMS_ESTABLISH_REQ        |
     |                                     * ===============================> *
(10) |                                     |       MMSMS_ESTABLISH_CNF        |
     |                                     * <=============================== *
(11) |                                     |          MMSMS_DATA_REQ          |
     |                                     |            (CP_DATA)             |
     |                                     * ===============================> *
(12) |                                     |          MMSMS_DATA_IND          |
     |                                     |             (CP_ACK)             |
     |                                     * <=============================== *
     |                                     |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	MMSMS_ESTABLISH_REQ		
		ti	TI_MO
(2)	MMSMS_ESTABLISH_CNF		
		ti	TI_MO

(3) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MO
	cp_user_data_ul	RP_SMMA
	}	
(4) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MO_TO_MS
	}	

History:	17-Jun-2002	FK	Initial

3.10.31 SMS481: SMS Ready Indication after Memory Available Notification

Description: With the preamble a Memory Available Notification (RP-SMMA) was sent to the network. This is acknowledged with a RP-ACK message. The connection is released and the SMS Ready Notification is given to the MMI.

Variants: <A>...<C>

Preamble:

<A>	SMS480A
<B>	SMS480D
<C>	SMS480F

```
   MMI                               SMS                             SIM/MM
     |                                |                                 |
MUTE(1000)
(1)  |                                |         MMSMS_DATA_IND          |
     |                                |            (CP_DATA)            |
     |                                * <============================== *
(2)  |                                |         MMSMS_DATA_REQ          |
     |                                |             (CP_ACK)            |
     |                                * ==============================> *
(3)  |                                |        MMSMS_RELEASE_REQ        |
     |                                * ==============================> *
(4)  |                                |         SIM_UPDATE_REQ          |
     |                                * ==============================> *
MUTE(500)
(5)  |                                |         SIM_UPDATE_CNF          |
     |                                * <============================== *
(6)  |       MNSMS_REPORT_IND         |                                 |
     * <============================= *                                 |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_ACK_SMMA
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MO
	}	
(3) MMSMS_RELEASE_REQ		
	ti	TI_MO
(4) SIM_UPDATE_REQ		
	source	SRC_SMS
	offset	OFFSET_1
	datafield	SIM_SMSS
	length	SIMREC_SMSS_MEM_FLAG_LEN
	trans_data	SIMREC_SMSS_MEM_FLAG_AVAIL
(5) SIM_UPDATE_CNF		
	datafield	SIM_SMSS
	cause	SIM_NO_ERROR
(6) MNSMS_REPORT_IND		
	state	SMS_STATE_READY

History:	27-Nov-2001	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.32 SMS482: Repeat RP-SMMA after Timeout TR1M (Timer TRAM)

Description: The preamble indicates 'Memory Full', but in fact there is storage available for SMS on the SIM. The MS sends a RP-SMMA to the network which does nor reply with RP-ACK. After timeout TR1M the connection is released. After timeout TRAM the MS retries RP-SMMA, which is done successfully. EF(SMSS) is updated.

Preamble: SMS480A

```
   MMI                              SMS                         SIM/MM
    |                                |                              |
MUTE (37500)
(1)  |                               |       MMSMS_RELEASE_REQ      |
     |                               * ============================> *
MUTE (27500)
(2)  |                               |      MMSMS_ESTABLISH_REQ     |
     |                               * ============================> *
```

```
(3) |                                 |      MMSMS_ESTABLISH_CNF       |
    |                                 * <============================= *
(4) |                                 |         MMSMS_DATA_REQ         |
    |                                 |            (CP_DATA)           |
    |                                 * =============================> *
(5) |                                 |         MMSMS_DATA_IND         |
    |                                 |            (CP_ACK)            |
    |                                 * <============================= *
(6) |                                 |         MMSMS_DATA_IND         |
    |                                 |            (CP_DATA)           |
    |                                 * <============================= *
(7) |                                 |         MMSMS_DATA_REQ         |
    |                                 |            (CP_ACK)            |
    |                                 * =============================> *
(8) |                                 |       MMSMS_RELEASE_REQ        |
    |                                 * =============================> *
(9) |                                 |         SIM_UPDATE_REQ         |
    |                                 * =============================> *
MUTE(500)
(10)|                                 |         SIM_UPDATE_CNF         |
    |                                 * <============================= *
(11)|        MNSMS_REPORT_IND         |                                |
    * <============================== *                                |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                 |                                |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_RELEASE_REQ		
	ti	TI_MO
(2) MMSMS_ESTABLISH_REQ		
	ti	TI_MO
(3) MMSMS_ESTABLISH_CNF		
	ti	TI_MO
(4) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MO
	cp_user_data_ul	RP_SMMA_REP
	}	
(5) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MO_TO_MS
	}	

(6) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_ACK_SMMA_REP
	}	
(7) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MO
	}	
(8) MMSMS_RELEASE_REQ		
	ti	TI_MO
(9) SIM_UPDATE_REQ		
	source	SRC_SMS
	offset	OFFSET_1
	datafield	SIM_SMSS
	length	SIMREC_SMSS_MEM_FLAG_LEN
	trans_data	SIMREC_SMSS_MEM_FLAG_AVAIL
(10) SIM_UPDATE_CNF		
	datafield	SIM_SMSS
	cause	SIM_NO_ERROR
(11) MNSMS_REPORT_IND		
	state	SMS_STATE_READY

History:	18-Jan-2000	FK	Initial
	7-Nov-2000	LW	Timeout
	17-Jan-2001	FK	Preamble adapted, initialisation complete
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.33 SMS491: Modifying and Sending a Stored Message (MO)

Description: This test case simulates a scenario, which is similar to the things the ACI does when the "AT+CMSS" command is received. A message is read (in the preamble) and the status of all messages is requested in order to verify that the previous read has changed the status correctly. Then the same message is submitted with a different destination address and stored under the same record number as before. In addition the SIM message reference is updated by the SMS entity.

