GSM Protocol Stack

Test Specification

GSMS

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

Date: 20 January 2003
Document No.: 8443.408.01.003
File: GSMS.DOC

Table of Contents

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 AG 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 AG. Condat AG accepts no liability for any loss or damage arising from the use of any information contained in this document.

The software described in this document is furnished under a licence agreement and may be used or copied only in accordance with the terms of the agreement. It is an offence to copy the software in any way except as specifically set out in the agreement. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Condat AG.

TI Berlin AG

Alt Moabit 91a

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
8443.408.01.001	10-Jan-2001	LW	Initial
8443.408.01.002	7-Jun-2002	FK	Adaption to new SAP MNSMS (GPRS 1.3.3)
8443.408.01.003	27-Jan-2003	FK	Corrections to LLC Flow Control

0.2 References

[1] GSM 05.02 version 8.0.0 Release 1999
Digital cellular telecommunications system (Phase 2+);
Multiplexing and multiple access on the radio path

[2] GSM 04.60 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);

General Packet Radio Service (GPRS);
Mobile Station (MS) - Base Station System (BSS) interface;
Radio Link Control/ Medium Access Control (RLC/MAC) protocol

[3] GSM 04.08 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Mobile radio interface layer 3 specification

[4] GSM 03.64 version 6.1.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Overall description of the GPRS radio interface; Stage 2

[5] GSM 03.60 version 6.3.1 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Service description; Stage 2

[6] GSM 04.07 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Mobile radio interface signalling layer 3; General aspects

[7] GSM 04.64 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
General Packet Radio Service (GPRS);
Mobile Station - Serving GPRS Support Node (MS-SGSN)
Logical Link Control (LLC) layer specification

[8] GSM 05.08 version 6.4.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Radio subsystem link control

[9] GSM 05.10 version 6.3.0 Release 1997
Digital cellular telecommunications system (Phase 2+);
Radio subsystem synchronization

[10] GSM 03.20 TS 100 929: July 1998 (GSM 03.20 version 6.0.1)
Security related network functions, ETSI

[11] Draft GSM 03.22: August 1998 (GSM 03.22 version 6.1.0)
Functions related to Mobile Station (MS) in idle mode and group receive mode, ETSI

[12] GSM 04.65 V6.3.0: Subnetwork Dependant Convergence Protocol
ETSI, March 1999

[13] ITU-T V42bis ITU-T, Recommendation V.42 bis 1990

[14] GSM 09.60 GPRS Tunneling Protocol (GTP) across the Gn and Gp Interface

[15] RFC 1661 IETF STD 51 July 1994
The Point-to-Point Protocol (PPP)

[16] RFC 1662 IETF STD 51 July 1994
PPP in HDLC-like Framing

[17] RFC 1570 January 1994
PPP LCP Extensions

[18] RFC 1989 August 1996
PPP Link Quality Monitoring

[19] RFC 1332 May 1992
The PPP Internet Protocol Control Protocol (IPCP)

[20] RFC 1877 December 1995
PPP IPCP Extensions for Name Server Addresses

[21] RFC 2153 May 1997
PPP Vendor Extensions

[22] RFC 1334 October 1992
PPP Authentication Protocols (for Password Authentication Protocol only)

[23] RFC 1994 August 1996
PPP Challenge Handshake Authentication Protocol (CHAP)

[24] TIA/EIA-136-370
Packet-Data Services – Enhanced General Packet Radio for TIA/EIA-136 (EGPRS-136) - Overview, Telecommunications Industry Association

[25] TIA/EIA-136-376
Packet-Data Services – EGPRS-136 Mobility Management, Telecommunications Industry Association

[26] TIA/EIA-136-972
Packet-Data Services – Stage 2 Description, Telecommunications Industry Association

0.3 Abbreviations

ACI	Application Control Interface
AGCH	Access Grant Channel
AT	Attention sequence "AT" to indicate valid commands of the ACI
BCCH	Broadcast Control Channel
BS	Base Station
BSIC	Base Station Identification Code
C/R	Command/Response
C1	Path Loss Criterion

C2	Reselection Criterion
CBCH	Cell Broadcast Channel
CBQ	Cell Bar Qualify
CC	Call Control
CCCH	Common Control Channel
CCD	Condat Coder Decoder
CCI	Compression and Ciphering Interface
CHAP	Challenge Handshake Authentication Protocol
CKSN	Ciphering Key Sequence Number
CRC	Cyclic Redundancy Check
DCCH	Dedicated Control Channel
DCOMP	Identifier of the user data compression algorithm used for the N-DPU
DISC	Disconnect Frame
DL	Data Link Layer
DM	Disconnected Mode Frame
DTX	Discontinuous Transmission
E	Extension bit
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
GACI	GPRS Application Control Interface
GMM	GPRS Mobility Management
GP	Guard Period
GRR	GPRS RR
GSM	Global System for Mobile Communication
HDLC	High-level Data Link Control
HISR	High level Interrupt Service Routine
HPLMN	Home Public Land Mobile Network
I	Information Frame
IMEI	International Mobile Equipment Identity
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
IPCP	Internet Protocol Control Protocol
ITU	International Telecommunication Union
IWF	Interworking Function
Kc	Ciphering Key
L	Length Indicator
LAI	Location Area Information
LCP	Link Control Protocol
LISR	Low level Interrupt Service Routine
LLC	Logical Link Control
LPD	Link Protocol Discriminator
LQM	Link Quality Monitoring

M	More bit used to indicate the last segment of N-DPU
MAC	Medium Access Control
MCC	Mobile Country Code
MM	Mobility Management
MMI	Man Machine Interface
MNC	Mobile Network Code
MS	Mobile Station
MT	Mobile Termination
N(R)	Receive Number
N(S)	Send Number
NC	Network Control
NCC	National Colour Code
NCP	Network Control Protocol
NECI	New Establishment Causes included
N-PDU	Network Protocol Data Unit
NSAPI	Network Layer Service Access Point Identifier
OTD	Observed Time Difference
P	Poll Bit
P/F	Poll/Final Bit
PACCH	Packet Associated Control Channel
PAP	Password Authentication Protocol
PBCCH	Packet BCCH
PCCCH	Packet CCCH
PCOMP	Identifier of the protocol control information compression algorithm used for the N-DPU
PDCH	Packet Data Channel
PDP	Packet Data Protocol e.g. IP or X.25
PDTCH	Packet Data Traffic Channel
PRACH	Packet RACH
PSI	Packet System Information
PCH	Paging Channel
PCO	Point of Control and Observation
PDU	Protocol Data Unit
PL	Physical Layer
PLMN	Public Land Mobile Network
PPC	Packet Physical Convergence
PPP	Point-to-Point Protocol
PTP	Point to Point
QoS	Quality of Service
RACH	Random Access Channel
REJ	Reject Frame
RLC	Radio Link Control
RNR	Receive Not Ready Frame
RR	Radio Resource Management
RR	Receive Ready Frame
RTD	Real Time Difference
RTOS	Real Time Operating System

SABM	Set Asynchronous Balanced Mode
SACCH	Slow Associated Control Channel
SAP	Service Access Point
SAPI	Service Access Point Identifier
SDCCH	Stand alone Dedicated Control Channel
SDU	Service Data Unit
SGSN	Serving GPRS Support Node
SIM	Subscriber Identity Module
SM	Session Management
SMS	Short Message Service
SMSCB	Short Message Service Cell Broadcast
SNDCP	Subnetwork Dependant Convergence Protocol
SNSM	SNDCP-SM
SS	Supplementary Services
TAP	Test Application Program
TBF	Temporary Block Flow
TCH	Traffic Channel
TCH/F	Traffic Channel Full Rate
TCH/H	Traffic Channel Half Rate
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
TE	Terminal Equipment - e. g. a PC
TFI	Temporary Flow Identifier
TLLI	Temporary Logical Link Identifier
TMSI	Temporary Mobile Subscriber Identity
TOM	Tunnelling of Messages
TQI	Temporary Queuing Identifier
UA	Unnumbered Acknowledgement Frame
UART	Universal Asynchronous Receiver Transmitter
UI	Unnumbered Information Frame
USF	Uplink State Flag
V(A)	Acknowledgement State Variable
V(R)	Receive State Variable
V(S)	Send State Variable
VPLMN	Visited Public Land Mobile Network

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 protocol stack for GPRS consists of several entities. Each entity has one ore more service access points, over which the entity provides a service for the upper entity.

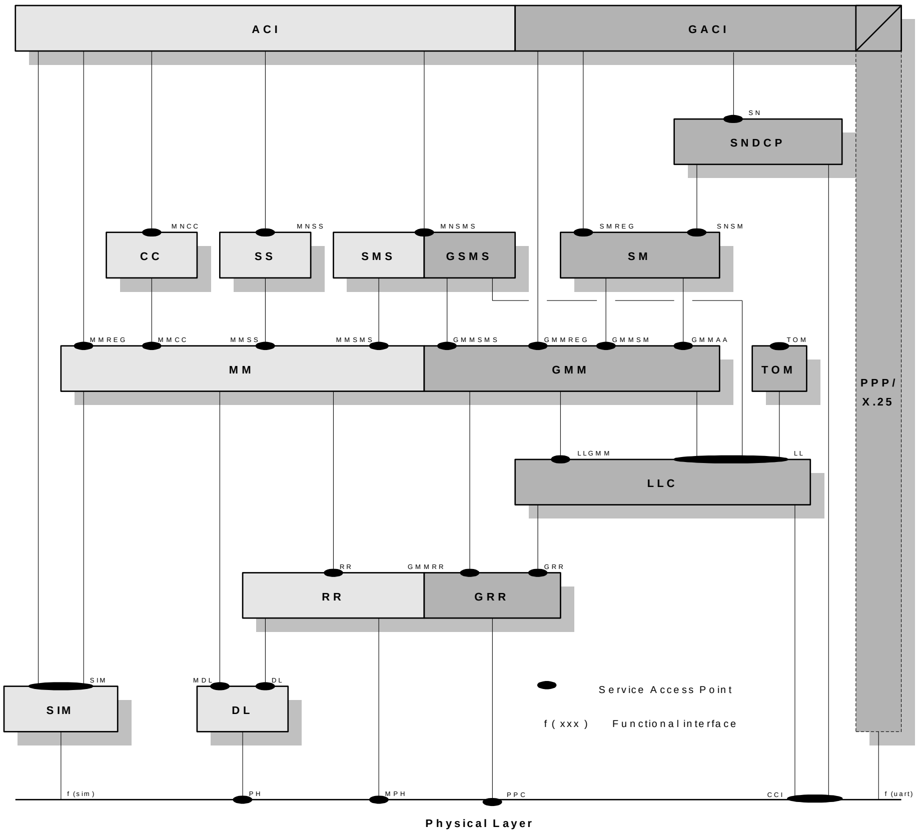

Figure 1: Mobile-station protocol architecture

The information units passed via the SAPs are called primitives and consists of an operation code and several parameters. See the Users Guide for details.

