GSM Fax & Data Services

Test Specification

ACICC

Author: Texas Instruments Berlin AG

Alt-Moabit 90a
D-10559 Berlin
Germany

Date: 06 January 2003
Document No.: 8411.417.98.103
File: ACICC.DOC

0 Table of Contents

1 Document Control

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

TI Berlin AG
Alt Moabit 90a
10559 Berlin
Germany

Telephone: +49.30.3983-0
Fax: +49.30.3983-1300
Internet: http://www.ti.com

1.1 Document History

Document Id.	Date	Author	Remarks
8411.417.98.100	5 Oktober 1999	DAK	Initial
8411.417.98.101	18-Mar-2002	HM	Made document work again after LND changes.
8411.417.98.102	25-Nov-2002	SKA	Cause Concept and other changes
8411.417.98.103	06-Jan-2003	FK	TTY/CTM service

1.2 References

[1] Rec. T.4 Standardisation of group 3 facsimile apparatus for document transmission;
(CCITT-T.4, 1984)

[2] ITU-T Recommendation T.30; Series T: Terminal equipments and protocols for telematic services;
Procedures for document facsimile transmission in the general switched
telephone network;
(ITU-T.30, 1996)

[3] ITU-T Recommendation T.31; Terminals for telematic services;
Asynchronous facsimile DCE control - service class 1
(ITU-T.31, 1995)

[4] ITU-T Recommendation T.32; Terminals for telematic services;
Asynchronous facsimile DCE control - service class 2
(ITU-T.32, 1995)

[5] Rec. T.35; Terminal equipment and protocols for telematic services;
Procedures for the allocation of CCITT definde codes for non-standard facilities;
(CCITT-T.35, 1991)

[6] ITU-T Recommendation V.25 ter; Series V: data communication over the telephone network;
Interfaces and voiceband modems; Serial asynchronous automatic dialling and control
(ITU-T V.25 ter, 1997)

[7] Rec. V.42 bis Data compression procedures for data circuit terminating equipment (DCE) using error correction procedures;
(CCITT-V.42 bis, 1990)

[8] Rec. V.110 (Blue book, Vol. VIII, Fascicle VIII.1) Support of data terminal equipments (DTEs) with V-series type interfaces by an integrated services digital network (ISDN);
(CCITT-V.110, 1988)

[9] European digital cellular telecommunications system (Phase 2);
GSM Public Land Mobile Network (PLMN) connection types;
(GSM 3.10, September 1994, version 4.3.1)

[10] European digital cellular telecommunications system (Phase 2);
Technical realisation of facsimile group 3 transparent;
(GSM 3.45, September 1995, version 4.5.0)

[11] Digital cellular telecommunications system (Phase 2);
Mobile radio interface layer 3 specification;
(GSM 4.08, November 1996, version 4.17.0)

[12] European digital cellular telecommunications system (Phase 2);
Rate adaptation on the Mobile Station - Base Station System (MS - BSS) Interface;
(GSM 4.21, May 1995, version 4.6.0)

[13] European digital cellular telecommunications system (Phase 2);
Radio Link Protocol (RLP) for data and telematic services on the Mobile Station - Base Station System (MS - BSS) interface and the Base Station System - Mobile-service Switching Centre (BSS - MSC) interface
(GSM 4.22, September 1994, version 4.3.0)

[14] European digital cellular telecommunications system (Phase 2);
Radio Link Protocol (RLP) for data and telematic services on the Mobile Station - Base Station

System (MS - BSS) interface and the Base Station System - Mobile-service Switching Centre (BSS - MSC) interface
(Amendment prA1 for GSM 4.22, version 4.3.0)
(GSM 4.22, March 1995, version 4.4.0)

[15] European digital cellular telecommunications system (Phase 2);
General on Terminal Adaptation Functions (TAF) for Mobile Stations (MS);
(GSM 7.01, December 1995, version 4.10.0)

[16] European digital cellular telecommunications system (Phase 2);
Terminal Adaptation Functions (TAF) for services using asynchronous bearer capabilities;
(GSM 7.02, September 1994, version 4.5.1)

[17] European digital cellular telecommunications system (Phase 2);
Terminal Adaptation Functions (TAF) for services using synchronous bearer capabilities;
(GSM 7.03, September 1994, version 4.5.1)

[18] Digital cellular telecommunications system (Phase 2);
Use of Data Terminal Equipment - Data Circuit terminating Equipment (DTE - DCE) interface for Short Message Service (SMS) and Cell Broadcast Services (CBS);
(GSM 7.05, November 1996, version 4.8.0)

[19] Digital cellular telecommunications system (Phase 2);
AT command set for GSM Mobile Equipment (ME)
(GSM 7.07, May 1996, version 4.1.0)

[20] Digital cellular telecommunication system (Phase 2);
Mobile Station (MS) conformance specification;
Part 1: Conformance specification
(GSM 11.10-1, November 1996, version 4.17.0)

[21] Digital cellular telecommunications system (Phase 2);
Mobile Station (MS) conformance specification;
Part 2: Protocol Implementation Conformance Statement (PICS)
proforma specification
(GSM 11.10-2, May 1996, version 4.15.0)

[22] Digital cellular telecommunications system (Phase 2);
Mobile Station (MS) conformance specification;
Part 3: Layer 3 (L3) Abstract Test Suite (ATS)
(GSM 11.10-3, November 1996, version 4.17.0)

[23] Proposal for Rate Adaptation implemented on a DSP;
(C. Bianconi, Texas Instruments, January 1998, version 1.0)

[24] MCU-DSP Interfaces for Data Applications;
Specification S844
(C. Bianconi, Texas Instruments, March 1998, version 0.1)

[25] Users Guide
6147.300.96.100; Condat GmbH

[26] Service Access Point RA
8411.100.98.100; Condat GmbH

[27] Service Access Point RLP
8411.101.98.100; Condat GmbH

[28] Service Access Point L2R
8411.102.98.100; Condat GmbH

[29] Service Access Point FAD
8411.103.98.100; Condat GmbH

[30] Service Access Point T30
8411.104.98.100; Condat GmbH

[31] Service Access Point ACI
8411.105.98.100; Condat GmbH

[32] Message Sequence Charts RLP
8411.201.98.100; Condat GmbH

[33] Message Sequence Charts L2R
8411.202.98.100; Condat GmbH

[34] Message Sequence Charts FAD
8411.203.98.100; Condat GmbH

[35] Message Sequence Charts T30
8411.204.98.100; Condat GmbH

[36] Message Sequence Charts ACI
8411.205.98.100; Condat GmbH

[37] Proposal for Fax & Data Integration; March 1998
8411.300.98.100; Condat GmbH

[38] Test Specification RLP
8411.401.98.100; Condat GmbH

[39] Test Specification L2R
8411.402.98.100; Condat GmbH

[40] Test Specification FAD
8411.403.98.100; Condat GmbH

[41] Test Specification T30
8411.404.98.100; Condat GmbH

[42] Test Specification ACI
8411.405.98.100; Condat GmbH

[43] SDL Specification RLP
8411.501.98.100; Condat GmbH

[44] SDL Specification L2R
8411.502.98.100; Condat GmbH

[45] SDL Specification FAD
8411.503.98.100; Condat GmbH

[46] SDL Specification T30
8411.504.98.100; Condat GmbH

[47] SDL Specification ACI
8411.505.98.100; Condat GmbH

[48] Technical Documentation RLP
8411.701.98.100; Condat GmbH

[49] Technical Documentation L2R
8411.702.98.100; Condat GmbH

[50] Technical Documentation FAD
8411.703.98.100; Condat GmbH

[51] Technical Documentation T30
8411.704.98.100; Condat GmbH

[52] Technical Documentation ACI
8411.705.98.100; Condat GmbH

1.3 Abbreviations

ACI	AT Command Interpreter
AGCH	Access Grant Channel
AT	Attention sequence "AT" to indicate valid commands of the ACI
BCCH	Broadcast Control Channel
BCS	Binary Coded Signals
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
CKSN	Ciphering Key Sequence Number
CRC	Cyclic Redundancy Check
DCCH	Dedicated Control Channel
DISC	Disconnect Frame
DL	Data Link Layer
DM	Disconnected Mode Frame
DTX	Discontinuous Transmission
EA	Extension Bit Address Field
EL	Extension Bit Length Field
EMMI	Electrical Man Machine Interface
EOL	End Of Line
F	Final Bit
F&D	Fax and Data Protocol Stack
FACCH	Fast Associated Control Channel
FHO	Forced Handover
GP	Guard Period
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
ITU	International Telecommunication Union
IWF	Interworking Function