Variants: <A>...<H>

Preamble: SMS013

```
     MMI                                SMS                               SIM/MM
      |                                  |                                  |
(1)   |         MNSMS_SUBMIT_REQ         |                                  |
      * ===============================> *                                  |
(2)   |                                  |       SIM_READ_RECORD_REQ        |
      |                                  * ===============================> *
MUTE(500)
```

```
(3)  |                                 |      SIM_READ_RECORD_CNF        |
     |                                 * <============================== *
(4)  |                                 |      MMSMS_ESTABLISH_REQ        |
     |                                 * ==============================> *
(5)  |                                 |         SIM_UPDATE_REQ          |
     |                                 * ==============================> *
MUTE(500)
(6)  |                                 |      MMSMS_ESTABLISH_CNF        |
     |                                 * <============================== *
(7)  |                                 |         MMSMS_DATA_REQ          |
     |                                 |            (CP_DATA)            |
     |                                 * ==============================> *
MUTE(500)
(8)  |                                 |         SIM_UPDATE_CNF          |
     |                                 * <============================== *
MUTE(500)
(9)  |                                 |         MMSMS_DATA_IND          |
     |                                 |            (CP_ACK)             |
     |                                 * <============================== *
MUTE(2000)
     |                                 |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_SUBMIT_REQ		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
<A>	condx	SMS_CONDX_OVR_NON
<B>	condx	SMS_CONDX_OVR_NON
<C>	condx	SMS_CONDX_OVR_NON
<D>	condx	SMS_CONDX_OVR_NON
<E>	condx	SMS_CONDX_OVR_MO
<F>	condx	SMS_CONDX_OVR_MO
<G>	condx	SMS_CONDX_OVR_ANY
<H>	condx	SMS_CONDX_OVR_ANY
<A>	modify	SMS_MODIFY_NON
<B>	modify	SMS_MODIFY_TPOA
<C>	modify	SMS_MODIFY_SCA
<D>	modify	SMS_MODIFY_TPOA_SCA
<E>	modify	SMS_MODIFY_NON
<F>	modify	SMS_MODIFY_TPOA
<G>	modify	SMS_MODIFY_SCA
<H>	modify	SMS_MODIFY_TPOA_SCA
	sms_sdu	SMS_SDU_MO_CHANGE
(2) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
(3) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_2
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MO
(4) MMSMS_ESTABLISH_REQ		
	ti	TI_MO

(5) SIM_UPDATE_REQ

	source	SRC_SMS
	offset	OFFSET_0
	datafield	SIM_SMSS
	length	SIMREC_SMSS_MSG_REF_LEN
	trans_data	SIMREC_SMSS_MSG_REF

(6) MMSMS_ESTABLISH_CNF

	ti	TI_MO

(7) MMSMS_DATA_REQ

	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MO
<A>	cp_user_data_ul	RP_DATA_SUBMIT_MO
<B>	cp_user_data_ul	RP_DATA_SUBMIT_DA
<C>	cp_user_data_ul	RP_DATA_SUBMIT_SCA
<D>	cp_user_data_ul	RP_DATA_SUBMIT_DA_SCA
<E>	cp_user_data_ul	RP_DATA_SUBMIT_MO
<F>	cp_user_data_ul	RP_DATA_SUBMIT_DA
<G>	cp_user_data_ul	RP_DATA_SUBMIT_SCA
<H>	cp_user_data_ul	RP_DATA_SUBMIT_DA_SCA
	}	

(8) SIM_UPDATE_CNF

	datafield	SIM_SMSS
	cause	SIM_NO_ERROR

(9) MMSMS_DATA_IND

	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MO_TO_MS
	}	

History:	24-Oct-2000	LW	Initial
	12-Dec-2000	LW	Adaption to new TAP
	17-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.34 SMS492: Update SIM with Sent Message (MO)

Description: In the preamble a MO-SM was accepted which needs to be sent. The establish request was issued to MM, now we wait for the confirmation that an SMS connection was established and start sending. After receiving the confirmation that sending was successful, a SIM update request is issued.

Variants: <A>...<H>

Preamble:

<A>	SMS491A
<B>	SMS491B
<C>	SMS491C
<D>	SMS491D
<E>	SMS491E

<F>	SMS491F
<G>	SMS491G
<H>	SMS491H

```
   MMI                                  SMS                                  SIM/MM
    |                                    |                                    |
(2) |                                    |           MMSMS_DATA_IND           |
    |                                    |             (CP_DATA)              |
    |                                    * <================================= *
(3) |                                    |           MMSMS_DATA_REQ           |
    |                                    |              (CP_ACK)              |
    |                                    * =================================> *
(4) |                                    |        SIM_UPDATE_RECORD_REQ       |
    |                                    * =================================> *
(5) |                                    |          MMSMS_RELEASE_REQ         |
    |                                    * =================================> *
MUTE(500)
(6) |                                    |        SIM_UPDATE_RECORD_CNF       |
    |                                    * <================================= *
(7) |         MNSMS_SUBMIT_CNF           |                                    |
    * <================================= *                                    |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                    |                                    |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_ACK_DLNK
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MO
	}	

(3)	SIM_UPDATE_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_1
	<B>	record	SIM_RECORD_1
	<C>	record	SIM_RECORD_1
	<D>	record	SIM_RECORD_1
	<E>	record	SIM_RECORD_2
	<F>	record	SIM_RECORD_2
	<G>	record	SIM_RECORD_2
	<H>	record	SIM_RECORD_2
		length	LENGTH_SMS
	<A>	linear_data	SIM_SMS_SBM_MO
	<B>	linear_data	SIM_SMS_SBM_DA
	<C>	linear_data	SIM_SMS_SBM_SCA
	<D>	linear_data	SIM_SMS_SBM_DA_SCA
	<E>	linear_data	SIM_SMS_SBM_MO
	<F>	linear_data	SIM_SMS_SBM_DA
	<G>	linear_data	SIM_SMS_SBM_SCA
	<H>	linear_data	SIM_SMS_SBM_DA_SCA
(4)	MMSMS_RELEASE_REQ	ti	TI_MO
(5)	SIM_UPDATE_RECORD_CNF		
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_1
	<B>	record	SIM_RECORD_1
	<C>	record	SIM_RECORD_1
	<D>	record	SIM_RECORD_1
	<E>	record	SIM_RECORD_2
	<F>	record	SIM_RECORD_2
	<G>	record	SIM_RECORD_2
	<H>	record	SIM_RECORD_2
		cause	SIM_NO_ERROR
(6)	MNSMS_SUBMIT_CNF		
		mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_1
	<B>	rec_num	SIM_RECORD_1
	<C>	rec_num	SIM_RECORD_1
	<D>	rec_num	SIM_RECORD_1
	<E>	rec_num	SIM_RECORD_2
	<F>	rec_num	SIM_RECORD_2
	<G>	rec_num	SIM_RECORD_2
	<H>	rec_num	SIM_RECORD_2
		cause	SMS_NO_ERROR
		tp_mr	TP_MR_3N1
		sms_sdu	SMS_SDU_EMPTY

History:	25-Oct-2000	LW	Initial
	13-Dec-2000	LW	Adapted to new TAP
	17-Jan-2001	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.35 SMS493: Modifying and Sending a Stored Message (MT)

Description: A MT-SM is read from SIM, modified according to the parameters of the send command, sent to the network and stored as MO-SM on the SIM.