The entities of the GPRS protocol stack are:

1.1 GRR (RLC/MAC) – Radio Link Control/Medium Access Control

This layer contains two functions: The Radio Link Control function provides a radio-solution-dependent reliable link. The Medium Access Control function controls the access signalling (request and grant) procedures for the radio channel, and the mapping of LLC frames onto the GSM physical channel.

1.2 LLC – Logical Link Control

The LLC entity provides multiple highly reliable logical links for asynchronous data transfer between the MS and the network. It supports variable-length information frames, acknowledged and unacknowledged data transfer, flow and sequence control, error detection and recovery, notification of unrecoverable errors, user identity confidentiality, and ciphering of user and signaling data.

1.3 GMM – GPRS Mobility Management

The GMM entity provides procedures for the mobility of the MS, such as informing the network of its present location, and user identity confidentiality. It manages the GMM context (attach, detach, routing area updating), supports security functions such as authentication of user and MS, controls ciphering of data, and initiates the response to paging messages.

1.4 SM – Session Management

The main function of the session management (SM) is to support PDP context handling of the user terminal. Session Management activates, modifies and deletes the contexts for packet data protocols (PDP). Session Management services are provided at the SMREG-SAP and the SNSM-SAP for anonymous and non-anonymous access. The non-anonymous and anonymous access procedures for PDP context activation and PDP context deactivation are available at the SMREG-SAP. In addition there exists a PDP context modification for non-anonymous PDP contexts.

1.5 SNDCP - Subnetwork Dependant Convergence Protocol

SNDCP carries out all functions related to transfer of Network layer Protocol Data Units (N-PDUs) over GPRS in a transparent way. SNDCP helps to improve channel efficiency by means of compression techniques. The set of protocol entities above SNDCP consists of commonly used network protocols. They all use the same SNDCP entity, which then performs multiplexing of data coming from different sources to be sent using the service provided by the LLC layer.

1.6 GACI – GPRS Application Control Interface

The GACI is the GPRS extension of the ACI. It is specified in GSM 07.07 and 07.60. It is responsible for processing of the GPRS related AT Commands to setup, activate and deactivate the PDP context parameter. It also provides functionality for the interworking between GMM/SM/SNDCP and a packet oriented protocol like PPP.

1.7 USART - Universal Synchronous Asynchronous Receiver Transmitter Driver

The USART is a hardware component that facilitates a connection between the mobile station and terminal equipment (e.g. a PC). This interface uses some of the circuits described in V.24.

The data exchange provided by this unit is serial and asynchronous (synchronous communication is not in the scope of this document). A driver that uses interrupts to manage a circular buffer for the sending and receiving direction is necessary in order to use this component in the GPRS. The driver has to be able to perform flow control.

1.8 TOM – Tunnelling of Messages

The TOM entity is present if and only if HS136 is supported (the feature flag FF_HS136 is enabled).

The main function of TOM is to tunnel non-GSM signalling messages between the MS and the SGSN. The only non-GSM signalling which is currently supported by TOM is for the EGPRS-136 system (according to TIA/EIA-136-376). Data transfer in both uplink and downlink direction is possible. Two different priorities (high, low) of signalling data transfer are supported. TOM uses the unacknowledged mode of LLC and the acknowledged mode of GRR (RLC/MAC).

This document describes the tests for Short Messages Service via GPRS (GSMS).

2 Parameters

/* Declarations for GSMS only */
DECLARATION(SMS_DEFAULT_QOS)
DECLARATION(DEF_RES_UNITDATA_REQ1)
DECLARATION(DEF_RES_UNITDATA_REQ2)
DECLARATION(DEF_RES_UNITDATA_REQ3)
DECLARATION(DEF_RES_UNITDATA_REQ4)
DECLARATION(DEF_RES_UNITDATA_IND)
DECLARATION(DEF_SMS_LL_TLLI_1)

/*
Note: The following parameters are included from the standard SMS test document. Please add GSMS specific definitions below this field!
*/
/* 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", 1400)
        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", 1400)
		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
```

/*

Note: Definitions below this line are not imported from the SMS test document and therefore GSMS specific.

*/

```
BEGIN_PSTRUCT ( "ll_qos", SMS_DEFAULT_QOS)
                SET_COMP ("delay", LL_DELAY_SUB)
                SET_COMP ("relclass", LL_RLC_PROT)
                SET_COMP ("peak", LL_PEAK_SUB)
                SET_COMP ("preced", LL_PRECED_SUB)
                SET_COMP ("mean", LL_MEAN_SUB)
                SKIP_COMP ("reserved_1")
                SKIP_COMP ("reserved_2")
                SKIP_COMP ("reserved_3")
ENDSTRUCT

/* reserved parameters for GSMS */
BEGIN_PSTRUCT ("reserved_unitdata_req1", DEF_RES_UNITDATA_REQ1)
                SKIP_COMP ("ref_nsapi")
                SKIP_COMP ("ref_npdu_num")
                SKIP_COMP ("ref_seg_num")
ENDSTRUCT

LONG        DEF_RES_UNITDATA_REQ2               0x00000000

BYTE        SEG_POS_612                         0x0

BYTE        ATTACHED_COUNTER_612                0x0

LONG        DEF_SMS_LL_TLLI_1                   0x00000000

SHORT       DEF_RES_UNITDATA_REQ3           0x0000

LONG        DEF_RES_UNITDATA_REQ4           0x0

SHORT       DEF_RES_UNITDATA_REQ5           0x0

LONG        DEF_RES_UNITDATA_IND1           0x0

BYTE        DEF_RES_UNITDATA_IND3           0x0

BYTE        DEF_RES_UNITDATA_IND4           0x0

BYTE        DEF_RES_UNITDATA_IND5           0x0

SHORT       GSM_CP_NETWORK_FAILURE          (CP_NETWORK_FAILURE | 0x700)
```

3 TEST CASES

3.1 Routing (internal)

3.1.1 GSMS000: Setup the routing and PCO view for the GSMS test

Description: Routings for the GSMS tests are set.
(Note: When compiling this and all following test cases do not forget to specify that the entity under test is SMS. For example: "mkalltc GSMS EUT=SMS")

Preamble: None

```
      MMI                            SMS                            SIM/MM
      |                               |                               |
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 (LLC RESET)
COMMAND (GMM 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 (LLC REDIRECT CLEAR)
COMMAND (GMM 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 (SMS REDIRECT LLC TAP)
COMMAND (SMS REDIRECT GMM 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 (LLC REDIRECT GMM NULL)
COMMAND (LLC REDIRECT SMS NULL)
      |                               |                               |
COMMAND (GMM REDIRECT SMS NULL)
COMMAND (GMM REDIRECT SM NULL)
      |                               |                               |
COMMAND (TAP REDIRECT TAP SMS)
      |                               |                               |
```

Parametrization

Primitive	Parameter	Value

History: 6-Jan-98 SZ Initial

3.2 Initialisation Phase

3.2.1 GSMS001: Configuring the SMS Entity (no SMS messages stored)

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.

Preamble: GSMS000

```
   MMI                                SMS                            SIM/LLC
     |                                |                                |
(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)
(1)  |                                |          SIM_READ_CNF           |
     |                                * <============================== *
(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)|                                 |       LL_GETUNITDATA_REQ        |
     |                                * ==============================> *
(11)|        MNSMS_REPORT_IND         |                                 |
     * <============================= *                                 |
MUTE(500)
(12)|       MNSMS_CONFIGURE_REQ       |                                 |
     * =============================> *                                 |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) SIM_SMS_INSERT_IND		
	phase	PHASE_2_SIM
	tp_mr	TP_MR_3
	mem_cap_avail	TRUE
	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
	cause	SIM_CAUSE_UNKN_FILE_ID
	length	LEN_0
	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
trans_data	IMSI_NORMAL

(7) SIM_READ_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_1
length	LENGTH_SMS

(8) 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

(9) SIM_READ_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_2
length	LENGTH_SMS

(10) 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

(11) SIM_READ_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_3
length	LENGTH_SMS

(12) 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

(13) 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

(14) LL_GETUNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(15) MNSMS_REPORT_IND		
	state	SMS_STATE_READY
(16) MNSMS_CONFIGURE_REQ		
	pref_mem_3	MEM_SM
	mt	MT1
	ds	DS0
	mhc	SMS_MHC_DEF

History:	21-May-2002	FK	renewed
	21-Jan-2003	FK	LL_GETUNITDATA_REQ moved

3.3 Configuration of SMS Entity for GSMS Service

3.3.1 GSMS601: Preferred Destination Configuration

Description: The preferred destination for MO short messages is set to all currently possible combinations:

Variant A:	use GPRS only
Variant B:	use CCT only
Variant C:	use GPRS in preference
Variant D:	use CCT in preference
Variant E:	use GPRS only, without receiving SIM_SMS_INSERT_IND before

Variants: <A>...<E>

Preamble:

<A>	GSMS001
<B>	GSMS001
<C>	GSMS001
<D>	GSMS001
<E>	GSMS000

```
     MMI                            SMS                              SIM/LLC
      |                              |                                 |
(1)   |     MNSMS_MO_SERV_REQ        |                                 |
      *==============================>*                                |
(2)   |     MNSMS_MO_SERV_CNF         |                                |
      *<==============================*                                |
MUTE (500)
      |                               |                                |
```

Parametrization

Primitive	Parameter	Value
(1) MNSMS_MO_SERV_REQ		
<A>	mo_sms_serv	GPRS_SMS_GPRS_ONLY
<B>	mo_sms_serv	GPRS_SMS_CCT_ONLY
<C>	mo_sms_serv	GPRS_SMS_GPRS_PREF
<D>	mo_sms_serv	GPRS_SMS_CCT_PREF
<E>	mo_sms_serv	GPRS_SMS_GPRS_ONLY

(2) MNSMS_MO_SERV_CNF

<A>	mo_sms_serv	GPRS_SMS_GPRS_ONLY
<B>	mo_sms_serv	GPRS_SMS_CCT_ONLY
<C>	mo_sms_serv	GPRS_SMS_GPRS_PREF
<D>	mo_sms_serv	GPRS_SMS_CCT_PREF
<E>	mo_sms_serv	GPRS_SMS_GPRS_ONLY

History:	21-Nov-2000	LW	Initial

3.3.2 GSMS602: Flow Control Ready Indication for GPRS

Description: LLC indicates the readiness of flow control for SMS transfer via GPRS. SMS indicates to LLC that it is able to receive data.