Kc	Authentication Key
L	Length Indicator
LAI	Location Area Information
LISR	Low level Interrupt Service Routine
LPD	Link Protocol Discriminator
M	More Data Bit
MCC	Mobile Country Code
MM	Mobility Management
MMI	Man Machine Interface
MNC	Mobile Network Code
MS	Mobile Station
MSG	Message phase in the GSM 3.45 protocol
N(R)	Receive Number
N(S)	Send Number
NCC	National Colour Code
NECI	New Establishment Causes included
OTD	Observed Time Difference
P	Poll Bit
P/F	Poll/Final Bit
PCH	Paging Channel
PCO	Point of Control and Observation
PDU	Protocol Description Unit
PL	Physical Layer
PLMN	Public Land Mobile Network
RACH	Random Access Channel
REJ	Reject Frame
RNR	Receive Not Ready Frame
RR	Radio Resource Management
RR	Receive Ready Frame
RTD	Real Time Difference
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	Slow Dedicated Control Channel
SIM	Subscriber Identity Module
SMS	Short Message Service
SMSCB	Short Message Service Cell Broadcast
SS	Supplementary Services
T.4	CCITT Standardisation for Document coding of Group 3 Facsimile Apparatus
TAP	Test Application Program
TCH	Traffic Channel

TCH/F	Traffic Channel Full Rate
TCH/H	Traffic Channel Half Rate
TDMA	Time Division Multiple Access
TE	Terminal Equipment - e. g. a PC
TMSI	Temporary Mobile Subscriber Identity
UA	Unnumbered Acknowledgement Frame
UI	Unnumbered Information Frame
V(A)	Acknowledgement State Variable
V(R)	Receive State Variable
V(S)	Send State Variable
VPLMN	Visiting Public Land Mobile Network

1.4 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).

2 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 fax and data transmission consists of several entities. Each entity has one ore more service access points, over which the entity provides a service for the upper entity. The entity, which is described in this document, is coloured grey in the following figure :

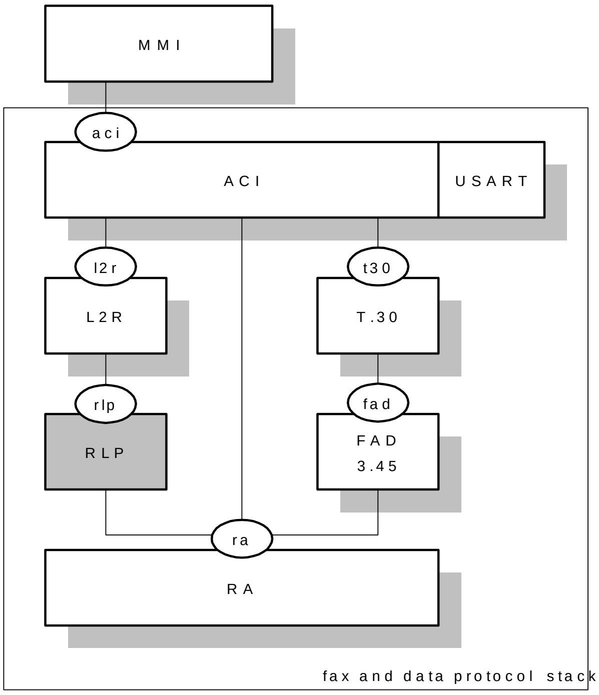

Figure 2-1: Architecture of the fax and data protocol stack

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 fax and data protocol stack are:

2.1 RA – Rate Adaptation

This entity performs an adaptation between an asynchronous or synchronous data stream with several bit rates on to the fixed bit rate used at the TCH. This is performed by the rate adaptation functions RA1' and RA0 described in GSM 04.21.

2.2 RLP – Radio Link Protocol

This entity provides a Layer 2 protocol for asynchronous reliable data transfer as specified in GSM 04.22. It includes error correction, sequence numbers and a mechanism for repeating corrupted and lost messages.

2.3 L2R – Layer 2 Relay Functionality

The L2R provides relay functions in order to adapt the character-oriented data received from the TE via USART to the bit-oriented RLP protocol.

2.4 FAD 03.45 – Fax Adaptation Protocol

The fax adaptation protocol, as specified in GSM 03.45, provides synchronisation with the BCS and MSG modems of the peer entity. It uses byte repetition in conjunction with a voting algorithm to handle corruption on the TCH data stream. The non-transparent fax protocol in accordance with GSM 03.46 is not part of this implementation.

The fax adapter enables T.30 to send BCS at 300 BPS and T.4 MSG in 2400, 4800, 7200 and 9600 BPS.

2.5 T.30 – Fax Protocol Entity

The protocol uses binary coded signals packed in HDLC frames to set up and release a connection in the message phase of the FAX transmission. This entity is specified in the ITU-T.30. The main tasks of this unit are:

- Building the HDLC frames with CRC.
- Performing bit stuffing/de-stuffing.
- Executing a sequence of 5 phases: 1.) set up, 2.) pre-message procedures, 3.) transmission/reception, 4.) post message procedures, 5.) waiting for call release.

2.6 ACI – AT Command Interpreter

The ACI is specified in GSM 07.07. It is responsible for call establishment via the GSM voice protocol stack and terminal adaptation for asynchronous transparent character-oriented data transmission. The ACI is able to receive AT commands and send the replies over the USART driver to a remote PC. This makes it possible to control the voice and data protocol stack from a remote application running on a PC. The ACI also provides a unique interface for an internal MMI in the MS.

2.7 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 F&D. The driver has to be able to perform flow control.