Variants: <A>...<H>

Preamble: SMS015

```
   MMI                              SMS                             SIM/MM
    |                                |                                 |
(1) |        MNSMS_SUBMIT_REQ        |                                 |
    * =============================> *                                 |
(2) |                                |       SIM_READ_RECORD_REQ       |
    |                                * ==============================> *
MUTE(500)
(3) |                                |       SIM_READ_RECORD_CNF       |
    |                                * <============================== *
(4) |                                |       MMSMS_ESTABLISH_REQ       |
    |                                * ==============================> *
(5) |                                |         SIM_UPDATE_REQ          |
    |                                * ==============================> *
MUTE(500)
(6) |                                |       MMSMS_ESTABLISH_CNF       |
    |                                * <============================== *
(7) |                                |         MMSMS_DATA_REQ          |
    |                                |            (CP_DATA)            |
    |                                * ==============================> *
MUTE(500)
(8) |                                |         MMSMS_DATA_IND          |
    |                                |             (CP_ACK)            |
    |                                * <============================== *
MUTE(500)
(9) |                                |         SIM_UPDATE_CNF          |
    |                                * <============================== *
MUTE(2000)
    |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_SUBMIT_REQ		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_3
<A>	condx	SMS_CONDX_OVR_NON
<B>	condx	SMS_CONDX_OVR_NON
<C>	condx	SMS_CONDX_OVR_NON
<D>	condx	SMS_CONDX_OVR_NON
<E>	condx	SMS_CONDX_OVR_MO
<F>	condx	SMS_CONDX_OVR_MO
<G>	condx	SMS_CONDX_OVR_ANY
<H>	condx	SMS_CONDX_OVR_ANY
<A>	modify	SMS_MODIFY_NON
<B>	modify	SMS_MODIFY_TPOA
<C>	modify	SMS_MODIFY_SCA
<D>	modify	SMS_MODIFY_TPOA_SCA
<E>	modify	SMS_MODIFY_NON
<F>	modify	SMS_MODIFY_TPOA
<G>	modify	SMS_MODIFY_SCA
<H>	modify	SMS_MODIFY_TPOA_SCA
	sms_sdu	SMS_SDU_MO_CHANGE
(2) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_3
	length	LENGTH_SMS

(3) SIM_READ_RECORD_CNF

datafield	SIM_SMS
cause	SIM_NO_ERROR
record	SIM_RECORD_3
max_record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_MT

(4) MMSMS_ESTABLISH_REQ

ti	TI_MO

(5) SIM_UPDATE_REQ

source	SRC_SMS
offset	OFFSET_0
datafield	SIM_SMSS
length	SIMREC_SMSS_MSG_REF_LEN
trans_data	SIMREC_SMSS_MSG_REF

(6) MMSMS_ESTABLISH_CNF

ti	TI_MO

(7) MMSMS_DATA_REQ

	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MO
<A>	cp_user_data_ul	RP_DATA_SUBMIT_7DEF
<B>	cp_user_data_ul	RP_DATA_SUBMIT_7DEF_DA
<C>	cp_user_data_ul	RP_DATA_SUBMIT_7DEF_SCA
<D>	cp_user_data_ul	RP_DATA_SUBMIT_7DEF_DA_SCA
<E>	cp_user_data_ul	RP_DATA_SUBMIT_7DEF
<F>	cp_user_data_ul	RP_DATA_SUBMIT_7DEF_DA
<G>	cp_user_data_ul	RP_DATA_SUBMIT_7DEF_SCA
<H>	cp_user_data_ul	RP_DATA_SUBMIT_7DEF_DA_SCA
	}	

(8) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MO_TO_MS
}	

(9) SIM_UPDATE_CNF

datafield	SIM_SMSS
cause	SIM_NO_ERROR

History:	25-Jan-2001	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.36 SMS494: Update SIM with Forwarded Message (MT)

Description: In the preamble a MO-SM was accepted which needs to be sent. The establish request was issued to MM, now we wait for the confirmation that an SMS connection was established and start sending. After receiving the confirmation that sending was successful, a SIM update request is issued.

Variants: <A>...<H>

Preamble:

<A>	SMS493A
<B>	SMS493B
<C>	SMS493C
<D>	SMS493D
<E>	SMS493E
<F>	SMS493F
<G>	SMS493G
<H>	SMS493H

```
   MMI                               SMS                              SIM/MM
     |                                |                                 |
(2)  |                                |         MMSMS_DATA_IND          |
     |                                |            (CP_DATA)            |
     |                                * <============================== *
(3)  |                                |         MMSMS_DATA_REQ          |
     |                                |            (CP_ACK)             |
     |                                * ==============================> *
(4)  |                                |      SIM_UPDATE_RECORD_REQ      |
     |                                * ==============================> *
(5)  |                                |        MMSMS_RELEASE_REQ        |
     |                                * ==============================> *
MUTE(500)
(6)  |                                |      SIM_UPDATE_RECORD_CNF      |
     |                                * <============================== *
(7)  |        MNSMS_SUBMIT_CNF        |                                 |
    * <============================== *                                 |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_ACK_DLNK
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MO
	}	

(3)	SIM_UPDATE_RECORD_REQ		
		source	SRC_SMS
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_2
	<B>	record	SIM_RECORD_2
	<C>	record	SIM_RECORD_2
	<D>	record	SIM_RECORD_2
	<E>	record	SIM_RECORD_2
	<F>	record	SIM_RECORD_2
	<G>	record	SIM_RECORD_3
	<H>	record	SIM_RECORD_3
		length	LENGTH_SMS
	<A>	linear_data	SIM_SMS_SBM_7DEF
	<B>	linear_data	SIM_SMS_SBM_7DEF_DA
	<C>	linear_data	SIM_SMS_SBM_7DEF_SCA
	<D>	linear_data	SIM_SMS_SBM_7DEF_DA_SCA
	<E>	linear_data	SIM_SMS_SBM_7DEF
	<F>	linear_data	SIM_SMS_SBM_7DEF_DA
	<G>	linear_data	SIM_SMS_SBM_7DEF_SCA
	<H>	linear_data	SIM_SMS_SBM_7DEF_DA_SCA
(4)	MMSMS_RELEASE_REQ		
		ti	TI_MO
(5)	SIM_UPDATE_RECORD_CNF		
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_2
	<B>	record	SIM_RECORD_2
	<C>	record	SIM_RECORD_2
	<D>	record	SIM_RECORD_2
	<E>	record	SIM_RECORD_2
	<F>	record	SIM_RECORD_2
	<G>	record	SIM_RECORD_3
	<H>	record	SIM_RECORD_3
		cause	SIM_NO_ERROR
(6)	MNSMS_SUBMIT_CNF		
		mem_type	MEM_SM
	<A>	rec_num	SIM_RECORD_2
	<B>	rec_num	SIM_RECORD_2
	<C>	rec_num	SIM_RECORD_2
	<D>	rec_num	SIM_RECORD_2
	<E>	rec_num	SIM_RECORD_2
	<F>	rec_num	SIM_RECORD_2
	<G>	rec_num	SIM_RECORD_3
	<H>	rec_num	SIM_RECORD_3
		cause	SMS_NO_ERROR
		tp_mr	TP_MR_3N1
		sms_sdu	SMS_SDU_EMPTY

History:	25-Jan-2001	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.37 SMS495: Unsuccessful Transmission of a Forwarded Message

Description: An RP-ERROR message is received, this is reported to the MMI and the connection is released.