Variants: <A>...<J>

Preamble:

<A>	GSMS601A
<B>	GSMS601B
<C>	GSMS601C
<D>	GSMS601D
<E>	GSMS001
<F>	GSMS601E
<G>	GSMS610C
<H>	GSMS610D
<I>	GSMS611G
<J>	GSMS611H

```
   MMI                                     SMS                                     LLC
    |                                       |                                       |
MUTE(500)
(1) |                                       |          LL_UNITREADY_IND             |
    |                                       * <==================================== *
    |                                       |                                       |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History:	22-Nov-2000	LW	Initial
	23-Sep-2002	FK	LL_GETUNITDATA_REQ removed
	04-Feb-2003	FK	additional Preambles for another scenario

3.4 Mobile Originated SM via GPRS

3.4.1 GSMS610: Sending MO-SM via GPRS: GMM Reg. Request

Description: The SMS entity is configured to use GPRS for sending mobile originated short messages. The MMI submits a message. Since this is the first message in a transaction, the SMS entity issues an registration state request to GMM.

Variants: <A>...<D>

Preamble:

<A>	GSMS602A
<B>	GSMS602C
<C>	GSMS601A
<D>	GSMS601C

```
     MMI                             SMS                              LLC
      |                               |                                 |
MUTE(500)
(1)   |     MNSMS_SUBMIT_REQ          |                                 |
      *==============================>*                                 |
(2)   |                               |     GMMSMS_REG_STATE_REQ        |
      |                               *================================>*
(3)   |                               |     SIM_UPDATE_REQ              |
      |                               *================================>*
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) GMMSMS_REG_STATE_REQ		
(3) SIM_UPDATE_REQ		
	source	SRC_SMS
	offset	OFFSET_0
	datafield	SIM_SMSS
	length	SIMREC_SMSS_MSG_REF_LEN
	trans_data	SIMREC_SMSS_MSG_REF

History:	23-Nov-2000	LW	Initial
	21-May-2002	FK	Adaption to new SAP MNSMS
	21-Jan-2003	FK	GMMSMS_REG_STATE_REQ is sent first
	04-Feb-2003	FK	additional Preambles for another scenario

3.4.2 GSMS611: GMM Registration Confirmation

Description: The SMS entity receives a confirmation for its registration information request from GMM.

Variant A: configured to use only GPRS, positive registration information

Variant B: configured to prefer using GPRS, positive registration information

Variant C: configured to use only GPRS, negative registration information

Variant D: configured to prefer using GPRS, negative registration information

Variants: <A>...<H>

Preamble:		
	<A>	GSMS610A
	<B>	GSMS610B
	<C>	GSMS610A
	<D>	GSMS610B
	<E>	GSMS602G
	<F>	GSMS602H
	<G>	GSMS610C
	<H>	GSMS610D

```
     MMI                             SMS                              LLC
      |                               |                                 |
MUTE(500)
```

```
(1)    |                                |      GMMSMS_REG_STATE_CNF        |
       |                                *<================================*
       |                                |                                  |
```

Parametrization

	Primitive	Parameter	Value
(1)	GMMSMS_REG_STATE_CNF		
	<A>	reg_state	SMS_RS_REGISTERED
	<B>	reg_state	SMS_RS_REGISTERED
	<C>	reg_state	SMS_RS_DEREGISTERED
	<D>	reg_state	SMS_RS_DEREGISTERED
	<E>	reg_state	SMS_RS_REGISTERED
	<F>	reg_state	SMS_RS_REGISTERED
	<G>	reg_state	SMS_RS_REGISTERED
	<H>	reg_state	SMS_RS_REGISTERED
		radio_priority_level	SMS_RP_LEVEL_4

History:	28-Nov-2000	LW	Initial
	04-Feb-2003	FK	additional Preambles for another scenario

3.4.3 GSMS612: Unit Data Request

Description: The SMS entity has received a positive registration information, therefore it now issues the unitdata request.

Variants: <A>...<F>

Preamble:

<A>	GSMS611A
<B>	GSMS611B
<C>	GSMS611E
<D>	GSMS611F
<E>	GSMS602I
<F>	GSMS602J

```
     MMI                               SMS                               LLC
       |                                |                                  |
(1)    |                                |      LL_UNITDATA_REQ             |
       |                                *================================>*
       |                                |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MO
	cp_user_data_ul	RP_DATA_SUBMIT_ABS
	}	

History:			
	28-Nov-2000	LW	Initial
	04-Feb-2003	FK	additional preambles for another scenarios

3.4.4 GSMS613: Positive Flow Control by LLC

Description: The SMS entity has received a positive registration information, therefore it now issues the unitdata request.

Variants: <A>...<H>

Preamble:

<A>	GSMS612A
<B>	GSMS612B
<C>	GSMS612C
<D>	GSMS612D
<E>	GSMS620E
<F>	GSMS620F
<G>	GSMS620G
<H>	GSMS620H
<I>	GSMS620I
<J>	GSMS620J

```
    MMI                                  SMS                               LLC
     |                                    |                                  |
(1)  |                                    |      LL_UNITREADY_IND            |
     |                                    *<================================*
     |                                    |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History:			
	28-Nov-2000	LW	Initial

3.4.5 GSMS614: Fallback to CCT (GPRS unregistered)

Description: The SMS entity has received a negative registration information, therefore it starts using normal CCT.

Preamble: GSMS611D

```
     MMI                            SMS                             LLC
      |                              |                                |
(1)   |                              |     MMSMS_ESTABLISH_REQ        |
      |                              *===============================>*
(2)   |                              |     MMSMS_ESTABLISH_CNF        |
      |                              *<================================*
(3)   |                              |     MMSMS_DATA_REQ             |
      |                              |       (CP_DATA)                |
      |                              *===============================>*
      |                              |                                |
```

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_DATA_SUBMIT_ABS
		}	

History: 15-Dec-2000 LW Initial

3.4.6 GSMS615: Sending MO-SM, GSMS preferred but not available

Description: The SMS entity is configured to prefer using GPRS for sending mobile originated short messages. The MMI submits a message. After confirmation of the GPRS registration nothing can be sent due to blocked flow .control. After 2 Times TC1M and error is returned to MMI with primitive MNSMS_SUBMIT_CNF. The SMS transaction is closed, nothing has to be sent to the network after flow control is unblocked.

Preamble: GSMS601C

```
     MMI                            SMS                             LLC
      |                              |                                |
(1)   |     MNSMS_SUBMIT_REQ         |                                |
      *=============================>*                                |
(2)   |                              |     GMMSMS_REG_STATE_REQ       |
      |                              *===============================>*
(3)   |                              |     SIM_UPDATE_REQ             |
      |                              *===============================>*
MUTE(500)
(4)   |                              |     GMMSMS_REG_STATE_CNF       |
      |                              *<================================*
MUTE(500)
```

```
(5)   |                              |     SIM_UPDATE_CNF              |
      |                              *<================================*
MUTE(27500)
(6)   |     MNSMS_SUBMIT_CNF         |                                 |
      *<=============================*                                 |
MUTE(500)
(7)   |                              |     LL_UNITREADY_IND            |
      |                              *<================================*
MUTE(500)
COMMAND (SMS STATUS PARTITION)
      |                              |                                 |
```

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) GMMSMS_REG_STATE_REQ		
(3) SIM_UPDATE_REQ		
	source	SRC_SMS
	offset	OFFSET_0
	datafield	SIM_SMSS
	length	SIMREC_SMSS_MSG_REF_LEN
	trans_data	SIMREC_SMSS_MSG_REF
(4) GMMSMS_REG_STATE_CNF		
	reg_state	SMS_RS_REGISTERED
	radio_priority_level	SMS_RP_LEVEL_4
(5) SIM_UPDATE_CNF		
	datafield	SIM_SMSS
	cause	SIM_NO_ERROR
(6) 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
(7) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History:	20-Dec-2000	LW	Initial
	21-May-2002	FK	Adaption to new SAP MNSMS
	20-Jan-2003	FK	Blocked flow control does not affect radio path

3.4.7 GSMS620: 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 'MO-Wait for CP DATA'.

Variants: <A>...<J>

Preamble:		
	<A>	GSMS613A
	<B>	GSMS613B
	<C>	GSMS613C
	<D>	GSMS613D
	<E>	GSMS612A
	<F>	GSMS612B
	<G>	GSMS612C
	<H>	GSMS612D
	<I>	GSMS612E
	<J>	GSMS612F

```
     MMI                              SMS                             LLC
      |                                |                               |
MUTE(500)
(1)   |                                |    LL_UNITDATA_IND            |
      |                                |       (CP_ACK)                |
      |                                *<==============================*
(2)   |                                |    LL_GETUNITDATA_REQ         |
      |                                *==============================>*
      |                                |                               |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MO_TO_MS
	}	
(2) LL_GETUNITDATA_REQ	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History:			
	30-Nov-00	LW	Initial
	04-Feb-2003	FK	additional preambles for another scenarios

3.4.8 GSMS621: Acknowledge of the Infrastructure (GPRS)

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>...<J>

Preamble:		
	<A>	GSMS620A
	<B>	GSMS620B
	<C>	GSMS620C
	<D>	GSMS620D
	<E>	GSMS613E
	<F>	GSMS613F
	<G>	GSMS613G
	<H>	GSMS613H
	<I>	GSMS613I
	<J>	GSMS613J

```
     MMI                             SMS                             LLC
      |                               |                               |
 (1)  |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_DATA)               |
      |                               *<==============================*
 (2)  |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_ACK)                |
      |                               *==============================>*
 (3)  |     MNSMS_SUBMIT_CNF          |                               |
      *<==============================*                               |
 (4)  |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
 (5)  |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
MUTE(500)
COMMAND (SMS STATUS PARTITION)
      |                               |                               |
```

Parametrization

	Primitive	Parameter	Value
(1)	LL_UNITDATA_IND		
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
		reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
		reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
		reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
		reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
		reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
		cipher	LL_CIPHER_OFF
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	D_CP_DATA
		ti	TI_MO_TO_MS
		cp_user_data_dl	RP_ACK_DLNK
		}	

(2) LL_UNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
ll_qos	SMS_DEFAULT_QOS
radio_prio	LL_RADIO_PRIO_4
cipher	LL_CIPHER_OFF
reserved_unitdata_req1	NOT_USED
seg_pos	NOT_USED
attached_counter	NOT_USED
reserved_unitdata_req4	NOT_USED
reserved_unitdata_req5	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) LL_GETUNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

(5) LL_UNITREADY_IND

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

History:	30-Nov-00	LW	Initial
	21-May-2002	FK	Adaption to new SAP MNSMS
	23-Sep-2002	FK	LL_GETUNITDATA_REQ triggered by LL_UNITDATA_IND

3.4.9 GSMS622: Timeout of TC1M and Retransmission

Description: The timer TC1M expires and the CP DATA message is retransmitted.

Variants: <A>...<B>

Preamble: <A> GSMS613A
<B> GSMS613B

```
     MMI                              SMS                            LLC
      |                                |                              |
MUTE(12500)
(1)   |                                |    LL_UNITDATA_REQ           |
      |                                *=============================>*
MUTE(500)
 (5)  |                                |    LL_UNITREADY_IND          |
      |                                *<=============================*
MUTE(500)
      |                                |                              |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MO
	cp_user_data_ul	RP_DATA_SUBMIT_ABS
	}	
(2) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History:	05-Dec-00	LW	Initial
	06-Jun-2002	FK	LL_UNITREADY_IND added

3.4.10 GSMS623: Wrong Message Received (CP-DATA)

Description: The control protocol waits for an CP-ACK message. Instead an CP-DATA message is received. A CP-ERROR message is sent to the infrastucture and the error is signalled to MMI.