3 Parameters

```
/* --- array declarations ---------------------------------------------------------------
----------------------------------- */
DECLARATION(PHN_NUM0)
DECLARATION(PHN_NUM1)
DECLARATION(EMPTY_PHN_NUM)
DECLARATION (A_ECC_FIELD )
/* --- structure declarations  -----------------------------------------------------------
--------------------------------- */
DECLARATION(CLED_PARTY0)
DECLARATION(CLED_PARTY1)
DECLARATION(CLED_PARTY_SUB_NONE)
DECLARATION(CONNECTED_NUMBER0)
DECLARATION(CONNECTED_NUMBER1)
DECLARATION(CLING_PARTY0)
DECLARATION(CLING_PARTY1)
DECLARATION(CLING_PARTY_SUB_NONE)
DECLARATION(REDIR_PARTY)
DECLARATION(REDIR_PARTY_SUB_NONE)
DECLARATION(CHM_VOICE)
DECLARATION(IMSI_FIELD)
DECLARATION(PREF_PLMN)
DECLARATION (S_BS_DAT_TRA_300)
DECLARATION (S_BS_DAT_TRA_1200)
DECLARATION (S_BS_DAT_TRA_1200_75)
DECLARATION (S_BS_DAT_TRA_2400)
DECLARATION (S_BS_DAT_TRA_2400_V26)
DECLARATION (S_BS_DAT_TRA_4800)
DECLARATION (S_BS_DAT_TRA_9600)
DECLARATION (S_BS_DAT_TRA_9600_V34)
DECLARATION (S_BS_DAT_TRA_14400_V34)
DECLARATION (S_BS_DAT_TRA_1200_TM_NONE)
DECLARATION (S_BS_DAT_TRA_2400_V120)
DECLARATION (S_BS_DAT_TRA_4800_V120)
DECLARATION (S_BS_DAT_TRA_9600_V120)
DECLARATION (S_BS_DAT_TRA_14400_V120)
DECLARATION (S_BS_DAT_TRA_300_TM_NONE)
DECLARATION (S_BS_DAT_NTRA_300)
DECLARATION (S_BS_DAT_NTRA_1200)
DECLARATION (S_BS_DAT_NTRA_1200_75)
DECLARATION (S_BS_DAT_NTRA_2400)
DECLARATION (S_BS_DAT_NTRA_2400_V26)
DECLARATION (S_BS_DAT_NTRA_4800)
DECLARATION (S_BS_DAT_NTRA_9600)
DECLARATION (S_BS_DAT_NTRA_9600_V34)
DECLARATION (S_BS_DAT_NTRA_14400_V34)
DECLARATION (S_BS_DAT_NTRA_1200_TM_NONE)
DECLARATION (S_BS_DAT_NTRA_2400_V120)
DECLARATION (S_BS_DAT_NTRA_4800_V120)
DECLARATION (S_BS_DAT_NTRA_9600_V120)
DECLARATION (S_BS_DAT_NTRA_14400_V120)
DECLARATION (S_BS_DAT_NTRA_300_TM_NONE)
DECLARATION (S_BS_DAT_TRA_AUTO)
DECLARATION (BC_PARA_SPEECH)
DECLARATION (BC_PARA_SPEECH_CTM)
DECLARATION (BC_PARA_SPEECH2)
DECLARATION (BC_PARA_NO_SERVICE)
DECLARATION (STK_PRO_FILE)
DECLARATION (EC_CODES)
DECLARATION (NO_PREF_LANG)
DECLARATION (IMSI_FIELD_DATA)
DECLARATION (SIM_SERV_ADN)
DECLARATION (PREF_PLMN_DATA)
DECLARATION (DATA_ADN_01)
DECLARATION (DATA_ADN_02)

/* Number definitions */
BYTE NOT_SPEC 255
/* Phonebook parameters*/
BYTE MAX_DATA 0xFF
BYTE MAX_ADN_DATA 0x1A
BYTE LDATA_ADN_01 26
BYTE LDATA_ADN_02 15
BYTE LDATA_ACM 3
BYTE NUM_0 0
BYTE NUM_1 1
BYTE NUM_2 2
BYTE NUM_3 3
BYTE NUM_4 4
BYTE NUM_5 5
BYTE NUM_6 6
BYTE NUM_7 7
BYTE NUM_8 8
BYTE NUM_9 9
BYTE NUM_10 10
BYTE NUM_12 12
BYTE NUM_50 50
BYTE NUM_82 82
SHORT NUM_1480 1480
SHORT NUM_1481 1481
SHORT NUM_1482 1482
SHORT NUM_1483 1483
SHORT NUM_9600 9600
SHORT NUM_4800 4800

/* MMI profile parameter, ADN, FDN, BDN supported */
BYTE MMI_PRO_FILE 0xE0

/*---"OK"—(successful operation) */
//      Message:
STRING(M_OK, "OK" )
BYTE LM_OK 2

/*---"NO CARRIER"---*/
//      Message:
STRING(M_NO_CARRIER, "NO CARRIER" )
BYTE LM_NO_CARRIER 10

/*---"NO ANSWER"---*/
//      Message:
STRING(M_NO_ANSWER, "NO ANSWER" )
BYTE LM_NO_ANSWER 9

/*---"BUSY"---*/
//      Message:
STRING(M_BUSY, "BUSY" )
BYTE LM_BUSY 4

/*---"BUSY"---*/
//      Message:
STRING(M_RING, "RING" )
BYTE LM_RING 4

/*---" CONNECT"---*/
//      Message:
STRING(M_CONNECT, "CONNECT" )
BYTE LM_CONNECT 7

/*---"NO DIALTONE"---*/
//      Message:
STRING(M_NO_DIALTONE, "NO DIALTONE" )
BYTE LM_NO_DIALTONE 10

/*---"ERROR"— */
//      Message:
STRING(M_ERROR, "ERROR" )
BYTE LM_ERROR 5

/*---"CME ERROR"—(verbose error result code) */
```

```
//      Message:
STRING(M_CME_ERR_INV_PARAM, "+EXT ERROR: parameter not allowed")
STRING(M_CME_ERR_INV_OPP, "+CME ERROR: operation not allowed" )
BYTE LM_CME_ERR_INV_OPP 33
STRING(M_CME_ERR_UNKN, "+CME ERROR: unknown" )
BYTE LM_CME_ERR_UNKN 19
STRING(M_CME_ERR_OTHER, "+CME ERROR: other error" )
BYTE LM_CME_ERR_OTHER 23
STRING(M_CME_ERR_PIN1_REQ, "+CME ERROR: SIM PIN required" )
BYTE LM_CME_ERR_PIN1_REQ 28
STRING(M_CME_ERR_PIN2_REQ, "+CME ERROR: SIM PIN2 required" )
BYTE LM_CME_ERR_PIN2_REQ 29
STRING(M_CME_ERR_PUK1_REQ, "+CME ERROR: SIM PUK required" )
BYTE LM_CME_ERR_PUK1_REQ 28
STRING(M_CME_ERR_PUK2_REQ, "+CME ERROR: SIM PUK2 required" )
BYTE LM_CME_ERR_PUK2_REQ 29
STRING(M_CME_ERR_NO_SIM, "+CME ERROR: SIM not inserted" )
BYTE LM_CME_ERR_NO_SIM 28
STRING(M_CME_ERR_SIM_FAIL, "+CME ERROR: SIM failure" )
BYTE LM_CME_ERR_SIM_FAIL 23
STRING(M_CME_ERR_SIM_WRONG, "+CME ERROR: SIM wrong" )
BYTE LM_CME_ERR_SIM_WRONG 21
STRING(M_CME_ERR_INV_PWD, "+CME ERROR: incorrect password" )
BYTE LM_CME_ERR_INV_PWD 30
STRING(M_CME_ERR_NOT_FOUND, "+CME ERROR: not found" )
BYTE LM_CME_ERR_NOT_FOUND 21

/*---"EXT ERROR"—(verbose error result code) */
//      Message:
STRING(M_EXT_ERR_PRM_NOT_ALWD, "+EXT ERROR: parameter not allowed" )
BYTE LM_EXT_ERR_PRM_NOT_ALWD 33

/*---"CMEE"—(CMEE_S) */
//      Command:
STRING(C_CMEE_S2, "AT+CMEE=2" )
BYTE LC_CMEE_S 9

/*---"CFUN=1"—(CFUN_S) */
//      Command:
STRING(C_CFUN_S, "AT+CFUN=1" )
BYTE LC_CFUN_S 10

/*---"CPBS="AD""—(CPBS_S) */
//      Command:
STRING(C_CPBS_S, "AT+CPBS=\"AD\"" )
BYTE LC_CPBS_S 13
```

```