Variants: <A>...<B>

Preamble:

<A>	SMS491D
<B>	SMS493D

```
   MMI                                 SMS                                 SIM/MM
     |                                   |                                   |
(1)  |                                   |          MMSMS_DATA_IND           |
     |                                   |        (CP_DATA(RP_ERROR))        |
     |                                   * <============================== *
(2)  |                                   |          MMSMS_DATA_REQ           |
     |                                   |              (CP_ACK)             |
     |                                   * ==============================> *
(4)  |         MNSMS_SUBMIT_CNF          |                                   |
     * <============================== *                                   |
(3)  |                                   |        MMSMS_RELEASE_REQ          |
     |                                   * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                   |                                   |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_ERR_TEMP_FAILURE
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MO
	}	
(3) MNSMS_SUBMIT_CNF		
	mem_type	NOT_USED
	rec_num	SIM_RECORD_0
	cause	SMS_RX_CS_TEMP_FAILURE
	tp_mr	TP_MR_3N1
	sms_sdu	SMS_SDU_EMPTY
(4) MMSMS_RELEASE_REQ	ti	TI_MO

History:	25-Jan-2001	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept
	11-Feb-2003	FK	MMSMS_RELEASE_REQ is last

3.10.38 SMS496: Unsuccessful Storing of a Forwarded Message

Description: An update record error is received, it is reported to the MMI, that sending the message was successful, but the parameter 'rec_num' contains the value '0' to notify that the message was not stored.

Variants: <A>...<B>

Preamble:

<A> SMS491H
<B> SMS493H

```
   MMI                                SMS                                SIM/MM
    |                                  |                                  |
(2) |                                  |          MMSMS_DATA_IND          |
    |                                  |            (CP_DATA)             |
    |                                  * <=============================== *
(3) |                                  |          MMSMS_DATA_REQ          |
    |                                  |             (CP_ACK)             |
    |                                  * ===============================> *
(4) |                                  |      SIM_UPDATE_RECORD_REQ       |
    |                                  * ===============================> *
(5) |                                  |        MMSMS_RELEASE_REQ         |
    |                                  * ===============================> *
MUTE(500)
(6) |                                  |      SIM_UPDATE_RECORD_CNF       |
    |                                  * <=============================== *
(7) |         MNSMS_SUBMIT_CNF         |                                  |
    * <=============================== *                                  |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                  |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_DATA_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_ACK_DLNK
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MO
	}	
(3) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
<A>	record	SIM_RECORD_2
<B>	record	SIM_RECORD_3
	length	LENGTH_SMS
<A>	linear_data	SIM_SMS_SBM_DA_SCA
<B>	linear_data	SIM_SMS_SBM_7DEF_DA_SCA
(4) MMSMS_RELEASE_REQ	ti	TI_MO

(5)	SIM_UPDATE_RECORD_CNF		
		datafield	SIM_SMS
	<A>	record	SIM_RECORD_2
	<B>	record	SIM_RECORD_3
		cause	SIM_CAUSE_OTHER_ERROR
(6)	MNSMS_SUBMIT_CNF		
		mem_type	MEM_SM
		rec_num	SIM_RECORD_0
		cause	SMS_NO_ERROR
		tp_mr	TP_MR_3N1
		sms_sdu	SMS_SDU_EMPTY

History:	25-Jan-2001	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.39 SMS497: Start Sent Stored Message while an Income MT-SM is stored

Description: A received SM is stored on SIM record 1. While waiting for the response from the SIM Entity, a command is given to sent the message already existent on record 2. The sent message shall not overwrite the original message, therefore it shall be stored on SIM record 3.

Preamble: SMS031H

```
   MMI                               SMS                             SIM/MM
     |                                |                                 |
(1)  |                                |       MMSMS_ESTABLISH_IND       |
     |                                |          (CP_DATA(MT))          |
     |                                * <============================== *
(2)  |                                |         MMSMS_DATA_REQ          |
     |                                |           (CP_ACK(MT))          |
     |                                * ==============================> *
(3)  |                                |      SIM_UPDATE_RECORD_REQ      |
     |                                |               (MT)              |
     |                                * ==============================> *
MUTE(500)
(1)  |       MNSMS_SUBMIT_REQ         |                                 |
     * =============================> *                                 |
(2)  |                                |       SIM_READ_RECORD_REQ       |
     |                                |               (MO)              |
     |                                * ==============================> *
MUTE(500)
(2)  |                                |      SIM_UPDATE_RECORD_CNF      |
     |                                |               (MT)              |
     |                                * <============================== *
(5)  |       MNSMS_MESSAGE_IND        |                                 |
     * <============================= *                                 |
(6)  |                                |         MMSMS_DATA_REQ          |
     |                                |          (CP_DATA(MT))          |
     |                                * ==============================> *
MUTE(500)
(3)  |                                |       SIM_READ_RECORD_CNF       |
     |                                |               (MO)              |
     |                                * <============================== *
(4)  |                                |       MMSMS_ESTABLISH_REQ       |
     |                                * ==============================> *
(5)  |                                |         SIM_UPDATE_REQ          |
     |                                * ==============================> *
MUTE(500)
(6)  |                                |         MMSMS_DATA_IND          |
     |                                |           (CP_ACK(MT))          |
     |                                * <============================== *
```

```
(7)  |                                |       MMSMS_RELEASE_REQ         |
     |                                |              (MT)               |
     |                                * ==============================> *
MUTE(500)
(3)  |                                |      MMSMS_ESTABLISH_CNF        |
     |                                * <============================== *
(5)  |                                |         MMSMS_DATA_REQ          |
     |                                |          (CP_DATA(MO))          |
     |                                * ==============================> *
MUTE(500)
(6)  |                                |         MMSMS_DATA_IND          |
     |                                |          (CP_ACK(MO))           |
     |                                * <============================== *
MUTE(500)
(7)  |                                |         MMSMS_DATA_IND          |
     |                                |          (CP_DATA(MO))          |
     |                                * <============================== *
(8)  |                                |         MMSMS_DATA_REQ          |
     |                                |          (CP_ACK(MO))           |
     |                                * ==============================> *
(4)  |                                |     SIM_UPDATE_RECORD_REQ       |
     |                                |              (MO)               |
     |                                * ==============================> *
(7)  |                                |       MMSMS_RELEASE_REQ         |
     |                                |              (MO)               |
     |                                * ==============================> *
MUTE(500)
(9)  |                                |         SIM_UPDATE_CNF          |
     |                                * <============================== *
MUTE(500)
(6)  |                                |     SIM_UPDATE_RECORD_CNF       |
     |                                |              (MO)               |
     |                                * <============================== *
(7)  |       MNSMS_SUBMIT_CNF         |                                 |
     * <============================= *                                 |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT_2
	cp_user_data_dl	RP_DATA_DELIVER_7CL2
	}	

(2) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MT_2_FROM_MS
}	

(3) SIM_UPDATE_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_1
length	LENGTH_SMS
linear_data	SIM_SMS_CLASS_2

(4) MNSMS_SUBMIT_REQ

mem_type	MEM_SM
rec_num	SIM_RECORD_2
condx	SMS_CONDX_OVR_NON
modify	SMS_MODIFY_NON
sms_sdu	SMS_SDU_MO_CHANGE

(5) SIM_READ_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_2
length	LENGTH_SMS

(6) SIM_UPDATE_RECORD_CNF

datafield	SIM_SMS
record	SIM_RECORD_1
cause	SIM_NO_ERROR

(7) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SIM_RECORD_1
rec_max	SIM_RECORD_3
status	SMS_RECORD_REC_UNREAD
sms_sdu	SMS_SDU_DELIVER_7CL2

(8) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_2_FROM_MS
cp_user_data_ul	RP_ACK_ULNK
}	

(9) SIM_READ_RECORD_CNF

datafield	SIM_SMS
cause	SIM_NO_ERROR
record	SIM_RECORD_2
max_record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_MT

(10) MMSMS_ESTABLISH_REQ

ti	TI_MO

(11) SIM_UPDATE_REQ

source	SRC_SMS
offset	OFFSET_0
datafield	SIM_SMSS
length	SIMREC_SMSS_MSG_REF_LEN
trans_data	SIMREC_SMSS_MSG_REF

(12) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT_2
}	

(13) MMSMS_RELEASE_REQ

ti	TI_MT_2_FROM_MS

(14) MMSMS_ESTABLISH_CNF

ti	TI_MO

(15) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MO
cp_user_data_ul	RP_DATA_SUBMIT_7DEF
}	

(16) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MO_TO_MS
}	

(17) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	D_CP_DATA
ti	TI_MO_TO_MS
cp_user_data_dl	RP_ACK_DLNK
}	

(18) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MO
}	

(19) SIM_UPDATE_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_SBM_7DEF

(20) MMSMS_RELEASE_REQ

ti	TI_MO

(21) SIM_UPDATE_CNF

datafield	SIM_SMSS
cause	SIM_NO_ERROR

(22) SIM_UPDATE_RECORD_CNF

datafield	SIM_SMS
record	SIM_RECORD_3
cause	SIM_NO_ERROR

(23) MNSMS_SUBMIT_CNF

mem_type	MEM_SM
rec_num	SIM_RECORD_3
cause	SMS_NO_ERROR
tp_mr	TP_MR_3N1
sms_sdu	SMS_SDU_EMPTY

History:	28-Jan-2002	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.40 SMS498: Income MT-SM while Message to be sent is read from SIM

Description:

Preamble: SMS031H