Variants: <A>...<B>

Preamble: <A> GSMS613A
<B> GSMS613B

```
    MMI                              SMS                             LLC
     |                                |                               |
 (1) |                                |    LL_UNITDATA_IND            |
     |                                |      (CP_DATA)                |
     |                                *<==============================*
 (2) |                                |    LL_UNITDATA_REQ            |
     |                                |      (CP_ERROR)               |
     |                                *==============================>*
 (3) |     MNSMS_SUBMIT_CNF           |                               |
     *<===============================*                               |
 (4) |                                |    LL_GETUNITDATA_REQ         |
     |                                *==============================>*
MUTE(500)
 (5) |                                |    LL_UNITREADY_IND           |
     |                                *<==============================*
MUTE(500)
COMMAND (SMS STATUS PARTITION)
     |                                |                               |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_DATA_DELIVER_7DEF
	}	
(2) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	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) LL_GETUNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(5) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History:	14-Dec-00	LW	Initial
	03-Jun-2002	FK	Adaption to new SAP MNSMS
	23-Sep-2002	FK	LL_GETUNITDATA_REQ triggered by LL_UNITDATA_IND

3.4.11 GSMS624: Wrong Message Received (unknown)

Description: The control protocol waits for an CP-ACK message. Instead an unknown message is received. The error is signalled to MMI.

Variants: <A>...<B>

Preamble: <A> GSMS613A
<B> GSMS613B

```
     MMI                             SMS                             LLC
      |                               |                               |
 (1)  |                               |    LL_UNITDATA_IND            |
      |                               |      (unknown)                |
      |                               *<==============================*
 (2)  |                               |    LL_UNITDATA_REQ            |
      |                               |      (CP_ERROR)               |
      |                               *==============================>*
 (3)  |     MNSMS_SUBMIT_CNF          |                               |
      *<==============================*                               |
 (4)  |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
 (5)  |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
MUTE(500)
COMMAND (SMS STATUS PARTITION)
      |                               |                               |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	UNKN_SMS_MO_MSG

(2) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	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) LL_GETUNITDATA_REQ	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(5) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History:	14-Dec-00	LW	Initial
	03-Jun-2002	FK	Adaption to new SAP MNSMS
	23-Sep-2002	FK	LL_GETUNITDATA_REQ triggered by LL_UNITDATA_IND

3.4.12 GSMS625: CP Error Received

Description: The control protocol waits for an CP-ACK message. Instead a CP-ERROR message is received. The error is signalled to MMI.

Note: Test case is similar to case SMS029.

Variants: <A>...<B>

Preamble: <A> GSMS613A
<B> GSMS613B

```
     MMI                             SMS                            LLC
      |                               |                              |
 (1)  |                               |   LL_UNITDATA_IND            |
      |                               |     (CP_ERROR)               |
      |                               *<=============================*
 (2)  |     MNSMS_SUBMIT_CNF          |                              |
      *<==============================*                              |
 (3)  |                               |   LL_GETUNITDATA_REQ         |
      |                               *=============================>*
MUTE(500)
COMMAND (SMS STATUS PARTITION)
      |                               |                              |
```

Parametrization

	Primitive	Parameter	Value
(1)	LL_UNITDATA_IND		
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
		reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
		reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
		reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
		reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
		reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
		cipher	LL_CIPHER_OFF
		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)	LL_GETUNITDATA_REQ		
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID

History:	14-Dec-00	LW	Initial
	03-Jun-2002	FK	Adaption to new SAP MNSMS

3.4.13 GSMS626: Acknowledgement, Flow Control not ready

Description: A CP-DATA(RP-ACK) is received for the previously sent RP-DATA. CP-ACK cannot be sent immediately due to lack of flow control. This is done after reception of LL_UNITREADY_IND. Then the procedure ends.

Variants: <A>...<B>

Preamble:

<A>	GSMS620E
<B>	GSMS620F
<C>	GSMS620G
<D>	GSMS620H
<E>	GSMS620I
<F>	GSMS620J

```
     MMI                               SMS                            LLC
      |                                 |                              |
 (1)  |                                 |    LL_UNITDATA_IND           |
      |                                 |       (CP_DATA)              |
      |                                 *<=============================*
 (5)  |                                 |    LL_GETUNITDATA_REQ        |
      |                                 *=============================>*
MUTE(2000)
 (2)  |                                 |    LL_UNITREADY_IND          |
      |                                 *<=============================*
 (3)  |                               |    LL_UNITDATA_REQ           |
      |                               |       (CP_ACK)               |
      |                               *==============================>*
 (4)  |     MNSMS_SUBMIT_CNF          |                              |
      *<==============================*                              |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
      |                               |                              |
```

Parametrization

Primitive	Parameter	Value
(6) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_ACK_DLNK
	}	
(7) LL_GETUNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(8) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(9) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MO
	}	

(10) 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

History:	30-Nov-00	LW	Initial
	21-May-2002	FK	Adaption to new SAP MNSMS
	23-Sep-2002	FK	LL_GETUNITDATA_REQ triggered by LL_UNITDATA_IND

3.4.14 GSMS630: Error Signalled by the Infrastructure

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.

Variants: <A>...<B>

Preamble: <A> GSMS620A
<B> GSMS620B

```
     MMI                              SMS                              LLC
      |                                |                                |
 (1)  |                                |    LL_UNITDATA_IND             |
      |                                |      (CP_DATA)                 |
      |                                *<===============================*
 (2)  |                                |    LL_UNITDATA_REQ             |
      |                                |       (CP_ACK)                 |
      |                                *===============================>*
 (3)  |     MNSMS_SUBMIT_CNF           |                                |
      *<===============================*                                |
 (4)  |                                |    LL_GETUNITDATA_REQ          |
      |                                *===============================>*
MUTE (500)
 (5)  |                                |    LL_UNITREADY_IND            |
      |                                *<===============================*
MUTE (500)
COMMAND (SMS STATUS PARTITION)
      |                                |                                |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_ERR_CONGESTION
	}	
(2) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	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) LL_GETUNITDATA_REQ	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(5) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History:	14-Dec-00	LW	Initial
	03-Jun-2002	FK	Adaption to new SAP MNSMS
	23-Sep-2002	FK	LL_GETUNITDATA_REQ triggered by LL_UNITDATA_IND

3.4.15 GSMS631: 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 protocol. The RP-ERROR message is included into a CP-DATA message and sent to the infrastructure. The error is reported to the user.

Variants: <A>...<B>

Preamble: <A> GSMS620A
<B> GSMS620B

```
     MMI                             SMS                              LLC
      |                               |                               |
 (1)  |                               |    LL_UNITDATA_IND            |
      |                               |      (CP_DATA)                |
      |                               *<==============================*
 (2)  |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_ACK)                |
      |                               *==============================>*
 (3)  |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
 (4)  |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
 (6)  |                               |    LL_UNITDATA_REQ            |
      |                               |      (CP_DATA)                |
      |                               *==============================>*
 (6)  |     MNSMS_SUBMIT_CNF          |                               |
      *<==============================*                               |
MUTE(500)
 (7)  |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
MUTE(500)
 (8)  |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_ACK)                |
      |                               *<==============================*
 (9)  |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
COMMAND (SMS STATUS PARTITION)
      |                               |                               |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_DATA_DELIVER_7DEF
	}	
(2) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MO
	}	
(3) LL_GETUNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(4) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

(5) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MO
	cp_user_data_ul	RP_ERR_PROTOCOL
	}	
(6) 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
(7) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(8) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MO_TO_MS
	}	
(9) LL_GETUNITDATA_REQ	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History:	14-Dec-00	LW	Initial
	03-Jun-2002	FK	Adaption to new SAP MNSMS

3.4.16 GSMS632: Timeout TR1M

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.

Variants: <A>...<B>

Preamble: <A> GSMS620A

<B> GSMS620B

```
     MMI                           SMS                            LLC
      |                              |                              |
TIMEOUT(37500)
 (1)  |     MNSMS_SUBMIT_CNF         |                              |
      *<=============================*                              |
 (2)  |                              |    LL_UNITDATA_REQ           |
      |                              |      (CP_ERROR)              |
      |                              *=============================>*
MUTE(500)
 (3)  |                              |    LL_UNITREADY_IND          |
      |                              *<=============================*
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) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ERROR
	ti	TI_MO
	cp_cause	SMS_CP_CS_PROTOCOL_ERROR
	}	
(3) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History:			
	15-Dec-00	LW	Initial
	03-Jun-2002	FK	Adaption to new SAP MNSMS
	23-Sep-2002	FK	misalignment resolved

3.4.17 GSMS633: Mobile Originated Short Message Command via GPRS

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.

Note: This test case corresponds to case SMS081 in the circuit switched environment.

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>	GSMS602A
<B>	GSMS602A
<C>	GSMS602A
<D>	GSMS621A

```
    MMI                            SMS                            SIM/MM
     |                               |                               |
(1)  |     MNSMS_COMMAND_REQ         |                               |
     *==============================>*                               |
(3)  |                               |    GMMSMS_REG_STATE_REQ       |
     |                               *===============================>*
(4)  |                               |    GMMSMS_REG_STATE_CNF       |
     |                               *<==============================*
(2)  |                               |    SIM_UPDATE_REQ             |
     |                               *==============================>*
(5)  |                               |    LL_UNITDATA_REQ            |
     |                               |       (CP_DATA)               |
     |                               *==============================>*
(6)  |                               |    SIM_UPDATE_CNF             |
     |                               *<==============================*
(6)  |                               |    LL_UNITREADY_IND           |
     |                               *<==============================*
```

```
 (7)  |                               |    LL_UNITDATA_IND           |
      |                               |       (CP_ACK)               |
      |                               *<==============================*
 (8)  |                               |    LL_GETUNITDATA_REQ        |
      |                               *==============================>*
 (9)  |                               |    LL_UNITDATA_IND           |
      |                               |       (CP_DATA)              |
      |                               *<==============================*
 (10) |                               |    LL_UNITDATA_REQ           |
      |                               |       (CP_ACK)               |
      |                               *==============================>*
 (11) |     MNSMS_COMMAND_CNF         |                               |
      *<==============================*                               |
 (13) |                               |    LL_GETUNITDATA_REQ        |
      |                               *==============================>*
MUTE(500)
 (14) |                               |    LL_UNITREADY_IND          |
      |                               *<==============================*
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) GMMSMS_REG_STATE_REQ		
(3) GMMSMS_REG_STATE_CNF		
	reg_state	SMS_RS_REGISTERED
	radio_priority_level	SMS_RP_LEVEL_4
(4) 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