/*---"CLIR_COLP"—(CLIR_COLP_S) */
//      Command:
STRING(C_CLIR_COLP_S, "AT+CLIR=2;+COLP=1" )
BYTE LC_CLIR_COLP_S 18

/*---"Dail"—(D) */
//      Command:
STRING(C_D0, "ATD01234567;" )
STRING(C_D1, "ATD+4901234567;" )
BYTE LC_D0 12
BYTE LC_D1 15
STRING(C_PHB_D1, "ATD>\"Meier, Max\"" )
STRING(C_PHB_D2, "ATD>\"Meier, Max\";" )
STRING(C_PHB_D3, "ATD>\"Meier, Max\",G;" )
STRING(C_PHB_D4, "ATD>\"Meier, Max\",I;" )
BYTE LC_PHB_D1 17
BYTE LC_PHB_D2 18
BYTE LC_PHB_D3 19
//      Message:
STRING(M_D0, "+COLP: \"01234567\",129,,128" )
STRING(M_D1, "+COLP: \"+4901234567\",145,,128" )
BYTE LM_D0 26
BYTE LM_D1 29

/*--- "+CHUP" (CHUP_T) ---*/
//      Commqand:
STRING(C_CHUP_T, "AT+CHUP=?" )
BYTE LC_CHUP_T 9
STRING(C_CHUP_Q, "AT+CHUP?" )
BYTE LC_CHUP_Q 8

/*--- "+CHUP" (CHUP_S) ---*/
//      Commqand:
STRING(C_CHUP_S, "AT+CHUP" )
BYTE LC_CHUP_S 8

/*--- "+CMOD=?" (CMOD_T) ---*/
//      Commqand:
STRING(C_CMOD_T, "AT+CMOD=?" )
BYTE LC_CMOD_T 9
//      Message:
STRING(M_CMOD_T, "+CMOD: (0-3)" )
BYTE LM_CMOD_T 12

/*--- "+CMOD?" (CMOD_Q) ----*/
//      Commqand:
STRING(C_CMOD_Q, "AT+CMOD?" )
```

```
BYTE LC_CMOD_Q 8
//      Message:
STRING(M_CMOD_Q0, "+CMOD: 0" )
STRING(M_CMOD_Q1, "+CMOD: 1" )
STRING(M_CMOD_Q2, "+CMOD: 2" )
STRING(M_CMOD_Q3, "+CMOD: 3" )
BYTE LM_CMOD_Q 8

/*--- "+CMOD=*" (CMOD_S) ---*/
//      Commqand:
STRING(C_CMOD_S0, "AT+CMOD=0" )
STRING(C_CMOD_S1, "AT+CMOD=1" )
STRING(C_CMOD_S2, "AT+CMOD=2" )
STRING(C_CMOD_S3, "AT+CMOD=3" )
STRING(C_CMOD_S9, "AT+CMOD=4" )
BYTE LC_CMOD_S 9

/*--- "+CSTA=?" (CSTA_T) ----*/
//      Commqand:
STRING(C_CSTA_T, "AT+CSTA=?" )
BYTE LC_CSTA_T 9
//      Message:
STRING(M_CSTA_T, "+CSTA: (129,145)" )
BYTE LM_CSTA_T 16

/*--- "+CSTA?" (CSTA_Q) ----*/
//      Commqand:
STRING(C_CSTA_Q, "AT+CSTA?" )
BYTE LC_CSTA_Q 8
//      Message:
STRING(M_CSTA_Q0, "+CSTA: 145" )
STRING(M_CSTA_Q1, "+CSTA: 129" )
BYTE LM_CSTA_Q 10

/*--- "+CSTA=*" (CSTA_S) ----*/
//      Commqand:
STRING(C_CSTA_S0, "AT+CSTA=145" )
STRING(C_CSTA_S1, "AT+CSTA=129" )
STRING(C_CSTA_S2, "AT+CSTA=" )
STRING(C_CSTA_S9, "AT+CSTA=255" )
BYTE LC_CSTA_S 11

/*--- "+CBST=?" (CBST_T) ----*/
//      Commqand:
STRING(C_CBST_T, "AT+CBST=?" )
BYTE LC_CBST_T 9
//      Message:
```

```
STRING(M_CBST_T, "+CBST: (0-7,12,14,34,36,38,39,43,65,66,68,70,71,75),(0),(0-3)" )
BYTE LM_CBST_T 61

/*--- "+CBST?" (CBST_Q) ----*/
//      Commqand:
STRING(C_CBST_Q, "AT+CBST?" )
BYTE LC_CBST_Q 9
//      Message:
STRING(M_CBST_Q00, "+CBST: 0,0,1" )     //autobauding only in non-transarent mode
STRING(M_CBST_Q01, "+CBST: 1,0,0" )
STRING(M_CBST_Q02, "+CBST: 1,0,1" )
STRING(M_CBST_Q03, "+CBST: 2,0,0" )
STRING(M_CBST_Q04, "+CBST: 2,0,1" )
STRING(M_CBST_Q05, "+CBST: 3,0,0" )
STRING(M_CBST_Q06, "+CBST: 3,0,1" )
STRING(M_CBST_Q07, "+CBST: 4,0,0" )
STRING(M_CBST_Q08, "+CBST: 4,0,1" )
STRING(M_CBST_Q09, "+CBST: 5,0,0" )
STRING(M_CBST_Q10, "+CBST: 5,0,1" )
STRING(M_CBST_Q11, "+CBST: 6,0,0" )
STRING(M_CBST_Q12, "+CBST: 6,0,1" )
STRING(M_CBST_Q13, "+CBST: 7,0,0" )
STRING(M_CBST_Q14, "+CBST: 7,0,1" )
STRING(M_CBST_Q15, "+CBST: 12,0,0" )
STRING(M_CBST_Q16, "+CBST: 12,0,1" )
STRING(M_CBST_Q17, "+CBST: 14,0,0" )
STRING(M_CBST_Q18, "+CBST: 14,0,1" )
STRING(M_CBST_Q19, "+CBST: 34,0,0" )
STRING(M_CBST_Q20, "+CBST: 34,0,1" )
STRING(M_CBST_Q21, "+CBST: 36,0,0" )
STRING(M_CBST_Q22, "+CBST: 36,0,1" )
STRING(M_CBST_Q23, "+CBST: 38,0,0" )
STRING(M_CBST_Q24, "+CBST: 38,0,1" )
STRING(M_CBST_Q25, "+CBST: 39,0,0" )
STRING(M_CBST_Q26, "+CBST: 39,0,1" )
STRING(M_CBST_Q27, "+CBST: 43,0,0" )
STRING(M_CBST_Q28, "+CBST: 43,0,1" )
STRING(M_CBST_Q29, "+CBST: 65,0,0" )
STRING(M_CBST_Q30, "+CBST: 65,0,1" )
STRING(M_CBST_Q31, "+CBST: 66,0,0" )
STRING(M_CBST_Q32, "+CBST: 66,0,1" )
STRING(M_CBST_Q33, "+CBST: 68,0,0" )
STRING(M_CBST_Q34, "+CBST: 68,0,1" )
STRING(M_CBST_Q35, "+CBST: 70,0,0" )
STRING(M_CBST_Q36, "+CBST: 70,0,1" )
STRING(M_CBST_Q37, "+CBST: 71,0,0" )
STRING(M_CBST_Q38, "+CBST: 71,0,1" )
STRING(M_CBST_Q39, "+CBST: 75,0,0" )
STRING(M_CBST_Q40, "+CBST: 75,0,1" )
BYTE LM_CBST_Q0 12
BYTE LM_CBST_Q1 13              //for M_CSTA_Q > 14

/*--- "+CBST=*" (CBST_S) ----*/
//    Commqand
STRING(C_CBST_S00, "AT+CBST=0,0,1" )    //autobauding only in non-transarent mode
STRING(C_CBST_S01, "AT+CBST=1,0,0" )
STRING(C_CBST_S02, "AT+CBST=1,0,1" )
STRING(C_CBST_S03, "AT+CBST=2,0,0" )
STRING(C_CBST_S04, "AT+CBST=2,0,1" )
STRING(C_CBST_S05, "AT+CBST=3,0,0" )
STRING(C_CBST_S06, "AT+CBST=3,0,1" )
STRING(C_CBST_S07, "AT+CBST=4,0,0" )
STRING(C_CBST_S08, "AT+CBST=4,0,1" )
STRING(C_CBST_S09, "AT+CBST=5,0,0" )
STRING(C_CBST_S10, "AT+CBST=5,0,1" )
STRING(C_CBST_S11, "AT+CBST=6,0,0" )
STRING(C_CBST_S12, "AT+CBST=6,0,1" )
STRING(C_CBST_S13, "AT+CBST=7,0,0" )    //standard settings
STRING(C_CBST_S14, "AT+CBST=7,0,1" )
STRING(C_CBST_S15, "AT+CBST=12,0,0" )
STRING(C_CBST_S16, "AT+CBST=12,0,1" )
STRING(C_CBST_S17, "AT+CBST=14,0,0" )
STRING(C_CBST_S18, "AT+CBST=14,0,1" )
STRING(C_CBST_S19, "AT+CBST=34,0,0" )
STRING(C_CBST_S20, "AT+CBST=34,0,1" )
STRING(C_CBST_S21, "AT+CBST=36,0,0" )
STRING(C_CBST_S22, "AT+CBST=36,0,1" )
STRING(C_CBST_S23, "AT+CBST=38,0,0" )