```
   MMI                               SMS                              SIM/MM
     |                                |                                  |
(1)  |        MNSMS_SUBMIT_REQ        |                                  |
    * ==============================> *                                  |
(2)  |                                |       SIM_READ_RECORD_REQ        |
     |                                |               (MO)               |
     |                                * ==============================> *
MUTE(500)
(1)  |                                |       MMSMS_ESTABLISH_IND        |
     |                                |           (CP_DATA(MT))          |
     |                                * <=============================== *
(2)  |                                |          MMSMS_DATA_REQ          |
     |                                |           (CP_ACK(MT))           |
     |                                * ==============================> *
(4)  |                                |      SIM_UPDATE_RECORD_REQ       |
     |                                |               (MT)               |
     |                                * ==============================> *
MUTE(500)
```

```
(3) |                                 |       SIM_READ_RECORD_CNF       |
    |                                 |               (MO)              |
    |                                 * <============================== *
(4) |                                 |       MMSMS_ESTABLISH_REQ       |
    |                                 * ==============================> *
(5) |                                 |         SIM_UPDATE_REQ          |
    |                                 * ==============================> *
MUTE(500)
(2) |                                 |      SIM_UPDATE_RECORD_CNF      |
    |                                 |               (MT)              |
    |                                 * <============================== *
(5) |        MNSMS_MESSAGE_IND        |                                 |
    * <============================== *                                 |
(6) |                                 |         MMSMS_DATA_REQ          |
    |                                 |          (CP_DATA(MT))          |
    |                                 * ==============================> *
MUTE(500)
(9) |                                 |         SIM_UPDATE_CNF          |
    |                                 * <============================== *
MUTE(500)
(6) |                                 |         MMSMS_DATA_IND          |
    |                                 |           (CP_ACK(MT))          |
    |                                 * <============================== *
(7) |                                 |        MMSMS_RELEASE_REQ        |
    |                                 |               (MT)              |
    |                                 * ==============================> *
MUTE(500)
(3) |                                 |       MMSMS_ESTABLISH_CNF       |
    |                                 * <============================== *
(5) |                                 |         MMSMS_DATA_REQ          |
    |                                 |          (CP_DATA(MO))          |
    |                                 * ==============================> *
MUTE(500)
(6) |                                 |         MMSMS_DATA_IND          |
    |                                 |           (CP_ACK(MO))          |
    |                                 * <============================== *
MUTE(500)
(7) |                                 |         MMSMS_DATA_IND          |
    |                                 |          (CP_DATA(MO))          |
    |                                 * <============================== *
(8) |                                 |         MMSMS_DATA_REQ          |
    |                                 |           (CP_ACK(MO))          |
    |                                 * ==============================> *
(4) |                                 |      SIM_UPDATE_RECORD_REQ      |
    |                                 |               (MO)              |
    |                                 * ==============================> *
(7) |                                 |        MMSMS_RELEASE_REQ        |
    |                                 |               (MO)              |
    |                                 * ==============================> *
MUTE(500)
(6) |                                 |      SIM_UPDATE_RECORD_CNF      |
    |                                 |               (MO)              |
    |                                 * <============================== *
(7) |         MNSMS_SUBMIT_CNF        |                                 |
    * <============================== *                                 |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                 |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_SUBMIT_REQ		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	condx	SMS_CONDX_OVR_NON
	modify	SMS_MODIFY_NON
	sms_sdu	SMS_SDU_MO_CHANGE
(2) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
(3) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT_2
	cp_user_data_dl	RP_DATA_DELIVER_7CL2
	}	
(4) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_2_FROM_MS
	}	
(5) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_2
(6) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_2
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT
(7) MMSMS_ESTABLISH_REQ		
	ti	TI_MO
(8) SIM_UPDATE_REQ		
	source	SRC_SMS
	offset	OFFSET_0
	datafield	SIM_SMSS
	length	SIMREC_SMSS_MSG_REF_LEN
	trans_data	SIMREC_SMSS_MSG_REF

(9) SIM_UPDATE_RECORD_CNF

datafield	SIM_SMS
record	SIM_RECORD_1
cause	SIM_NO_ERROR

(10) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SIM_RECORD_1
rec_max	SIM_RECORD_3
status	SMS_RECORD_REC_UNREAD
sms_sdu	SMS_SDU_DELIVER_7CL2

(11) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_2_FROM_MS
cp_user_data_ul	RP_ACK_ULNK
}	

(12) SIM_UPDATE_CNF

datafield	SIM_SMSS
cause	SIM_NO_ERROR

(13) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT_2
}	

(14) MMSMS_RELEASE_REQ

ti	TI_MT_2_FROM_MS

(15) MMSMS_ESTABLISH_CNF

ti	TI_MO

(16) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MO
cp_user_data_ul	RP_DATA_SUBMIT_7DEF
}	

(17) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MO_TO_MS
}	

(18) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	D_CP_DATA
ti	TI_MO_TO_MS
cp_user_data_dl	RP_ACK_DLNK
}	

(19) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MO
}	

(20) SIM_UPDATE_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_SBM_7DEF

(21) MMSMS_RELEASE_REQ

ti	TI_MO

(22) SIM_UPDATE_RECORD_CNF

datafield	SIM_SMS
record	SIM_RECORD_3
cause	SIM_NO_ERROR

(23) MNSMS_SUBMIT_CNF

mem_type	MEM_SM
rec_num	SIM_RECORD_3
cause	SMS_NO_ERROR
tp_mr	TP_MR_3N1
sms_sdu	SMS_SDU_EMPTY

History:	30-Jan-2002	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.10.41 SMS499: Income MT-SM while Sent Message is stored on SIM