(5) LL_UNITDATA_REQ

	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	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) SIM_UPDATE_CNF

datafield	SIM_SMSS
cause	SIM_NO_ERROR

(7) LL_UNITREADY_IND

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

(8) LL_UNITDATA_IND

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
cipher	LL_CIPHER_OFF
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MO_TO_MS
}	

(9) LL_GETUNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

(10) LL_UNITDATA_IND

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
cipher	LL_CIPHER_OFF
sdu	
{	
component	SMS
direction	DOWNLINK
pd	D_CP_DATA
ti	TI_MO_TO_MS
cp_user_data_dl	RP_ACK_DLNK
}	

(11) LL_UNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
ll_qos	SMS_DEFAULT_QOS
radio_prio	LL_RADIO_PRIO_4
cipher	LL_CIPHER_OFF
reserved_unitdata_req1	NOT_USED
seg_pos	NOT_USED
attached_counter	NOT_USED
reserved_unitdata_req4	NOT_USED
reserved_unitdata_req5	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MO
}	

(12) 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

(13) LL_GETUNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

(14) LL_UNITREADY_IND

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

History:	6-Jan-98	SZ	Initial
	31-Oct-00	LW	Added variants B - D
	20-Dec-00	LW	New test case GSMS633 using GPRS dl
	03-Jun-2002	FK	Adaption to new SAP MNSMS
	23-Sep-2002	FK	LL_GETUNITDATA_REQ triggered by LL_UNITDATA_IND

3.5 Mobile Terminated SM via GPRS

3.5.1 GSMS650: MT-SM Indicated by LLC

Description: A LL UNITDATA PDU is received which contains a RP DATA message with a mobile terminated SM which is stored in the SIM (similar to test case SMS019 in circuit switched mode). Preambles G,H test the repetition of SM reception via GPRS

Variants: <A>...<G>

Preamble:

<A>	GSMS602A
<B>	GSMS602B
<C>	GSMS602C
<D>	GSMS602D
<E>	GSMS602E
<F>	GSMS662A
<G>	GSMS662B

```
     MMI                             SMS                            LLC
      |                               |                               |
 (1)  |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_DATA)               |
      |                               *<==============================*
 (2)  |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_ACK)                |
      |                               *==============================>*
 (3)  |                               |    SIM_UPDATE_RECORD_REQ      |
      |                               *==============================>*
 (4)  |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
      |                               |                               |
```

Parametrization

	Primitive	Parameter	Value
(1)	LL_UNITDATA_IND		
		sapi	LL_SAPI_7
		tlli	DEF_SMS_LL_TLLI_1
		reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
		reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
		reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
		reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
		reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
		cipher	LL_CIPHER_OFF
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	D_CP_DATA
		ti	TI_MT
		cp_user_data_dl	RP_DATA_DELIVER_7DEF
		}	

(2) LL_UNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
ll_qos	SMS_DEFAULT_QOS
radio_prio	LL_RADIO_PRIO_4
cipher	LL_CIPHER_OFF
reserved_unitdata_req1	NOT_USED
seg_pos	NOT_USED
attached_counter	NOT_USED
reserved_unitdata_req4	NOT_USED
reserved_unitdata_req5	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_2
<G>	record	SIM_RECORD_2
	length	LENGTH_SMS
	linear_data	SIM_SMS_MT_DELIVER_7DEF

(4) LL_GETUNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

History:	05-Dec-00	LW	Initial
	23-Sep-2002	FK	LL_GETUNITDATA_REQ triggered by LL_UNITDATA_IND
	06-Feb-2003	FK	LL_UNITREADY_IND removed; preambles added

3.5.2 GSMS651: Refuse Second Mobile Terminated Connection: MM Downlink

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: GSMS650A

```
     MMI                             SMS                           SIM/MM
      |                               |                               |
 (1)  |                               |    MMSMS_ESTABLISH_IND        |
      |                               |         (CP_DATA)             |
      |                               *<===============================*
 (2)  |                               |    MMSMS_DATA_REQ             |
      |                               |      (CP_DATA)                |
      |                               *===============================>*
 (3)  |                               |    MMSMS_RELEASE_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_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
	6-Dec-00	LW	modified for GPRS
	06-Feb-2003	FK	Start with LL_UNITREADY_IND

3.5.3 GSMS652: Refuse Second Mobile Terminated Connection: LL Downlink

Description: Another LL UNITDATA PDU is received which contains a RP DATA message with a mobile terminated SM. Since only one mobile terminated connection is allowed, RP ERROR message is sent and the SM is not accepted.

Preamble: GSMS650B

```
     MMI                             SMS                             LLC
      |                               |                               |
 (5)  |                               |    LL_UNITREADY_IND           |
      |                               *<===============================*
MUTE(500)
```

```
 (1)  |                                |    LL_UNITDATA_IND              |
      |                                |       (CP_DATA)                 |
      |                                *<===============================*
 (2)  |                                |    LL_UNITDATA_REQ              |
      |                                |       (CP_ACK)                  |
      |                                *===============================>*
 (3)  |                                |    LL_GETUNITDATA_REQ           |
      |                                *===============================>*
MUTE(500)
 (4)  |                                |    LL_UNITREADY_IND             |
      |                                *<===============================*
      |                                |                                 |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(2) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	DEF_SMS_LL_TLLI_1
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT_2
	cp_user_data_dl	RP_DATA_DELIVER_2
	}	
(3) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	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
	}	

Primitive	Parameter	Value
(4) LL_GETUNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(5) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History:	06-Dec-00	LW	Initial
	23-Sep-2002	FK	LL_GETUNITDATA_REQ triggered by LL_UNITDATA_IND

3.5.4 GSMS653: Reception of a CP-ERROR Message

Description: The SMS entity has no established connections but receives a LL_UNITDATA_IND. A new instance is created for this transaction. The incoming message is a CP-ERROR message. The message is ignored. The instance is freed.

Note: Corresponds to SMS111 in circuit switched case.

Preamble: GSMS602A

```
    MMI                                  SMS                                LLC
     |                                    |                                  |
 (1) |                                    |     LL_UNITDATA_IND              |
     |                                    |        (CP_ERROR)                |
     |                                    *<=================================*
 (2) |                                    |     LL_GETUNITDATA_REQ           |
     |                                    *=================================>*
MUTE(500)
     |                                    |                                  |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	DEF_SMS_LL_TLLI_1
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ERROR
	ti	TI_MT
	cp_cause	SMS_CP_CS_NETWORK_FAILURE
	}	
(2) LL_GETUNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History:	20-Dec-00	LW	Initial
	06-Jun-2002	FK	Adaption to new SAP MNSMS

3.5.5 GSMS654: Reception of a CP-ACK Message

Description: The SMS entity has no established connections but receives a LL_UNITDATA_IND. A new instance is created for this transaction. The incoming message is a CP-ACK message. The message is ignored. The instance is freed.

Note: Corresponds to SMS112 in circuit switched case.

Preamble: GSMS602A

```
     MMI                              SMS                             LLC
      |                                |                               |
 (5)  |                                |    LL_UNITREADY_IND           |
      |                                *<==============================*
MUTE(500)
 (1)  |                                |    LL_UNITDATA_IND            |
      |                                |       (CP_ACK)                |
      |                                *<==============================*
 (2)  |                                |    LL_UNITDATA_REQ            |
      |                                *==============================>*
 (3)  |                                |    LL_GETUNITDATA_REQ         |
      |                                *==============================>*
MUTE(500)
 (4)  |                                |    LL_UNITREADY_IND           |
      |                                *<==============================*
MUTE(2000)
      |                                |                               |
```

Parametrization

	Primitive	Parameter	Value
(1)	LL_UNITREADY_IND		
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
(2)	LL_UNITDATA_IND		
		sapi	LL_SAPI_7
		tlli	DEF_SMS_LL_TLLI_1
		reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
		reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
		reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
		reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
		reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
		cipher	LL_CIPHER_OFF
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	

(3) LL_UNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
ll_qos	SMS_DEFAULT_QOS
radio_prio	LL_RADIO_PRIO_4
cipher	LL_CIPHER_OFF
reserved_unitdata_req1	NOT_USED
seg_pos	NOT_USED
attached_counter	NOT_USED
reserved_unitdata_req4	NOT_USED
reserved_unitdata_req5	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ERROR
ti	TI_MT_FROM_MS
cp_cause	SMS_CP_CS_MSG_NOT_COMP
}	

(4) LL_GETUNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

(5) LL_UNITREADY_IND

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

History:	20-Dec-00	LW	Initial
	06-Jun-2002	FK	Adaption to new SAP MNSMS
	23-Sep-2002	FK	LL_GETUNITDATA_REQ triggered by LL_UNITDATA_IND
	06-Feb-2003	FK	Start with LL_UNITREADY_IND

3.5.6 GSMS655: Refuse Second MT SM on LL Downlink, Flow Control not ready

Description: Another LL UNITDATA PDU is received which contains a RP DATA message with a mobile terminated SM. Since only one mobile terminated connection is allowed, the SM cannot not be accepted. Flow control not ready prevents sending of an error response.

Preamble: GSMS650B

```
     MMI                               SMS                              LLC
      |                                 |                                |
MUTE(500)
 (1)  |                                 |    LL_UNITDATA_IND             |
      |                                 |       (CP_DATA)                |
      |                                 *<==============================*
 (3)  |                                 |    LL_GETUNITDATA_REQ          |
      |                                 *==============================>*
MUTE(500)
      |                                 |                                |
```

Parametrization

Primitive	Parameter	Value
(6) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	DEF_SMS_LL_TLLI_1
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT_2
	cp_user_data_dl	RP_DATA_DELIVER_2
	}	
(7) LL_GETUNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History: 06-Feb-2003 FK Initial

3.5.7 GSMS656: Reception of a CP-ACK Message, Flow Control not ready

Description: The SMS entity has no established connections but receives a LL_UNITDATA_IND. A new instance is created for this transaction. The incoming message is a CP-ACK message. The message is ignored. The instance is freed.

Note: Corresponds to SMS112 in circuit switched case.

Preamble: GSMS601A

```
     MMI                                SMS                              LLC
      |                                  |                                |
MUTE(500)
 (1)  |                                  |    LL_UNITDATA_IND             |
      |                                  |       (CP_ACK)                 |
      |                                  *<===============================*
 (3)  |                                  |    LL_GETUNITDATA_REQ          |
      |                                  *===============================>*
MUTE(500)
 (4)  |                                  |    LL_UNITREADY_IND            |
      |                                  *<===============================*
 (2)  |                                  |    LL_UNITDATA_REQ             |
      |                                  |       (CP_ERROR)               |
      |                                  *===============================>*
      |                                  |                                |
```

Parametrization

Primitive	Parameter	Value
(6) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	DEF_SMS_LL_TLLI_1
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MT
	}	
(7) LL_GETUNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(8) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(9) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ERROR
	ti	TI_MT_FROM_MS
	cp_cause	SMS_CP_CS_MSG_NOT_COMP
	}	

History: 06-Feb-2003 FK Initial

3.5.8 GSMS660: Writing to SIM Successful, RP-ACK to Infrastructure

Description: The MT-SM has been successfully written to the SIM, therefore an RP acknowledgement is sent.

Variants: <A>...<C>

Preamble:

<A>	GSMS650C
<B>	GSMS651
<C>	GSMS655

```
   SIM/ACI                           SMS                            LLC
      |                               |                               |
 (5)  |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
MUTE(500)
 (1)  |     SIM_UPDATE_RECORD_CNF     |                               |
      *==============================>*                               |
 (2)  |     MNSMS_MESSAGE_IND         |                               |
      *<==============================*                               |
 (3)  |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_DATA)               |
      |                               *==============================>*
MUTE(500)
 (4)  |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
      |                               |                               |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(2) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_SMS
	record	SIM_RECORD_1
	cause	SIM_NO_ERROR
(3) 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_7DEF
(4) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ACK_ULNK
	}	
(5) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History: 06-Dec-00 LW Initial

3.5.9 GSMS661: Writing to SIM Successful, RP-ACK has to wait for Flow Control

Description: The MT-SM has been successfully written to the SIM, therefore an RP acknowledgement is sent.