STRING(C_CBST_S24, "AT+CBST=38,0,1" )
STRING(C_CBST_S25, "AT+CBST=39,0,0" )
STRING(C_CBST_S26, "AT+CBST=39,0,1" )
STRING(C_CBST_S27, "AT+CBST=43,0,0" )
STRING(C_CBST_S28, "AT+CBST=43,0,1" )
STRING(C_CBST_S29, "AT+CBST=65,0,0" )
STRING(C_CBST_S30, "AT+CBST=65,0,1" )
STRING(C_CBST_S31, "AT+CBST=66,0,0" )
STRING(C_CBST_S32, "AT+CBST=66,0,1" )
STRING(C_CBST_S33, "AT+CBST=68,0,0" )
STRING(C_CBST_S34, "AT+CBST=68,0,1" )
STRING(C_CBST_S35, "AT+CBST=70,0,0" )
STRING(C_CBST_S36, "AT+CBST=70,0,1" )
STRING(C_CBST_S37, "AT+CBST=71,0,0" )
STRING(C_CBST_S38, "AT+CBST=71,0,1" )
STRING(C_CBST_S39, "AT+CBST=75,0,0" )
STRING(C_CBST_S40, "AT+CBST=75,0,1" )
STRING(C_CBST_S99, "AT+CBST=0,0,0" )
BYTE LC_CBST_S0 13
BYTE LC_CBST_S1 14

/*--- "+CRLP=?" (CRLP_T) ----*/
//      Commqand:
STRING(C_CRLP_T, "AT+CRLP=?" )
BYTE LC_CRLP_T 10
//      Message:
STRING(M_CRLP_T, "+CRLP: (0-61),(0-61),(39-255),(1-255)" )
BYTE LM_CRLP_T 37

/*--- "+CRLP?" (CRLP_Q) ----*/
//      Commqand:
STRING(C_CRLP_Q, "AT+CRLP?" )
BYTE LC_CRLP_Q 9
//      Message:
STRING(M_CRLP_Q0, "+CRLP: 61,61,48,6" )
STRING(M_CRLP_Q1, "+CRLP: 0,61,48,6" )
STRING(M_CRLP_Q2, "+CRLP: 48,61,48,6" )
STRING(M_CRLP_Q3, "+CRLP: 48,0,48,6" )
STRING(M_CRLP_Q4, "+CRLP: 48,49,48,6" )
STRING(M_CRLP_Q5, "+CRLP: 48,49,39,6" )
STRING(M_CRLP_Q6, "+CRLP: 48,49,250,6" )
STRING(M_CRLP_Q7, "+CRLP: 48,49,250,1" )
STRING(M_CRLP_Q8, "+CRLP: 48,49,250,200" )
BYTE LM_CRLP_Q0 17
BYTE LM_CRLP_Q1 16
BYTE LM_CRLP_Q2 17
BYTE LM_CRLP_Q3 16
BYTE LM_CRLP_Q4 17
BYTE LM_CRLP_Q5 17
BYTE LM_CRLP_Q6 18
BYTE LM_CRLP_Q7 18
BYTE LM_CRLP_Q8 20

/*--- "+CRLP=..." (CRLP_S) ----*/
//      Commqand:
STRING(C_CRLP_S0, "AT+CRLP=0" )
STRING(C_CRLP_S1, "AT+CRLP=48" )
STRING(C_CRLP_S2, "AT+CRLP=48,0" )
STRING(C_CRLP_S3, "AT+CRLP=48,49" )
STRING(C_CRLP_S4, "AT+CRLP=48,49,39" )
STRING(C_CRLP_S5, "AT+CRLP=48,49,250" )
STRING(C_CRLP_S6, "AT+CRLP=48,49,250,1" )
STRING(C_CRLP_S7, "AT+CRLP=48,49,250,200" )
STRING(C_CRLP_S10, "AT+CRLP=64" )
STRING(C_CRLP_S11, "AT+CRLP=48,64" )
STRING(C_CRLP_S12, "AT+CRLP=48,48,30" )
STRING(C_CRLP_S13, "AT+CRLP=48,48,256" )
STRING(C_CRLP_S14, "AT+CRLP=48,48,48,0" )
STRING(C_CRLP_S15, "AT+CRLP=48,48,48,256" )
BYTE LC_CRLP_S0 9
BYTE LC_CRLP_S1 10
BYTE LC_CRLP_S2 12
BYTE LC_CRLP_S3 13
BYTE LC_CRLP_S4 16
BYTE LC_CRLP_S5 17
BYTE LC_CRLP_S6 19
BYTE LC_CRLP_S7 21
BYTE LC_CRLP_S10 10
BYTE LC_CRLP_S11 13
BYTE LC_CRLP_S12 16
BYTE LC_CRLP_S13 17
BYTE LC_CRLP_S14 18
BYTE LC_CRLP_S15 20

/*--- "+CR=?" (CR_T) ----*/
//      Commqand:
STRING(C_CR_T, "AT+CR=?" )
BYTE LC_CR_T 8
//      Message:
STRING(M_CR_T, "+CR: (0,1)" )
BYTE LM_CR_T 10

/*--- "+CR?" (CR_Q) ----*/
//      Commqand:
STRING(C_CR_Q, "AT+CR?" )
BYTE LC_CR_Q 7
//      Message:
STRING(M_CR_Q0, "+CR: 0" )
STRING(M_CR_Q1, "+CR: 1" )
BYTE LM_CR_Q 6

/*--- "+CR=*" (CR_S) ----*/
//      Commqand:
STRING(C_CR_S0, "AT+CR=0" )
STRING(C_CR_S1, "AT+CR=1" )
STRING(C_CR_S9, "AT+CR=2" )
BYTE LC_CR_S 7

/*--- "+CRC=?" (CRC_T) ----*/
//      Commqand:
```

```
STRING( C_CRC_T, "AT+CRC=?" )
BYTE LC_CRC_T 9
//      Message:
STRING( M_CRC_T, "+CRC: (0,1)" )
BYTE LM_CRC_T 11

/*--- "+CRC?" (CRC_Q) ----*/
//      Commqand:
STRING( C_CRC_Q, "AT+CRC?" )
BYTE LC_CRC_Q 8
//      Message:
STRING( M_CRC_Q0, "+CRC: 0" )
STRING( M_CRC_Q1, "+CRC: 1" )
BYTE LM_CRC_Q 7

/*--- "+CRC=*" (CRC_S) ----*/
//      Commqand:
STRING( C_CRC_S0, "AT+CRC=0" )
STRING( C_CRC_S1, "AT+CRC=1" )
STRING( C_CRC_S9, "AT+CRC=2" )
BYTE LC_CRC_S 8

/*--- "+CCUG=?" (CCUG_T) ----*/
//      Commqand:
STRING( C_CCUG_T, "AT+CCUG=?" )
BYTE LC_CCUG_T 10
//      Message:
STRING( M_CCUG_T, "+CCUG: (0,1),(0-10),(0-3)" )
BYTE LM_CCUG_T 25

/*--- "+CCUG?" (CCUG_Q) ----*/
//      Commqand:
STRING( C_CCUG_Q, "AT+CCUG?" )
BYTE LC_CCUG_Q 9
//      Message:
STRING( M_CCUG_Q00, "+CCUG: 0,0,0" )
STRING( M_CCUG_Q01, "+CCUG: 0,1,0" )
STRING( M_CCUG_Q02, "+CCUG: 0,2,0" )
STRING( M_CCUG_Q03, "+CCUG: 0,3,0" )
STRING( M_CCUG_Q04, "+CCUG: 0,4,0" )
STRING( M_CCUG_Q05, "+CCUG: 0,5,0" )
STRING( M_CCUG_Q06, "+CCUG: 0,6,0" )
STRING( M_CCUG_Q07, "+CCUG: 0,7,0" )
STRING( M_CCUG_Q08, "+CCUG: 0,8,0" )
STRING( M_CCUG_Q09, "+CCUG: 0,9,0" )
STRING( M_CCUG_Q10, "+CCUG: 0,10,0" )
STRING( M_CCUG_Q11, "+CCUG: 1,0,0" )
STRING( M_CCUG_Q12, "+CCUG: 1,1,0" )
STRING( M_CCUG_Q13, "+CCUG: 1,2,0" )
STRING( M_CCUG_Q14, "+CCUG: 1,3,0" )
STRING( M_CCUG_Q15, "+CCUG: 1,4,0" )
STRING( M_CCUG_Q16, "+CCUG: 1,5,0" )
STRING( M_CCUG_Q17, "+CCUG: 1,6,0" )
STRING( M_CCUG_Q18, "+CCUG: 1,7,0" )
STRING( M_CCUG_Q19, "+CCUG: 1,8,0" )
STRING( M_CCUG_Q20, "+CCUG: 1,9,0" )
STRING( M_CCUG_Q21, "+CCUG: 1,10,0" )
STRING( M_CCUG_Q22, "+CCUG: 0,0,1" )
STRING( M_CCUG_Q23, "+CCUG: 0,1,1" )
STRING( M_CCUG_Q24, "+CCUG: 0,2,1" )
STRING( M_CCUG_Q25, "+CCUG: 0,3,1" )
STRING( M_CCUG_Q26, "+CCUG: 0,4,1" )
STRING( M_CCUG_Q27, "+CCUG: 0,5,1" )
STRING( M_CCUG_Q28, "+CCUG: 0,6,1" )
STRING( M_CCUG_Q29, "+CCUG: 0,7,1" )
STRING( M_CCUG_Q30, "+CCUG: 0,8,1" )
STRING( M_CCUG_Q31, "+CCUG: 0,9,1" )
STRING( M_CCUG_Q32, "+CCUG: 0,10,1" )
STRING( M_CCUG_Q33, "+CCUG: 1,0,1" )
STRING( M_CCUG_Q34, "+CCUG: 1,1,1" )
STRING( M_CCUG_Q35, "+CCUG: 1,2,1" )