Description:

Preamble: SMS031H

```
   MMI                               SMS                             SIM/MM
    |                                 |                                 |
(1) |        MNSMS_SUBMIT_REQ         |                                 |
    * ==============================> *                                 |
(2) |                                 |       SIM_READ_RECORD_REQ       |
    |                                 |               (MO)              |
    |                                 * ==============================> *
MUTE(500)
(3) |                                 |       SIM_READ_RECORD_CNF       |
    |                                 |               (MO)              |
    |                                 * <============================== *
(4) |                                 |       MMSMS_ESTABLISH_REQ       |
    |                                 * ==============================> *
(5) |                                 |         SIM_UPDATE_REQ          |
    |                                 * ==============================> *
MUTE(500)
(9) |                                 |         SIM_UPDATE_CNF          |
    |                                 * <============================== *
MUTE(500)
(3) |                                 |       MMSMS_ESTABLISH_CNF       |
    |                                 * <============================== *
(5) |                                 |         MMSMS_DATA_REQ          |
    |                                 |          (CP_DATA(MO))          |
    |                                 * ==============================> *
MUTE(500)
(6) |                                 |         MMSMS_DATA_IND          |
    |                                 |           (CP_ACK(MO))          |
    |                                 * <============================== *
MUTE(500)
(7) |                                 |         MMSMS_DATA_IND          |
    |                                 |          (CP_DATA(MO))          |
    |                                 * <============================== *
(8) |                                 |         MMSMS_DATA_REQ          |
    |                                 |           (CP_ACK(MO))          |
    |                                 * ==============================> *
(4) |                                 |      SIM_UPDATE_RECORD_REQ      |
    |                                 |               (MO)              |
    |                                 * ==============================> *
(7) |                                 |        MMSMS_RELEASE_REQ        |
    |                                 |               (MO)              |
    |                                 * ==============================> *
MUTE(500)
(1) |                                 |       MMSMS_ESTABLISH_IND       |
    |                                 |          (CP_DATA(MT))          |
    |                                 * <============================== *
(2) |                                 |         MMSMS_DATA_REQ          |
    |                                 |           (CP_ACK(MT))          |
    |                                 * ==============================> *
(4) |                                 |      SIM_UPDATE_RECORD_REQ      |
    |                                 |               (MT)              |
    |                                 * ==============================> *
MUTE(500)
(6) |                                 |      SIM_UPDATE_RECORD_CNF      |
    |                                 |               (MO)              |
    |                                 * <============================== *
(7) |        MNSMS_SUBMIT_CNF         |                                 |
    * <============================== *                                 |
MUTE(500)
(2) |                                 |      SIM_UPDATE_RECORD_CNF      |
    |                                 |               (MT)              |
    |                                 * <============================== *
```

```
(5)  |        MNSMS_MESSAGE_IND         |                                 |
     * <=============================== *                                 |
(6)  |                                  |          MMSMS_DATA_REQ         |
     |                                  |           (CP_DATA(MT))         |
     |                                  * ===============================> *
MUTE(500)
(6)  |                                  |          MMSMS_DATA_IND         |
     |                                  |            (CP_ACK(MT))         |
     |                                  * <=============================== *
(7)  |                                  |        MMSMS_RELEASE_REQ        |
     |                                  |                (MT)             |
     |                                  * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                  |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_SUBMIT_REQ		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_2
	condx	SMS_CONDX_OVR_NON
	modify	SMS_MODIFY_NON
	sms_sdu	SMS_SDU_MO_CHANGE
(2) SIM_READ_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_2
	length	LENGTH_SMS
(3) SIM_READ_RECORD_CNF		
	datafield	SIM_SMS
	cause	SIM_NO_ERROR
	record	SIM_RECORD_2
	max_record	SIM_RECORD_3
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT
(4) MMSMS_ESTABLISH_REQ		
	ti	TI_MO
(5) SIM_UPDATE_REQ		
	source	SRC_SMS
	offset	OFFSET_0
	datafield	SIM_SMSS
	length	SIMREC_SMSS_MSG_REF_LEN
	trans_data	SIMREC_SMSS_MSG_REF
(6) SIM_UPDATE_CNF		
	datafield	SIM_SMSS
	cause	SIM_NO_ERROR
(7) MMSMS_ESTABLISH_CNF		
	ti	TI_MO

(8) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MO
cp_user_data_ul	RP_DATA_SUBMIT_7DEF
}	

(9) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MO_TO_MS
}	

(10) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	D_CP_DATA
ti	TI_MO_TO_MS
cp_user_data_dl	RP_ACK_DLNK
}	

(11) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MO
}	

(12) SIM_UPDATE_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_1
length	LENGTH_SMS
linear_data	SIM_SMS_SBM_7DEF

(13) MMSMS_RELEASE_REQ

ti	TI_MO

(14) MMSMS_ESTABLISH_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	D_CP_DATA
ti	TI_MT_2
cp_user_data_dl	RP_DATA_DELIVER_7CL2
}	

(15) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MT_2_FROM_MS
}	

(16) SIM_UPDATE_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_3
length	LENGTH_SMS
linear_data	SIM_SMS_CLASS_2

(17) SIM_UPDATE_RECORD_CNF

datafield	SIM_SMS
record	SIM_RECORD_1
cause	SIM_NO_ERROR

(18) MNSMS_SUBMIT_CNF

mem_type	MEM_SM
rec_num	SIM_RECORD_1
cause	SMS_NO_ERROR
tp_mr	TP_MR_3N1
sms_sdu	SMS_SDU_EMPTY

(19) SIM_UPDATE_RECORD_CNF

datafield	SIM_SMS
record	SIM_RECORD_3
cause	SIM_NO_ERROR

(20) MNSMS_MESSAGE_IND

mem_type	MEM_SM
rec_num	SIM_RECORD_3
rec_max	SIM_RECORD_3
status	SMS_RECORD_REC_UNREAD
sms_sdu	SMS_SDU_DELIVER_7CL2

(21) MMSMS_DATA_REQ

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_2_FROM_MS
cp_user_data_ul	RP_ACK_ULNK
}	

(22) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT_2
}	

(23) MMSMS_RELEASE_REQ

ti	TI_MT_2_FROM_MS

History:	30-Jan-2002	FK	Initial
	25-Oct-2002	FK	Adaption to Cause Concept

3.11 SIM Toolkit - SMS Download

3.11.1 SMS511: Class 2 - PID SMS Download, no support by SIM

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 2 message and the protocol identifier for SMS Download. The SIM has not indicated SMS download support. The message is stored on the SIM card.

Preamble: SMS031C

```
   MMI                                    SMS                                  SIM/MM
     |                                     |                                     |
(1)  |                                     |       MMSMS_ESTABLISH_IND          |
     |                                     |            (CP_DATA)                |
     |                                     * <=============================== *
(2)  |                                     |          MMSMS_DATA_REQ            |
     |                                     |            (CP_ACK)                 |
     |                                     * ===============================> *
(3)  |                                     |      SIM_UPDATE_RECORD_REQ         |
     |                                     * ===============================> *
MUTE(500)
(4)  |                                     |      SIM_UPDATE_RECORD_CNF         |
     |                                     * <=============================== *
(5)  |         MNSMS_MESSAGE_IND           |                                     |
     * <=============================== *                                     |
(6)  |                                     |          MMSMS_DATA_REQ            |
     |                                     |            (CP_DATA)                |
     |                                     * ===============================> *
(7)  |                                     |          MMSMS_DATA_IND            |
     |                                     |            (CP_ACK)                 |
     |                                     * <=============================== *
```