Preamble: GSMS650C

```
   SIM/ACI                           SMS                            LLC
      |                               |                               |
 (1)  |     SIM_UPDATE_RECORD_CNF     |                               |
      *==============================>*                               |
 (2)  |     MNSMS_MESSAGE_IND         |                               |
      *<==============================*                               |
MUTE(500)
 (3)  |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
 (4)  |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_DATA)               |
      |                               *==============================>*
      |                               |                               |
```

Parametrization

Primitive	Parameter	Value
(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_7DEF
(8) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(9) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	U_CP_DATA
	ti	TI_MT_FROM_MS
	cp_user_data_ul	RP_ACK_ULNK
	}	

History: 06-Dec-00 LW Initial

3.5.10 GSMS662: Receive CP-ACK for RP-ACK

Description: The RP-ACK data request has been successfully received by the peer entity and an acknowledgement for it is received in the CP layer.

Variants: <A>...<B>

Preamble: <A> GSMS660A
<B> GSMS661

```
     SIM                              SMS                            LLC
      |                                |                              |
MUTE(500)
 (1)  |                                |    LL_UNITREADY_IND          |
      |                                *<=============================*
MUTE(500)
 (2)  |                                |    LL_UNITDATA_IND           |
      |                                |       (CP_ACK)               |
      |                                *<=============================*
 (3)  |                                |    LL_GETUNITDATA_REQ        |
      |                                *=============================>*
MUTE(2000)
      |                                |                              |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(2) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MT
	}	
(3) LL_GETUNITDATA_REQ	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

History: 12-Dec-00 LW Initial

3.5.11 GSMS663: Receiving MT-SM (Class 0), Flow Control not ready

Description: A MT-SM (Class 0) is received. Acknowledgement cannot be sent, because flow control is not ready. Responses have to be sent via LLC, disregarding the preamble setting of CSD only for MO-SM.

Preamble: GSMS601B

```
   MMI                              SMS                              LLC
    |                                |                                |
(2)  |                               |         LL_UNITDATA_IND        |
    |                                |            (CP_DATA)           |
    |                                * <============================= *
(3)  |       MNSMS_MESSAGE_IND       |                                |
    * <============================= *                                |
(4)  |                               |       LL_GETUNITDATA_REQ       |
    |                                * =============================> *
MUTE(500)
(5)  |                               |        LL_UNITREADY_IND        |
    |                                * <============================= *
(6)  |                               |         LL_UNITDATA_REQ        |
    |                                |             (CP_ACK)           |
    |                                * =============================> *
MUTE(500)
(7)  |                               |        LL_UNITREADY_IND        |
    |                                * <============================= *
(8)  |                               |         LL_UNITDATA_REQ        |
    |                                |            (CP_DATA)           |
    |                                * =============================> *
MUTE(500)
    |                                |                                |
```

Parametrization

Primitive	Parameter	Value
(4) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	DEF_SMS_LL_TLLI_1
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL0_DEF
	}	
(5) MNSMS_MESSAGE_IND		
	mem_type	NOT_PRESENT_8BIT
	rec_num	NOT_USED
	rec_max	NOT_USED
	status	SMS_RECORD_REC_UNREAD
	sms_sdu	SMS_SDU_DELIVER_7CL0_DEF
(6) LL_GETUNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(7) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

(8) LL_UNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
ll_qos	SMS_DEFAULT_QOS
radio_prio	LL_RADIO_PRIO_4
cipher	LL_CIPHER_OFF
reserved_unitdata_req1	NOT_USED
seg_pos	NOT_USED
attached_counter	NOT_USED
reserved_unitdata_req4	NOT_USED
reserved_unitdata_req5	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MT_FROM_MS
}	

(9) LL_UNITREADY_IND

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

(10) LL_UNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
ll_qos	SMS_DEFAULT_QOS
radio_prio	LL_RADIO_PRIO_4
cipher	LL_CIPHER_OFF
reserved_unitdata_req1	NOT_USED
seg_pos	NOT_USED
attached_counter	NOT_USED
reserved_unitdata_req4	NOT_USED
reserved_unitdata_req5	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MT_FROM_MS
cp_user_data_ul	RP_ACK_RESP
}	

History: 06-Feb-2003 FK Initial

3.6 Interworking MO-SM with MT-SM in GSMS

3.6.1 GSMS670: Sending of MO-SM parallel to Reception of Class 1 Message

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. (similar to test case SMS051).

Variants: <A>...<B>

Preamble: <A> GSMS613A
<B> GSMS613B

```
     SIM                             SMS                            LLC
      |                               |                               |
 (1)  |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_DATA: MT)           |
      |                               *<==============================*
 (2)  |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_ACK: MT)            |
      |                               *==============================>*
 (3)  |                               |    SIM_UPDATE_RECORD_REQ      |
      |                               *==============================>*
 (4)  |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
 (5)  |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
 (6)  |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_ACK: MO)            |
      |                               *<==============================*
 (7)  |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
 (8)  |                               |    SIM_UPDATE_RECORD_CNF      |
      |                               *<==============================*
 (9)  |     MNSMS_MESSAGE_IND         |                               |
      *<==============================*                               |
 (10) |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_DATA: MT)           |
      |                               *==============================>*
MUTE(500)
 (11) |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
MUTE(500)
 (12) |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_ACK: MT)            |
      |                               *<==============================*
 (13) |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
 (14) |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_DATA: MO)           |
      |                               *<==============================*
 (15) |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_ACK: MO)            |
      |                               *==============================>*
 (16) |     MNSMS_SUBMIT_CNF          |                               |
      *<==============================*                               |
 (17) |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
 (18) |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
MUTE(500)
COMMAND (SMS STATUS PARTITION)
      |                               |                               |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL1
	}	
(2) LL_UNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	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) LL_GETUNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(5) LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

(6)	LL_UNITDATA_IND		
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
		reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
		reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
		reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
		reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
		reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
		cipher	LL_CIPHER_OFF
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MO_TO_MS
		}	
(7)	LL_GETUNITDATA_REQ	sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
(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
		sms_sdu	SMS_SDU_DELIVER_7CL1
(10)		LL_UNITDATA_REQ	
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
		ll_qos	SMS_DEFAULT_QOS
		radio_prio	LL_RADIO_PRIO_4
		cipher	LL_CIPHER_OFF
		reserved_unitdata_req1	NOT_USED
		seg_pos	NOT_USED
		attached_counter	NOT_USED
		reserved_unitdata_req4	NOT_USED
		reserved_unitdata_req5	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	U_CP_DATA
		ti	TI_MT_FROM_MS
		cp_user_data_ul	RP_ACK_ULNK
		}	
(11)		LL_UNITREADY_IND	
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID

(12)

```
LL_UNITDATA_IND
sapi                    LL_SAPI_7
tlli                    LL_TLLI_INVALID
reserved_unitdata_ind1  DEF_RES_UNITDATA_IND1
reserved_unitdata_ind2  DEF_RES_UNITDATA_REQ1
reserved_unitdata_ind3  DEF_RES_UNITDATA_IND3
reserved_unitdata_ind4  DEF_RES_UNITDATA_IND4
reserved_unitdata_ind5  DEF_RES_UNITDATA_IND5
cipher                  LL_CIPHER_OFF
sdu
{
component               SMS
direction               DOWNLINK
pd                      B_CP_ACK
ti                      TI_MT
}
```

(13)

```
LL_GETUNITDATA_REQ    sapi      LL_SAPI_7
tlli                    LL_TLLI_INVALID
```

(14)

```
LL_UNITDATA_IND
sapi                    LL_SAPI_7
tlli                    LL_TLLI_INVALID
reserved_unitdata_ind1  DEF_RES_UNITDATA_IND1
reserved_unitdata_ind2  DEF_RES_UNITDATA_REQ1
reserved_unitdata_ind3  DEF_RES_UNITDATA_IND3
reserved_unitdata_ind4  DEF_RES_UNITDATA_IND4
reserved_unitdata_ind5  DEF_RES_UNITDATA_IND5
cipher                  LL_CIPHER_OFF
sdu
{
component               SMS
direction               DOWNLINK
pd                      D_CP_DATA
ti                      TI_MO_TO_MS
cp_user_data_dl         RP_ACK_DLNK
}
```

(15)

```
LL_UNITDATA_REQ
sapi                    LL_SAPI_7
tlli                    LL_TLLI_INVALID
ll_qos                  SMS_DEFAULT_QOS
radio_prio              LL_RADIO_PRIO_4
cipher                  LL_CIPHER_OFF
reserved_unitdata_req1  NOT_USED
seg_pos                 NOT_USED
attached_counter        NOT_USED
reserved_unitdata_req4  NOT_USED
reserved_unitdata_req5  NOT_USED
sdu
{
component               SMS
direction               UPLINK
pd                      B_CP_ACK
ti                      TI_MO
}
```

```
(16)                 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

(17)                 LL_GETUNITDATA_REQ  sapi      LL_SAPI_7
                     tlli                LL_TLLI_INVALID

(18)                 LL_UNITREADY_IND    sapi LL_SAPI_7
                     tlli                LL_TLLI_INVALID
```

History:	15-Dec-00	LW	Initial
	24-Sep-2002	FK	LL_GETUNITDATA_REQ triggered by LL_UNITDATA_IND

3.6.2 GSMS671: Handling of Flow Control Interference (Case 1)

Description: Same scenario as TC 670, but both the MO and MT operation got stuck for a short time due to a temporary stop of flow control provided by LLC. Both transactions shall carry on after receiving the READY signal, first the one which has no supervisory timer running.