STRING( M_CCUG_Q36, "+CCUG: 1,3,1" )
STRING( M_CCUG_Q37, "+CCUG: 1,4,1" )
STRING( M_CCUG_Q38, "+CCUG: 1,5,1" )
STRING( M_CCUG_Q39, "+CCUG: 1,6,1" )
STRING( M_CCUG_Q40, "+CCUG: 1,7,1" )
STRING( M_CCUG_Q41, "+CCUG: 1,8,1" )
STRING( M_CCUG_Q42, "+CCUG: 1,9,1" )
STRING( M_CCUG_Q43, "+CCUG: 1,10,1" )
STRING( M_CCUG_Q44, "+CCUG: 0,0,2" )
STRING( M_CCUG_Q45, "+CCUG: 0,1,2" )
STRING( M_CCUG_Q46, "+CCUG: 0,2,2" )
STRING( M_CCUG_Q47, "+CCUG: 0,3,2" )
STRING( M_CCUG_Q48, "+CCUG: 0,4,2" )
STRING( M_CCUG_Q49, "+CCUG: 0,5,2" )
STRING( M_CCUG_Q50, "+CCUG: 0,6,2" )
STRING( M_CCUG_Q51, "+CCUG: 0,7,2" )
STRING( M_CCUG_Q52, "+CCUG: 0,8,2" )
STRING( M_CCUG_Q53, "+CCUG: 0,9,2" )
STRING( M_CCUG_Q54, "+CCUG: 0,10,2" )
STRING( M_CCUG_Q55, "+CCUG: 1,0,2" )
STRING( M_CCUG_Q56, "+CCUG: 1,1,2" )
STRING( M_CCUG_Q57, "+CCUG: 1,2,2" )
STRING( M_CCUG_Q58, "+CCUG: 1,3,2" )
STRING(M_CCUG_Q59, "+CCUG: 1,4,2" )
STRING(M_CCUG_Q60, "+CCUG: 1,5,2" )
STRING(M_CCUG_Q61, "+CCUG: 1,6,2" )
STRING(M_CCUG_Q62, "+CCUG: 1,7,2" )
STRING(M_CCUG_Q63, "+CCUG: 1,8,2" )
STRING(M_CCUG_Q64, "+CCUG: 1,9,2" )
STRING(M_CCUG_Q65, "+CCUG: 1,10,2" )
STRING(M_CCUG_Q66, "+CCUG: 0,0,3" )
STRING(M_CCUG_Q67, "+CCUG: 0,1,3" )
STRING(M_CCUG_Q68, "+CCUG: 0,2,3" )
STRING(M_CCUG_Q69, "+CCUG: 0,3,3" )
STRING(M_CCUG_Q70, "+CCUG: 0,4,3" )
STRING(M_CCUG_Q71, "+CCUG: 0,5,3" )
STRING(M_CCUG_Q72, "+CCUG: 0,6,3" )
STRING(M_CCUG_Q73, "+CCUG: 0,7,3" )
STRING(M_CCUG_Q74, "+CCUG: 0,8,3" )
STRING(M_CCUG_Q75, "+CCUG: 0,9,3" )
STRING(M_CCUG_Q76, "+CCUG: 0,10,3" )
STRING(M_CCUG_Q77, "+CCUG: 1,0,3" )
STRING(M_CCUG_Q78, "+CCUG: 1,1,3" )
STRING(M_CCUG_Q79, "+CCUG: 1,2,3" )
STRING(M_CCUG_Q80, "+CCUG: 1,3,3" )
STRING(M_CCUG_Q81, "+CCUG: 1,4,3" )
STRING(M_CCUG_Q82, "+CCUG: 1,5,3" )
STRING(M_CCUG_Q83, "+CCUG: 1,6,3" )
STRING(M_CCUG_Q84, "+CCUG: 1,7,3" )
STRING(M_CCUG_Q85, "+CCUG: 1,8,3" )
STRING(M_CCUG_Q86, "+CCUG: 1,9,3" )
STRING(M_CCUG_Q87, "+CCUG: 1,10,3" )
BYTE LM_CCUG_Q0 12
BYTE LM_CCUG_Q1 13 //only for S10, S21, 32, 43, 54, 65, 76, 87

/*--- "+CCUG=*" (CCUG_S) ----*/
//      Commqand:
STRING(C_CCUG_S00, "AT+CCUG=0,0,0" )
STRING(C_CCUG_S01, "AT+CCUG=0,1,0" )
STRING(C_CCUG_S02, "AT+CCUG=0,2,0" )
STRING(C_CCUG_S03, "AT+CCUG=0,3,0" )
STRING(C_CCUG_S04, "AT+CCUG=0,4,0" )
STRING(C_CCUG_S05, "AT+CCUG=0,5,0" )
STRING(C_CCUG_S06, "AT+CCUG=0,6,0" )
STRING(C_CCUG_S07, "AT+CCUG=0,7,0" )
STRING(C_CCUG_S08, "AT+CCUG=0,8,0" )
STRING(C_CCUG_S09, "AT+CCUG=0,9,0" )
STRING(C_CCUG_S10, "AT+CCUG=0,10,0" )
STRING(C_CCUG_S11, "AT+CCUG=1,0,0" )
STRING(C_CCUG_S12, "AT+CCUG=1,1,0" )
STRING( C_CCUG_S13, "AT+CCUG=1,2,0" )
STRING( C_CCUG_S14, "AT+CCUG=1,3,0" )
STRING( C_CCUG_S15, "AT+CCUG=1,4,0" )
STRING( C_CCUG_S16, "AT+CCUG=1,5,0" )
STRING( C_CCUG_S17, "AT+CCUG=1,6,0" )
STRING( C_CCUG_S18, "AT+CCUG=1,7,0" )
STRING( C_CCUG_S19, "AT+CCUG=1,8,0" )
STRING( C_CCUG_S20, "AT+CCUG=1,9,0" )
STRING( C_CCUG_S21, "AT+CCUG=1,10,0" )
STRING( C_CCUG_S22, "AT+CCUG=0,0,1" )
STRING( C_CCUG_S23, "AT+CCUG=0,1,1" )
STRING( C_CCUG_S24, "AT+CCUG=0,2,1" )
STRING( C_CCUG_S25, "AT+CCUG=0,3,1" )
STRING( C_CCUG_S26, "AT+CCUG=0,4,1" )
STRING( C_CCUG_S27, "AT+CCUG=0,5,1" )
STRING( C_CCUG_S28, "AT+CCUG=0,6,1" )
STRING( C_CCUG_S29, "AT+CCUG=0,7,1" )
STRING( C_CCUG_S30, "AT+CCUG=0,8,1" )
STRING( C_CCUG_S31, "AT+CCUG=0,9,1" )
STRING( C_CCUG_S32, "AT+CCUG=0,10,1" )
STRING( C_CCUG_S33, "AT+CCUG=1,0,1" )
STRING( C_CCUG_S34, "AT+CCUG=1,1,1" )
STRING( C_CCUG_S35, "AT+CCUG=1,2,1" )
STRING( C_CCUG_S36, "AT+CCUG=1,3,1" )
STRING( C_CCUG_S37, "AT+CCUG=1,4,1" )
STRING( C_CCUG_S38, "AT+CCUG=1,5,1" )
STRING( C_CCUG_S39, "AT+CCUG=1,6,1" )
STRING( C_CCUG_S40, "AT+CCUG=1,7,1" )
STRING( C_CCUG_S41, "AT+CCUG=1,8,1" )
STRING( C_CCUG_S42, "AT+CCUG=1,9,1" )
STRING( C_CCUG_S43, "AT+CCUG=1,10,1" )
STRING( C_CCUG_S44, "AT+CCUG=0,0,2" )
STRING( C_CCUG_S45, "AT+CCUG=0,1,2" )
STRING( C_CCUG_S46, "AT+CCUG=0,2,2" )
STRING( C_CCUG_S47, "AT+CCUG=0,3,2" )
STRING( C_CCUG_S48, "AT+CCUG=0,4,2" )
STRING( C_CCUG_S49, "AT+CCUG=0,5,2" )
STRING( C_CCUG_S50, "AT+CCUG=0,6,2" )
STRING( C_CCUG_S51, "AT+CCUG=0,7,2" )
STRING( C_CCUG_S52, "AT+CCUG=0,8,2" )
STRING( C_CCUG_S53, "AT+CCUG=0,9,2" )
STRING( C_CCUG_S54, "AT+CCUG=0,10,2" )
STRING( C_CCUG_S55, "AT+CCUG=1,0,2" )
STRING( C_CCUG_S56, "AT+CCUG=1,1,2" )
STRING( C_CCUG_S57, "AT+CCUG=1,2,2" )
STRING( C_CCUG_S58, "AT+CCUG=1,3,2" )
STRING( C_CCUG_S59, "AT+CCUG=1,4,2" )
STRING(C_CCUG_S60, "AT+CCUG=1,5,2" )
STRING(C_CCUG_S61, "AT+CCUG=1,6,2" )
STRING(C_CCUG_S62, "AT+CCUG=1,7,2" )
STRING(C_CCUG_S63, "AT+CCUG=1,8,2" )
STRING(C_CCUG_S64, "AT+CCUG=1,9,2" )
STRING(C_CCUG_S65, "AT+CCUG=1,10,2" )
STRING(C_CCUG_S66, "AT+CCUG=0,0,3" )
STRING(C_CCUG_S67, "AT+CCUG=0,1,3" )
STRING(C_CCUG_S68, "AT+CCUG=0,2,3" )
STRING(C_CCUG_S69, "AT+CCUG=0,3,3" )
STRING(C_CCUG_S70, "AT+CCUG=0,4,3" )
STRING(C_CCUG_S71, "AT+CCUG=0,5,3" )
STRING(C_CCUG_S72, "AT+CCUG=0,6,3" )
STRING(C_CCUG_S73, "AT+CCUG=0,7,3" )
STRING(C_CCUG_S74, "AT+CCUG=0,8,3" )
STRING(C_CCUG_S75, "AT+CCUG=0,9,3" )
STRING(C_CCUG_S76, "AT+CCUG=0,10,3" )
STRING(C_CCUG_S77, "AT+CCUG=1,0,3" )
STRING(C_CCUG_S78, "AT+CCUG=1,1,3" )
STRING(C_CCUG_S79, "AT+CCUG=1,2,3" )