```
(8)  |                                       |         MMSMS_RELEASE_REQ         |
     |                                       * ==============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                       |                                   |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL2_SAT1
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_2_SAT
(4) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_NO_ERROR
(5) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_1
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_DELIVER_7CL2_SAT1
(6) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ACK_ULNK
	}	

(7) MMSMS_DATA_IND

d1	NOT_USED
d2	NOT_USED
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MT
}	

(8) MMSMS_RELEASE_REQ

ti	TI_MT_FROM_MS

History:	6-Jan-2000	LE	Initial
	30-Jan-2002	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.11.2 SMS512: Class 2 - PID SMS Download, wrong DCS

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 1 message and the protocol identifier for SMS Download. The SIM has indicated SMS download support. According to GSM 11.14, p. 52 the message shall be stored without any changes in normal EF$_{SMS}$ in this case. So the invalid combination of PID + DCS is stored as received and the user is alerted.

Preamble: SMS022

```
   MMI                                SMS                                SIM/MM
    |                                  |                                  |
(1) |                                  |        MMSMS_ESTABLISH_IND       |
    |                                  |             (CP_DATA)            |
    |                                  * <=============================== *
(2) |                                  |           MMSMS_DATA_REQ         |
    |                                  |              (CP_ACK)            |
    |                                  * ===============================> *
(3) |                                  |       SIM_UPDATE_RECORD_REQ      |
    |                                  * ===============================> *
MUTE(500)
(4) |                                  |       SIM_UPDATE_RECORD_CNF      |
    |                                  * <=============================== *
(5) |        MNSMS_MESSAGE_IND         |                                  |
    * <=============================== *                                  |
(6) |                                  |           MMSMS_DATA_REQ         |
    |                                  |             (CP_DATA)            |
    |                                  * ===============================> *
(7) |                                  |           MMSMS_DATA_IND         |
    |                                  |              (CP_ACK)            |
    |                                  * <=============================== *
(8) |                                  |         MMSMS_RELEASE_REQ        |
    |                                  * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                  |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) MMSMS_ESTABLISH_IND		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL1_SAT3
	}	
(2) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MT_FROM_MS
	}	
(3) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_2_INV_DCS
(4) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_NO_ERROR
(5) MNSMS_MESSAGE_IND		
	mem_type	MEM_SM
	rec_num	SIM_RECORD_1
	rec_max	SIM_RECORD_3
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_DELIVER_7CL1_SAT3
(6) MMSMS_DATA_REQ		
	d1	NOT_USED
	d2	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ACK_ULNK
	}	

(7)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(8)	MMSMS_RELEASE_REQ		
		ti	TI_MT_FROM_MS

History:	6-Jan-2000	LE	Initial
	8-Nov-2000	LW	Corrected update record request
	30-Jan-2002	FK	Major rework
	25-Oct-2002	FK	Adaption to Cause Concept

3.11.3 SMS521: Class 2 - PID SMS Download, support by SIM

Description: The Relay Layer receives a mobile terminated short message. The parameter data coding scheme of the message indicates that it is a class 2 message and the protocol identifier for SMS Download. The SIM has indicated SMS download support. The message is send to the SIM with an Envelope command.

Variant A:	class 2 message, 7-bit alphabet
Variant B:	class 2 message, 7-bit alphabet, general coding indication
Variant C:	class 2 message, 8-bit alphabet
Variant D:	class 2 message, 8-bit alphabet, compressed, general coding indication
Variant E:	One2One SIM-specific message, no One2One SIM present

Variants: <A>...<E>

Preamble: SMS022

```
   MMI                                     SMS                               SIM/MM
    |                                       |                                  |
(1) |                                       |       MMSMS_ESTABLISH_IND        |
    |                                       |            (CP_DATA)             |
    |                                       * <=============================== *
(2) |                                       |          MMSMS_DATA_REQ          |
    |                                       |             (CP_ACK)             |
    |                                       * ===============================> *
(3) |                                       |         SIM_TOOLKIT_REQ          |
    |                                       * ===============================> *
MUTE(500)
    |                                       |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	MMSMS_ESTABLISH_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	D_CP_DATA
		ti	TI_MT
	<A>	cp_user_data_dl	RP_DATA_DELIVER_7CL2_SAT1
	<B>	cp_user_data_dl	RP_DATA_DELIVER_7CL2_SAT2
	<C>	cp_user_data_dl	RP_DATA_DELIVER_8CL2_SAT1
	<D>	cp_user_data_dl	RP_DATA_DELIVER_8CL2_SAT2
	<E>	cp_user_data_dl	RP_DATA_DELIVER_121_C
		}	
(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	B_CP_ACK
		ti	TI_MT_FROM_MS
		}	
(3)	SIM_TOOLKIT_REQ		
		source	SRC_SMS
		req_id	REQ_ID_0
	<A>	stk_cmd	ENVELOPE_SMS_1
	<B>	stk_cmd	ENVELOPE_SMS_2
	<C>	stk_cmd	ENVELOPE_SMS_3
	<D>	stk_cmd	ENVELOPE_SMS_4
	<E>	stk_cmd	ENVELOPE_SMS_121_C

History:	6-Jan-98	SZ	Initial
	9-Nov-2000	LW	Former 515 is variant B, new C,D
	14-Dec-2000	LW	Adapted to new TAP
	30-Jan-2002	FK	Changed test environment
	25-Apr-2002	FK	One2One SIM-specific message added

3.11.4 SMS522: SMS Download, response 90 00

Description: The SIM entity responses to the envelope command with the error cause 90 00. SMS sends a RP-ACK to the network.

Preamble: SMS521A

```
    MMI                                        SMS                                  SIM/MM
     |                                          |                                      |
(1)  |          SIM_TOOLKIT_CNF                 |                                      |
     * ======================================> *                                      |
(2)  |                                          |           MMSMS_DATA_REQ             |
     |                                          |              (CP_DATA)               |
     |                                          * ==================================> *
(3)  |                                          |           MMSMS_DATA_IND             |
     |                                          |              (CP_ACK)                |
     |                                          * <================================== *
```

```
(4)  |                                    |         MMSMS_RELEASE_REQ         |
     |                                    * ==================================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                    |                                   |
```

Parametrization

	Primitive	Parameter	Value
(1)	SIM_TOOLKIT_CNF		
		cause	SIM_NO_ERROR
		req_id	REQ_ID_0
		stk_cmd	STK_CMD_EMPTY
(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	U_CP_DATA
		ti	TI_MT_FROM_MS
		cp_user_data_ul	RP_ACK_ULNK
		}	
(3)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(4)	MMSMS_RELEASE_REQ		
		ti	TI_MT_FROM_MS

History:			
	6-Jan-2000	LE	Initial
	30-Jan-2002	FK	Changed test environment
	25-Oct-2002	FK	Adaption to Cause Concept

3.11.5 SMS523: SMS Download, response 93 00

Description: The SIM entity responses to the envelope command with the error cause 93 00. SMS sends a RP-ERROR to the network.