Preamble: GSMS613A

```
     SIM                             SMS                             LLC
      |                               |                               |
 (1)  |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_DATA: MT)           |
      |                               *<==============================*
 (2)  |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_ACK: MT)            |
      |                               *==============================>*
 (3)  |                               |    SIM_UPDATE_RECORD_REQ      |
      |                               *==============================>*
 (4)  |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
 (5)  |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_ACK: MO)            |
      |                               *<==============================*
 (6)  |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
 (7)  |                               |    SIM_UPDATE_RECORD_CNF      |
      |                               *<==============================*
 (8)  |     MNSMS_MESSAGE_IND         |                               |
      *<==============================*                               |
MUTE(2000)
 (9)  |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
 (10) |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_DATA: MT)           |
      |                               *==============================>*
MUTE(500)
 (11) |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_ACK: MT)            |
      |                               *<==============================*
 (12) |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
```

```
 (13) |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_DATA: MO)           |
      |                               *<==============================*
 (15) |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(2000)
 (16) |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
 (17) |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_ACK: MO)            |
      |                               *==============================>*
 (14) |     MNSMS_SUBMIT_CNF          |                               |
      *<==============================*                               |
MUTE(500)
 (18) |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
MUTE(500)
COMMAND (SMS STATUS PARTITION)
      |                               |                               |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL1
	}	

（2）	LL_UNITDATA_REQ		
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
		ll_qos	SMS_DEFAULT_QOS
		radio_prio	LL_RADIO_PRIO_4
		cipher	LL_CIPHER_OFF
		reserved_unitdata_req1	NOT_USED
		seg_pos	NOT_USED
		attached_counter	NOT_USED
		reserved_unitdata_req4	NOT_USED
		reserved_unitdata_req5	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）	LL_GETUNITDATA_REQ		
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
（5）	LL_UNITDATA_IND		
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
		reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
		reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
		reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
		reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
		reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
		cipher	LL_CIPHER_OFF
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MO_TO_MS
		}	
（6）	LL_GETUNITDATA_REQ	sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
（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)	LL_UNITREADY_IND		
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
(10)		LL_UNITDATA_REQ	
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
		ll_qos	SMS_DEFAULT_QOS
		radio_prio	LL_RADIO_PRIO_4
		cipher	LL_CIPHER_OFF
		reserved_unitdata_req1	NOT_USED
		seg_pos	NOT_USED
		attached_counter	NOT_USED
		reserved_unitdata_req4	NOT_USED
		reserved_unitdata_req5	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	U_CP_DATA
		ti	TI_MT_FROM_MS
		cp_user_data_ul	RP_ACK_ULNK
		}	
(11)		LL_UNITDATA_IND	
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
		reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
		reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
		reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
		reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
		reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
		cipher	LL_CIPHER_OFF
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MT
		}	
(12)		LL_GETUNITDATA_REQ	sapi LL_SAPI_7
		tlli	LL_TLLI_INVALID

(13) LL_UNITDATA_IND

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
cipher	LL_CIPHER_OFF
sdu	
{	
component	SMS
direction	DOWNLINK
pd	D_CP_DATA
ti	TI_MO_TO_MS
cp_user_data_dl	RP_ACK_DLNK
}	

(14) LL_GETUNITDATA_REQ sapi LL_SAPI_7

tlli	LL_TLLI_INVALID

(15) LL_UNITREADY_IND

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

(16) LL_UNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
ll_qos	SMS_DEFAULT_QOS
radio_prio	LL_RADIO_PRIO_4
cipher	LL_CIPHER_OFF
reserved_unitdata_req1	NOT_USED
seg_pos	NOT_USED
attached_counter	NOT_USED
reserved_unitdata_req4	NOT_USED
reserved_unitdata_req5	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MO
}	

(17) 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

(18) LL_UNITREADY_IND sapi LL_SAPI_7

tlli	LL_TLLI_INVALID

History:	24-Sep-2002	FK	Initial
	06-Feb-2003	FK	Interworking of flow control changed

3.6.3 GSMS672: Handling of Flow Control Interference (Case 2)

Description: Same scenario as TC 670, but both the MO and MT operation got stuck for a short time due to a temporary stop of flow control provided by LLC. Both transactions shall carry on after receiving the READY signal, first the one which has no supervisory timer running. If this is not a significant difference, then the MO transaction is served first.

Preamble: GSMS612A

```
     SIM                             SMS                            LLC
      |                               |                               |
 (1)  |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_DATA: MT)           |
      |                               *<==============================*
 (2)  |                               |    SIM_UPDATE_RECORD_REQ      |
      |                               *==============================>*
 (3)  |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
 (4)  |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_ACK: MO)            |
      |                               *<==============================*
 (5)  |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(1000)
 (6)  |                               |    SIM_UPDATE_RECORD_CNF      |
      |                               *<==============================*
 (7)  |     MNSMS_MESSAGE_IND         |                               |
      *<==============================*                               |
MUTE(1000)
 (8)  |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
 (9)  |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_ACK: MT)            |
      |                               *==============================>*
MUTE(500)
 (10) |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
 (11) |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_DATA: MT)           |
      |                               *==============================>*
MUTE(500)
 (12) |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_DATA: MO)           |
      |                               *<==============================*
 (14) |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(1000)
 (15) |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_ACK: MT)            |
      |                               *<==============================*
 (16) |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
 (17) |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
 (18) |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_ACK: MO)            |
      |                               *==============================>*
 (13) |     MNSMS_SUBMIT_CNF          |                               |
      *<==============================*                               |
MUTE(500)
```

```
 (19) |                                |    LL_UNITREADY_IND           |
      |                                *<==============================*
MUTE(500)
COMMAND (SMS STATUS PARTITION)
      |                                |                               |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MT
	cp_user_data_dl	RP_DATA_DELIVER_7CL1
	}	
(2) SIM_UPDATE_RECORD_REQ		
	source	SRC_SMS
	datafield	SIM_SMS
	record	SIM_RECORD_1
	length	LENGTH_SMS
	linear_data	SIM_SMS_CLASS_1
(3) LL_GETUNITDATA_REQ		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
(4) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	B_CP_ACK
	ti	TI_MO_TO_MS
	}	
(5) LL_GETUNITDATA_REQ	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

```
(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)   LL_UNITREADY_IND
                              sapi                    LL_SAPI_7
                              tlli                    LL_TLLI_INVALID

(9)   LL_UNITDATA_REQ
                              sapi                    LL_SAPI_7
                              tlli                    LL_TLLI_INVALID
                              ll_qos                  SMS_DEFAULT_QOS
                              radio_prio              LL_RADIO_PRIO_4
                              cipher                  LL_CIPHER_OFF
                              reserved_unitdata_req1  NOT_USED
                              seg_pos                 NOT_USED
                              attached_counter        NOT_USED
                              reserved_unitdata_req4  NOT_USED
                              reserved_unitdata_req5  NOT_USED
                              sdu
                              {
                              component               SMS
                              direction               UPLINK
                              pd                      B_CP_ACK
                              ti                      TI_MT_FROM_MS
                              }

(10)                          LL_UNITREADY_IND
                              sapi                    LL_SAPI_7
                              tlli                    LL_TLLI_INVALID

(11)                          LL_UNITDATA_REQ
                              sapi                    LL_SAPI_7
                              tlli                    LL_TLLI_INVALID
                              ll_qos                  SMS_DEFAULT_QOS
                              radio_prio              LL_RADIO_PRIO_4
                              cipher                  LL_CIPHER_OFF
                              reserved_unitdata_req1  NOT_USED
                              seg_pos                 NOT_USED
                              attached_counter        NOT_USED
                              reserved_unitdata_req4  NOT_USED
                              reserved_unitdata_req5  NOT_USED
                              sdu
                              {
                              component               SMS
                              direction               UPLINK
                              pd                      U_CP_DATA
                              ti                      TI_MT_FROM_MS
                              cp_user_data_ul         RP_ACK_ULNK
                              }
```

(12)

```
LL_UNITDATA_IND
sapi                   LL_SAPI_7
tlli                   LL_TLLI_INVALID
reserved_unitdata_ind1 DEF_RES_UNITDATA_IND1
reserved_unitdata_ind2 DEF_RES_UNITDATA_REQ1
reserved_unitdata_ind3 DEF_RES_UNITDATA_IND3
reserved_unitdata_ind4 DEF_RES_UNITDATA_IND4
reserved_unitdata_ind5 DEF_RES_UNITDATA_IND5
cipher                 LL_CIPHER_OFF
sdu
{
component              SMS
direction              DOWNLINK
pd                     D_CP_DATA
ti                     TI_MO_TO_MS
cp_user_data_dl        RP_ACK_DLNK
}
```

(13)

```
LL_GETUNITDATA_REQ  sapi    LL_SAPI_7
tlli                   LL_TLLI_INVALID
```

(14)

```
LL_UNITDATA_IND
sapi                   LL_SAPI_7
tlli                   LL_TLLI_INVALID
reserved_unitdata_ind1 DEF_RES_UNITDATA_IND1
reserved_unitdata_ind2 DEF_RES_UNITDATA_REQ1
reserved_unitdata_ind3 DEF_RES_UNITDATA_IND3
reserved_unitdata_ind4 DEF_RES_UNITDATA_IND4
reserved_unitdata_ind5 DEF_RES_UNITDATA_IND5
cipher                 LL_CIPHER_OFF
sdu
{
component              SMS
direction              DOWNLINK
pd                     B_CP_ACK
ti                     TI_MT
}
```

(15)

```
LL_GETUNITDATA_REQ  sapi    LL_SAPI_7
tlli                   LL_TLLI_INVALID
```

(16)

```
LL_UNITREADY_IND
sapi                   LL_SAPI_7
tlli                   LL_TLLI_INVALID
```

(17)	LL_UNITDATA_REQ	
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID
	ll_qos	SMS_DEFAULT_QOS
	radio_prio	LL_RADIO_PRIO_4
	cipher	LL_CIPHER_OFF
	reserved_unitdata_req1	NOT_USED
	seg_pos	NOT_USED
	attached_counter	NOT_USED
	reserved_unitdata_req4	NOT_USED
	reserved_unitdata_req5	NOT_USED
	sdu	
	{	
	component	SMS
	direction	UPLINK
	pd	B_CP_ACK
	ti	TI_MO
	}	
(18)	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
(19)	LL_UNITREADY_IND	sapi LL_SAPI_7
	tlli	LL_TLLI_INVALID

History:	24-Sep-2002	FK	Initial
	06-Feb-2003	FK	Interworking of flow control changed

3.7 Memory Management Procedure

3.7.1 GSMS681: Memory Full, RP-SMMA via GPRS

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 and after receiveing an RP-ACK updates EF(SMSS).

Note: Corresponds to SMS017 in circuit switched case.