STRING(C_CCUG_S80, "AT+CCUG=1,3,3" )
STRING(C_CCUG_S81, "AT+CCUG=1,4,3" )
STRING(C_CCUG_S82, "AT+CCUG=1,5,3" )
STRING(C_CCUG_S83, "AT+CCUG=1,6,3" )
STRING(C_CCUG_S84, "AT+CCUG=1,7,3" )
STRING(C_CCUG_S85, "AT+CCUG=1,8,3" )
STRING(C_CCUG_S86, "AT+CCUG=1,9,3" )
STRING(C_CCUG_S87, "AT+CCUG=1,10,3" )
STRING(C_CCUG_S97, "AT+CCUG=2,0,0" )
STRING(C_CCUG_S98, "AT+CCUG=0,11,0" )
STRING(C_CCUG_S99, "AT+CCUG=0,0,4" )
BYTE LC_CCUG_S0 14
BYTE LC_CCUG_S1 15 //only for S10, S21, S32, S43, S54, S65, S76, S87, S98

/*--- "+CHLD=?" (CHLD_T) ----*/
//      Commqand:
STRING(C_CHLD_T, "AT+CHLD=?" )
BYTE LC_CHLD_T 10
//      Message:
STRING(M_CHLD_T, "+CHLD: (0,1,1x,2,2x,3,4)" )
BYTE LM_CHLD_T 24

/*--- "+CHLD?" (CHLD_Q) ----*/
//      Commqand:
STRING(C_CHLD_Q, "AT+CHLD?" )
BYTE LC_CHLD_Q 8
```

```
/*--- "+CHLD=" (CHLD_S) ----*/
//      Commqand:
STRING(C_CHLD_S0, "AT+CHLD=" )
STRING(C_CHLD_S1, "AT+CHLD=1" )
BYTE LC_CHLD_S0 9
BYTE LC_CHLD_S1 10

/*--- "+CTFR=?"(CTFR_T) ,"+CTFR?"(CTFR_Q) ----*/
//      Commqand:
STRING(C_CTFR_T, "AT+CTFR=?" )
BYTE LC_CTFR_T 10
STRING(C_CTFR_Q, "AT+CTFR?" )
BYTE LC_CTFR_Q 9

/*--- "+CTFR=..."(CTFR_S) ----*/
//      Commqand:
STRING(C_CTFR_S, "AT+CTFR=\"01234567\"" )
BYTE LC_CTFR_S 23

/*--- "+CAOC=?" (CAOC_T) ----*/
//      Commqand:
STRING(C_CAOC_T, "AT+CAOC=?" )
BYTE LC_CAOC_T 10
//      Message:
STRING(M_CAOC_T, "+CAOC: (0-2)" )
BYTE LM_CAOC_T 12

/*--- "+CAOC?" (CAOC_Q) ----*/
//      Commqand:
STRING(C_CAOC_Q, "AT+CAOC?" )
BYTE LC_CAOC_Q 9
//      Message:
STRING(M_CAOC_Q0, "+CAOC: 0" )
STRING(M_CAOC_Q1, "+CAOC: 1" )
STRING(M_CAOC_Q2, "+CAOC: 2" )
STRING(M_CCM, "+CAOC: \"000000\"" )
BYTE LM_CCM 15
BYTE LM_CAOC_Q 8

/*--- "+CAOC=*" (CAOC_S) ----*/
//      Commqand:
STRING(C_CAOC_S0, "AT+CAOC=0" )
STRING(C_CAOC_S1, "AT+CAOC=1" )
STRING(C_CAOC_S2, "AT+CAOC=2" )
STRING(C_CAOC_S9, "AT+CAOC=4" )
BYTE LC_CAOC_S 10
```

```
/*--- "+CACM=?" (CACM_T) ----*/
//      Commqand:
STRING(C_CACM_T, "AT+CACM=?" )
BYTE LC_CACM_T 10

/*--- "+CACM?" (CACM_Q) ----*/
//      Commqand:
STRING(C_CACM_Q, "AT+CACM?" )
BYTE LC_CACM_Q 9
//      Message:
STRING(M_CACM_Q, "+CACM: \"000000\"" )
BYTE LM_CACM_Q 15

/*--- "+CACM=\"12345\"" (CACM_T) ----*/
//      Commqand:
STRING(C_CACM_S, "AT+CACM=\"12345\"" )
BYTE LC_CACM_S 16

/*--- "+CAMM=?" (CAMM_T) ----*/
//      Commqand:
STRING(C_CAMM_T, "AT+CAMM=?" )
BYTE LC_CAMM_T 10

/*--- "+CAMM?" (CAMM_Q) ----*/
//      Commqand:
STRING(C_CAMM_Q, "AT+CAMM?" )
BYTE LC_CAMM_Q 9
//      Message:
STRING(M_CAMM_Q, "+CAMM: \"000000\"" )
BYTE LM_CAMM_Q 15

/*--- "+CLCC=?" (CLCC_T) ----*/
//      Commqand:
STRING(C_CLCC_T, "AT+CLCC=?" )
BYTE LC_CLCC_T 10

/*--- "+CLCC?" (CLCC_Q) ----*/
//      Commqand:
STRING(C_CLCC_Q, "AT+CLCC" )
BYTE LC_CLCC_Q 8
//      Message:
STRING(M_CLCC_Q, "+CLCC:" )
BYTE LM_CLCC_Q 9

/*--- "T" (T), "P" (P) ----*/
//      Commqand:
STRING(C_T, "ATT" )
```

```
BYTE LC_T 4
STRING(C_P, "ATP" )
BYTE LC_P 4

/*--- "S0" (S0_T) ----*/
//      Commqand:
STRING(C_S0_T, "ATS0=?" )
BYTE LC_S0_T 7
//      Message:
STRING(M_S0_T, "S0:(0-255)" )
BYTE LM_S0_T 10

/*--- "S0" (S0_Q) ----*/
//      Commqand:
STRING(C_S0_Q, "ATS0?" )
BYTE LC_S0_Q 6
//      Message:
STRING(M_S0_Q0, "000" )
STRING(M_S0_Q1, "001" )
STRING(M_S0_Q2, "002" )
BYTE LM_S0_Q 3

/*--- "S0" (S0_S) ----*/
//      Commqand:
STRING(C_S0_S, "ATS0=2" )
BYTE LC_S0_S 7

/*--- "S6" (S6_T) ----*/
//      Commqand:
STRING(C_S6_T, "ATS6=?" )
BYTE LC_S6_T 7
//      Message:
STRING(M_S6_T, "S6:(2-10)" )
BYTE LM_S6_T 9

/*--- "S6" (S6_Q) ----*/
//      Commqand:
STRING(C_S6_Q, "ATS6?" )
BYTE LC_S6_Q 6
//      Message:
STRING(M_S6_Q0, "002")
STRING(M_S6_Q1, "003")
STRING(M_S6_Q2, "004")
STRING(M_S6_Q3, "005")
STRING(M_S6_Q4, "006")
STRING(M_S6_Q5, "007")
STRING(M_S6_Q6, "008")
STRING(M_S6_Q7, "009")
STRING(M_S6_Q8, "010")
BYTE LM_S6_Q0 3
BYTE LM_S6_Q1 8

/*--- "S6" (S6_S) ----*/
//      Commqand:
STRING(C_S6_S0, "ATS6=2" )
STRING(C_S6_S1, "ATS6=3" )
STRING(C_S6_S2, "ATS6=4" )
STRING(C_S6_S3, "ATS6=5" )
STRING(C_S6_S4, "ATS6=6" )
STRING(C_S6_S5, "ATS6=7" )
STRING(C_S6_S6, "ATS6=8" )
STRING(C_S6_S7, "ATS6=9" )
STRING(C_S6_S8, "ATS6=10" )
STRING(C_S6_S9, "ATS6=11" )
BYTE LC_S6_S0 7
BYTE LC_S6_S1 8

/*--- "S7" (S7_T) ----*/
//      Commqand:
STRING(C_S7_T, "ATS7=?" )
BYTE LC_S7_T 7
//      Message:
STRING(M_S7_T, "S7:(1-255)" )
BYTE LM_S7_T 10

/*--- "S7" (S7_Q) ----*/
//      Commqand:
STRING(C_S7_Q, "ATS7?" )
BYTE LC_S7_Q 5
//      Message:
STRING(M_S7_Q0, "001" )
STRING(M_S7_Q1, "050" )
STRING(M_S7_Q2, "100" )
STRING(M_S7_Q3, "200" )
STRING(M_S7_Q4, "255" )
BYTE LM_S7_Q 3

/*--- "S7" (S7_S) ----*/
//      Commqand:
STRING(C_S7_S0, "ATS7=1" )
STRING(C_S7_S1, "ATS7=50" )
STRING(C_S7_S2, "ATS7=100" )
STRING(C_S7_S3, "ATS7=200" )
STRING(C_S7_S4, "ATS7=255" )
STRING(C_S7_S8, "ATS7=0" )
STRING(C_S7_S9, "ATS7=256" )
BYTE LC_S7_S0 7
BYTE LC_S7_S1 8
BYTE LC_S7_S2 9

/*--- "S8" (S8_T) ----*/
//      Commqand:
STRING(C_S8_T, "ATS8=?" )
BYTE LC_S8_T 7
//      Message:
STRING(M_S8_T, "S8:(0-255)" )
BYTE LM_S8_T 10

/*--- "S8" (S8_Q) ----*/
//      Commqand:
STRING(C_S8_Q, "ATS8?" )