Preamble: SMS521B

```
   MMI                                   SMS                                 SIM/MM
     |                                    |                                   |
(1)  |          SIM_TOOLKIT_CNF           |                                   |
     * ==================================> *                                   |
(2)  |                                    |          MMSMS_DATA_REQ           |
     |                                    |             (CP_DATA)             |
     |                                    * ==================================> *
(3)  |                                    |          MMSMS_DATA_IND           |
     |                                    |             (CP_ACK)              |
     |                                    * <================================== *
```

```
(4)  |                                     |         MMSMS_RELEASE_REQ        |
     |                                     * =================================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                     |                                    |
```

Parametrization

	Primitive	Parameter	Value
(1)	SIM_TOOLKIT_CNF		
		cause	SIM_CAUSE_SAT_BUSY
		req_id	REQ_ID_0
		stk_cmd	STK_CMD_EMPTY
(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	CP_DATA_RP_ERROR_SAT_BUSY
(3)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(4)	MMSMS_RELEASE_REQ		
		ti	TI_MT_FROM_MS

History:	6-Jan-2000	LE	Initial
	30-Jan-2002	FK	Changed test environment
	25-Oct-2002	FK	Adaption to Cause Concept

3.11.6 SMS524: SMS Download, response 9F 05

Description: The SIM entity responses to the envelope command with the error cause 9F 05. SMS sends a RP-ACK to the network.

Variants: <A>...<B>

Preamble:

<A> SMS521A
<B> SMS521C

```
   MMI                                    SMS                                  SIM/MM
     |                                     |                                    |
(1)  |         SIM_TOOLKIT_CNF             |                                    |
     * ==================================> *                                    |
(2)  |                                     |          MMSMS_DATA_REQ            |
     |                                     |             (CP_DATA)              |
     |                                     * =================================> *
(3)  |                                     |          MMSMS_DATA_IND            |
     |                                     |             (CP_ACK)               |
     |                                     * <================================= *
(4)  |                                     |         MMSMS_RELEASE_REQ          |
     |                                     * =================================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                     |                                    |
```

Parametrization

	Primitive	Parameter	Value
(1)	SIM_TOOLKIT_CNF		
		cause	SIM_NO_ERROR
		req_id	REQ_ID_0
	<A>	stk_cmd	STK_CMD_TPDU1
	<B>	stk_cmd	STK_CMD_TPDU2
(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
	<A>	sdu	CP_DATA_RP_ACK_TPDU1
	<B>	sdu	CP_DATA_RP_ACK_TPDU3
(3)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(4)	MMSMS_RELEASE_REQ		
		ti	TI_MT_FROM_MS

History:			
	6-Jan-2000	LE	Initial
	30-Jan-2002	FK	Changed test environment
	25-Oct-2002	FK	Adaption to Cause Concept

3.11.7 SMS525: SMS Download, response 9E 05

Description: The SIM entity responses to the envelope command with the error cause 9E 05. SMS sends a RP-ERROR to the network.

Variants: <A>...<B>

Preamble:

<A> SMS521B

<B> SMS521D

```
   MMI                                SMS                               SIM/MM
    |                                  |                                  |
(1) |          SIM_TOOLKIT_CNF         |                                  |
    * ===============================> *                                  |
(2) |                                  |          MMSMS_DATA_REQ          |
    |                                  |             (CP_DATA)            |
    |                                  * ===============================> *
(3) |                                  |          MMSMS_DATA_IND          |
    |                                  |             (CP_ACK)             |
    |                                  * <=============================== *
(4) |                                  |        MMSMS_RELEASE_REQ         |
    |                                  * ===============================> *
MUTE(500)
COMMAND (SMS STATUS PARTITION)
    |                                  |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	SIM_TOOLKIT_CNF		
		cause	SIM_CAUSE_DNL_ERROR
		req_id	REQ_ID_0
	<A>	stk_cmd	STK_CMD_TPDU1
	<B>	stk_cmd	STK_CMD_TPDU2
(2)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
	<A>	sdu	CP_DATA_RP_ERROR_TPDU2
	<B>	sdu	CP_DATA_RP_ERROR_TPDU4
(3)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(4)	MMSMS_RELEASE_REQ		
		ti	TI_MT_FROM_MS

History:	6-Jan-2000	LE	Initial
	30-Jan-2002	FK	Changed test environment
	25-Oct-2002	FK	Adaption to Cause Concept

3.12 Engineering Mode

3.12.1 SMS551: Acknowledge of the Infrastructure

Description: Tests the SMS061 with engineering mode – original description: A SMS connection is established and the response of the network receives. It is a CP-DATA message containing a RP-ACK message. The RP-ACK message is forwarded to the Relay Layer. The user is informed about the positive end of procedure. After reception of the CP-DATA message Control Protocol sends a CP-ACK message as response to the infrastructure. Relay Layer releases the SMS connection. The release of SMS connection is requested to MM.

Preamble: SMS046A

```
   MMI                              SMS                             SIM/MM
    |                                |                                 |
(1) |        EM_SMS_EVENT_REQ        |                                 |
    * =============================> *                                 |
MUTE(500)
(2) |                                |         MMSMS_DATA_IND          |
    |                                |            (CP_DATA)            |
    |                                * <============================== *
(3) |           EM_DATA_IND          |                                 |
    * <============================= *                                 |
(4) |                                |         MMSMS_DATA_REQ          |
    |                                |             (CP_ACK)            |
    |                                * ==============================> *
(5) |        MNSMS_SUBMIT_CNF        |                                 |
    * <============================= *                                 |
(6)  |                                  |        MMSMS_RELEASE_REQ          |
     |                                  * ================================> *
MUTE(500)
     |                                  |                                   |
```

Parametrization

	Primitive	Parameter	Value
(1)	EM_SMS_EVENT_REQ		
		bitmask_sms_h	Bitm_H
		bitmask_sms_l	Bitm_L
(2)	MMSMS_DATA_IND		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	D_CP_DATA
		ti	TI_MO_TO_MS
		cp_user_data_dl	RP_ACK_DLNK
		}	
(3)	EM_DATA_IND		
		entity	EM_ENTITY
(4)	MMSMS_DATA_REQ		
		d1	NOT_USED
		d2	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	B_CP_ACK
		ti	TI_MO
		}	
(5)	MNSMS_SUBMIT_CNF		
		mem_type	NOT_PRESENT_8BIT
		rec_num	SMS_RECORD_NOT_EXIST
		cause	SMS_NO_ERROR
		tp_mr	TP_MR_3N1
		sms_sdu	SMS_SDU_EMPTY
(6)	MMSMS_RELEASE_REQ		
		ti	TI_MO

History:	7-Nov-2001	OT	Initial
	13-Feb-2001	FK	Adaption to new SAP MNSMS
	25-Oct-2002	FK	Adaption to Cause Concept
	10-Feb-2003	FK	EM_DATA_IND sent before CP-ACK