Preamble: GSMS602F

```
    MMI                             SMS                             LLC
      |                               |                               |
 (1)  |                               |    SIM_SMS_INSERT_IND         |
      |                               *<==============================*
(9)   |       MNSMS_REPORT_IND        |                               |
     *<===============================*                               |
(10)  |                               |          SIM_READ_REQ         |
      |                               *  ============================>*
MUTE(500)
(11)  |                               |          SIM_READ_CNF         |
      |                               *  <============================*
(12)  |                               |          SIM_READ_REQ         |
      |                               *  ============================>*
MUTE(500)
 (1)  |                               |          SIM_READ_CNF         |
      |                               * <============================ *
 (2)  |                               |    SIM_READ_RECORD_REQ        |
      |                               *==============================>*
 (3)  |                              |    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      |                              |
     *<==============================*                              |
 (9)  |                              |    GMMSMS_REG_STATE_REQ      |
      |                              *==============================>*
(10)  |                             |     GMMSMS_REG_STATE_CNF     |
      |                             *<==============================*
(5)   |                              |    LL_GETUNITDATA_REQ       |
      |                              *===============================>*
(11)  |                              |    LL_UNITDATA_REQ           |
      |                              |      (CP_DATA)               |
      |                              *==============================>*
(12)  |                              |    LL_UNITDATA_IND           |
      |                              |       (CP_ACK)               |
      |                              *<==============================*
(13)  |                              |    LL_GETUNITDATA_REQ        |
      |                              *==============================>*
MUTE(500)
(14)  |                              |    LL_UNITREADY_IND          |
      |                              *<==============================*
      |                              |                              |
```

Parametrization

	Primitive	Parameter	Value
(1)	SIM_SMS_INSERT_IND		
		phase	PHASE_2_SIM
		tp_mr	TP_MR_3
		mem_cap_avail	FALSE
		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
		cause	SIM_CAUSE_UNKN_FILE_ID
		length	LEN_0
		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
trans_data	IMSI_NORMAL

(7) SIM_READ_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_1
length	LENGTH_SMS

(8) 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

(9) 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

(10) SIM_READ_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_2
length	LENGTH_SMS

(11) 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

(12) SIM_READ_RECORD_REQ

source	SRC_SMS
datafield	SIM_SMS
record	SIM_RECORD_3
length	LENGTH_SMS

(13) 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

(14)	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
(15)	GMMSMS_REG_STATE_REQ		
(16)	GMMSMS_REG_STATE_CNF		
		reg_state	SMS_RS_REGISTERED
		radio_priority_level	SMS_RP_LEVEL_4
(17)	LL_GETUNITDATA_REQ		
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
(18)	LL_UNITDATA_REQ		
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
		ll_qos	SMS_DEFAULT_QOS
		radio_prio	LL_RADIO_PRIO_4
		cipher	LL_CIPHER_OFF
		reserved_unitdata_req1	NOT_USED
		seg_pos	NOT_USED
		attached_counter	NOT_USED
		reserved_unitdata_req4	NOT_USED
		reserved_unitdata_req5	NOT_USED
		sdu	
		{	
		component	SMS
		direction	UPLINK
		pd	U_CP_DATA
		ti	TI_MO
		cp_user_data_ul	RP_SMMA
		}	
(19)	LL_UNITDATA_IND		
		sapi	LL_SAPI_7
		tlli	DEF_SMS_LL_TLLI_1
		reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
		reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
		reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
		reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
		reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
		cipher	LL_CIPHER_OFF
		sdu	
		{	
		component	SMS
		direction	DOWNLINK
		pd	B_CP_ACK
		ti	TI_MO_TO_MS
		}	
(20)	LL_GETUNITDATA_REQ		
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID
(21)	LL_UNITREADY_IND		
		sapi	LL_SAPI_7
		tlli	LL_TLLI_INVALID

History:	20-Dec-00	LW	Initial
	06-Jun-2002	FK	Adaption to new SAP MNSMS
	23-Sep-2002	FK	LL_GETUNITDATA_REQ triggered by LL_UNITDATA_IND
	04-Feb-2003	FK	LL_GETUNITDATA_REQ missing at the beginning

3.7.2 GSMS682: Update MCEF After Successful RP-SMMA

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 and after receiveing an RP-ACK updates EF(SMSS).

Note: Corresponds to SMS490 in circuit switched case.

Preamble: GSMS681

```
     MMI                             SMS                             LLC
      |                               |                               |
 (1)  |                               |    LL_UNITDATA_IND            |
      |                               |       (CP_DATA)               |
      |                               *<==============================*
 (2)  |                               |    LL_UNITDATA_REQ            |
      |                               |       (CP_ACK)                |
      |                               *==============================>*
 (3)  |                               |    SIM_UPDATE_REQ             |
      |                               *==============================>*
 (4)  |                               |    LL_GETUNITDATA_REQ         |
      |                               *==============================>*
MUTE(500)
 (5)  |                               |    LL_UNITREADY_IND           |
      |                               *<==============================*
MUTE(500)
 (6)  |                               |    SIM_UPDATE_CNF             |
      |                               *<==============================*
 (7)  |     MNSMS_REPORT_IND          |                               |
      *<==============================*                               |
MUTE(500)
COMMAND (SMS STATUS PARTITION)
      |                               |                               |
```

Parametrization

Primitive	Parameter	Value
(1) LL_UNITDATA_IND		
	sapi	LL_SAPI_7
	tlli	DEF_SMS_LL_TLLI_1
	reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
	reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
	reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
	reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
	reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
	cipher	LL_CIPHER_OFF
	sdu	
	{	
	component	SMS
	direction	DOWNLINK
	pd	D_CP_DATA
	ti	TI_MO_TO_MS
	cp_user_data_dl	RP_ACK_SMMA
	}	

(2) LL_UNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
ll_qos	SMS_DEFAULT_QOS
radio_prio	LL_RADIO_PRIO_4
cipher	LL_CIPHER_OFF
reserved_unitdata_req1	NOT_USED
seg_pos	NOT_USED
attached_counter	NOT_USED
reserved_unitdata_req4	NOT_USED
reserved_unitdata_req5	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MO
}	

(3) 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

(4) LL_GETUNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

(5) LL_UNITREADY_IND

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

(6) SIM_UPDATE_CNF

datafield	SIM_SMSS
cause	SIM_NO_ERROR

(7) MNSMS_REPORT_IND

state	SMS_STATE_READY

History:	20-Dec-00	LW	Initial
	07-Jun-2002	FK	Adaption to new SAP MNSMS
	23-Sep-2002	FK	LL_GETUNITDATA_REQ triggered by LL_UNITDATA_IND

3.7.3 GSMS683: GSMS Mobile Originated Traffic

Description: The SMS entity is configured to use GPRS for sending mobile originated short messages. The MMI submits a message. Since this is the first message in a transaction, the SMS entity issues a registration state request to GMM.

Variants: <A>...<B>

Preamble: <A> GSMS601A
<B> GSMS601C

```
     MMI                              SMS                               LLC
      |                                |                                  |
(1)   |      MNSMS_SUBMIT_REQ          |                                  |
      *==============================>*                                  |
(2)   |                                |      GMMSMS_REG_STATE_REQ        |
      |                                *================================>*
```

```
(3)   |                              |     SIM_UPDATE_REQ              |
      |                              *================================>*
MUTE(500)
(4)   |                              |     GMMSMS_REG_STATE_CNF        |
      |                              *<================================*
MUTE(2000)
(6)   |                              |     LL_UNITREADY_IND            |
      |                              *<================================*
(7)   |                              |     LL_UNITDATA_REQ             |
      |                              *================================>*
MUTE(2000)
(8)   |                              |     LL_UNITREADY_IND            |
      |                              *<================================*
(9)   |                              |     SIM_UPDATE_CNF              |
      |                              *<================================*
MUTE(500)
(8)   |                              |     LL_UNITDATA_IND             |
      |                              *<================================*
(9)   |                              |     LL_GETUNITDATA_REQ          |
      |                              *================================>*
MUTE(500)
(10)  |                              |     LL_UNITDATA_IND             |
      |                              *<================================*
(11)  |                              |     LL_UNITDATA_REQ             |
      |                              *================================>*
(12)  |     MNSMS_SUBMIT_CNF         |                                 |
      *<=============================*                                 |
(13)  |                              |     LL_GETUNITDATA_REQ          |
      |                              *================================>*
      |                              |                                 |
```

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)GMMSMS_REG_STATE_REQ		
(3)SIM_UPDATE_REQ		
	source	SRC_SMS
	offset	OFFSET_0
	datafield	SIM_SMSS
	length	SIMREC_SMSS_MSG_REF_LEN
	trans_data	SIMREC_SMSS_MSG_REF
(4) GMMSMS_REG_STATE_CNF		
	reg_state	SMS_RS_REGISTERED
	radio_priority_level	SMS_RP_LEVEL_4
(5)LL_UNITREADY_IND		
	sapi	LL_SAPI_7
	tlli	LL_TLLI_INVALID

(6)LL_UNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
ll_qos	SMS_DEFAULT_QOS
radio_prio	LL_RADIO_PRIO_4
cipher	LL_CIPHER_OFF
reserved_unitdata_req1	NOT_USED
seg_pos	NOT_USED
attached_counter	NOT_USED
reserved_unitdata_req4	NOT_USED
reserved_unitdata_req5	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	U_CP_DATA
ti	TI_MO
cp_user_data_ul	RP_DATA_SUBMIT_ABS
}	

(7)LL_UNITREADY_IND

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

(8)SIM_UPDATE_CNF

datafield	SIM_SMSS
cause	SIM_NO_ERROR

(9)LL_UNITDATA_IND

sapi	LL_SAPI_7
tlli	DEF_SMS_LL_TLLI_1
reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
cipher	LL_CIPHER_OFF
sdu	
{	
component	SMS
direction	DOWNLINK
pd	B_CP_ACK
ti	TI_MO_TO_MS
}	

(10) LL_GETUNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

(11) LL_UNITDATA_IND

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
reserved_unitdata_ind1	DEF_RES_UNITDATA_IND1
reserved_unitdata_ind2	DEF_RES_UNITDATA_REQ1
reserved_unitdata_ind3	DEF_RES_UNITDATA_IND3
reserved_unitdata_ind4	DEF_RES_UNITDATA_IND4
reserved_unitdata_ind5	DEF_RES_UNITDATA_IND5
cipher	LL_CIPHER_OFF
sdu	
{	
component	SMS
direction	DOWNLINK
pd	D_CP_DATA
ti	TI_MO_TO_MS
cp_user_data_dl	RP_ACK_DLNK
}	

(12) LL_UNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID
ll_qos	SMS_DEFAULT_QOS
radio_prio	LL_RADIO_PRIO_4
cipher	LL_CIPHER_OFF
reserved_unitdata_req1	NOT_USED
seg_pos	NOT_USED
attached_counter	NOT_USED
reserved_unitdata_req4	NOT_USED
reserved_unitdata_req5	NOT_USED
sdu	
{	
component	SMS
direction	UPLINK
pd	B_CP_ACK
ti	TI_MO
}	

(13) 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

(14) LL_GETUNITDATA_REQ

sapi	LL_SAPI_7
tlli	LL_TLLI_INVALID

History:	18-Aug-2002	LEP	Initial
	29-Jan-2003	FK	Order of primitives corrected