BYTE LC_S8_Q 6
//      Message:
STRING(M_S8_Q0, "002" )
STRING(M_S8_Q1, "050" )
STRING(M_S8_Q2, "100" )
STRING(M_S8_Q3, "200" )
STRING(M_S8_Q4, "255" )
STRING(M_S8_Q5, "000" )
BYTE LM_S8_Q 3

/*--- "S8" (S8_S) ----*/
//      Commqand:
STRING(C_S8_S0, "ATS8=0" )
STRING(C_S8_S1, "ATS8=50" )
STRING(C_S8_S2, "ATS8=100" )
STRING(C_S8_S3, "ATS8=200" )
STRING(C_S8_S4, "ATS8=255" )
STRING(C_S8_S9, "ATS8=256" )
BYTE LC_S8_S0 7
BYTE LC_S8_S1 8
BYTE LC_S8_S2 9

/*--- "S10" (S10_T) ----*/
//      Commqand:
STRING(C_S10_T, "ATS10=?" )
BYTE LC_S10_T 8
//      Message:
STRING(M_S10_T, "S10:(1-254)" )
BYTE LM_S10_T 11
```

```
/*--- "S10" (S10_Q) ----*/
//      Commqand:
STRING(C_S10_Q, "ATS10?" )
BYTE LC_S10_Q 7
//      Message:
STRING(M_S10_Q0, "001" )
STRING(M_S10_Q1, "050" )
STRING(M_S10_Q2, "100" )
STRING(M_S10_Q3, "200" )
STRING(M_S10_Q4, "254" )
BYTE LM_S10_Q 3

/*--- "S10" (S10_S) ----*/
//      Commqand:
STRING(C_S10_S0, "ATS10=1" )
STRING(C_S10_S1, "ATS10=50" )
STRING(C_S10_S2, "ATS10=100" )
STRING(C_S10_S3, "ATS10=200" )
STRING(C_S10_S4, "ATS10=254" )
STRING(C_S10_S8, "ATS10=0" )
STRING(C_S10_S9, "ATS10=256" )
BYTE LC_S10_S0 8
BYTE LC_S10_S1 9
BYTE LC_S10_S2 10

/*--- "+CEER" (CEER_T) ----*/
//      Commqand:
STRING(C_CEER_T, "AT+CEER=?" )
BYTE LC_CEER_T 9

 /*--- "+CEER" (CEER_Q) ----*/
//      Commqand:
STRING(C_CEER_Q, "AT+CEER?" )
BYTE LC_CEER_Q 8

/*--- "+CEER" (CEER_S) ----*/
//      Commqand:
STRING(C_CEER_S, "AT+CEER" )
BYTE LC_CEER_S 7
//      Message:
STRING(M_CEER_NO_ERR, "+CEER: 0,1,1,255,no error" )
STRING(M_CEER_TIMER_RECOVERY, "+CEER: 0,1,5,102,recovery on time expiry" )
STRING(M_CEER_INCOMP_DEST, "+CEER: 0,1,5,88,incompatible destination" )
STRING(M_CEER_AUTH_REJ, "+CEER: 1,1,4,136" )
STRING(M_CEER_UNSPEC, "+CEER: 0,0,4,111,protocol error" )
STRING(M_CEER_INT_NOT_PRES, "+CEER: 0,1,4,255,no error" )
STRING(M_CEER_CC_NOT_PRES, "+CEER: 0,1,5,255,no error" )
STRING(M_CEER_USR_BUSY, "+CEER: 0,0,5,17,user busy" )
STRING(M_CEER_ALRT_NO_ANSW,"+CEER: 0,0,5,19,user alerting, no answer")
STRING(M_CEER_UNASSIGNED,"+CEER: 0,0,5,1,unassigned number")
STRING(M_CEER_NO_ROUTE,"+CEER: 0,0,5,3,no route to destination")
STRING(M_CEER_NO_USR_RESP,"+CEER: 0,0,5,18,no user responding")
STRING(M_CEER_DEST_OOO,"+CEER: 0,0,5,27,destination out of order")
STRING(M_CEER_INV_FORMAT,"+CEER: 0,0,5,28,invalid number format")
STRING(M_CEER_CALL_CLEAR,"+CEER: 0,0,5,16,normal call clearing")
BYTE LM_CEER_NO_ERR 25
BYTE LM_CEER_TIMER_RECOVERY 40
BYTE LM_CEER_INCOMP_DEST 40
BYTE LM_CEER_AUTH_REJ 16
BYTE LM_CEER_UNSPEC 31
BYTE LM_CEER_INT_NOT_PRES 25
BYTE LM_CEER_NOT_PRES 25
BYTE LM_CEER_USR_BUSY 25
BYTE LM_CEER_ALRT_NO_ANSW 40
BYTE LM_CEER_UNASSIGNED 32
BYTE LM_CEER_NO_ROUTE 38
BYTE LM_CEER_NO_USR_RESP 34
BYTE LM_CEER_DEST_OOO 40
BYTE LM_CEER_INV_FORMAT 37
BYTE LM_CEER_CALL_CLEAR 36

/*---"+CRING"-(+CRING)---*/
//      Message:
STRING(M_CRING_VOICE, "+CRING: VOICE" )
STRING(M_CRING_ASYNC, "+CRING: ASYNC" )
BYTE LM_CRING_VOICE 13
BYTE LM_CRING_ASYNC 13

/*---"+CLIP=1;+CRC=1"(C_CLIP_CRC_S)---*/
//      Command:
STRING(C_CLIP_CRC_S, "AT+CLIP=1;+CRC=1" )
BYTE LC_CLIP_CRC_S 16
//      Message:
STRING(M_CLIP_Q01, "+CLIP: \"01234567\",129,,,,0" )
STRING(M_CLIP_Q02, "+CLIP: \"+01234567\",145,,,,0"   )
BYTE LM_CLIP_Q01 26
BYTE LM_CLIP_Q02 27

/*---"A"-(A)---*/
//      Command:
STRING(C_A, "ATA" )
BYTE LC_A 4
STRING(C_A_T, "ATA=?" )
BYTE LC_A_T 6
```

```
STRING(C_A_Q, "ATA?" )
BYTE LC_A_Q 5

/*---"H"-(H)---*/
//      Command:
STRING(C_H, "ATH" )
BYTE LC_H 4
STRING(C_H_T, "ATH=?" )
BYTE LC_H_T 6
STRING(C_H_Q, "ATH?" )
BYTE LC_H_Q 5

/*--- %CTTY ---*/
//      Command:
STRING(C_TTY_DIS_NOREQ, "AT%CTTY=0,0")
STRING(C_TTY_DIS_REQ, "AT%CTTY=0,1")
STRING(C_TTY_EN_NOREQ, "AT%CTTY=1,0")
STRING(C_TTY_EN_REQ, "AT%CTTY=1,1")
BYTE LC_TTY 11
//      QUERY:
STRING(C_TTY_QUERY, "AT%CTTY?")
BYTE LC_TTY_QUERY 8
//      TEST:
STRING(C_TTY_TEST, "AT%CTTY=?")
BYTE LC_TTY_TEST 9
//      MESSAGE:
STRING(M_TTY_DIS_NOREQ_UNKNOWN, "%CTTY: 0,0,2")
STRING(M_TTY_DIS_REQ_UNKNOWN, "%CTTY: 0,1,2")
STRING(M_TTY_EN_NOREQ_UNKNOWN, "%CTTY: 1,0,2")
STRING(M_TTY_EN_REQ_UNKNOWN, "%CTTY: 1,1,2")
STRING(M_TTY_DIS_REQ_OFF, "%CTTY: 0,1,0")
STRING(M_TTY_DIS_NOREQ_ON, "%CTTY: 0,0,1")
STRING(M_TTY_DIS_REQ_ON, "%CTTY: 0,1,1")
STRING(M_TTY_EN_REQ_OFF, "%CTTY: 1,1,0")
STRING(M_TTY_EN_NOREQ_ON, "%CTTY: 1,0,1")
STRING(M_TTY_EN_REQ_ON, "%CTTY: 1,1,1")
BYTE LM_TTY 12
STRING(M_TTY_TEST, "%CTTY: (0,1),(0,1)")
BYTE LM_TTY_TEST 18
STRING(M_TYI_NOREQ, "%CTYI: 0")
STRING(M_TYI_REQ, "%CTYI: 1")
STRING(M_TYI_NOGRANT, "%CTYI: 2")
STRING(M_TYI_GRANT, "%CTYI: 3")
BYTE LM_TYI 8
STRING(C_D_TTY_OFF, "ATD#55#01234567;" )
STRING(C_D_TTY_ON, "ATD*55#01234567;" )
BYTE LC_D_TTY 16
```

```
/*--------------------------------------  fields  ------------------------------
-------------------- */
/* --- bearer capability info, speech only ----*/

BEGIN_PSTRUCT ("bcpara", BC_PARA_SPEECH)
       SET_COMP ("rate",          UR_NOT_PRES)
       SET_COMP ("bearer_serv",   BEARER_SERV_SPEECH)
       SET_COMP ("conn_elem",     CONN_ELEM_NOT_PRES)
       SHOW_COMP ("stop_bits")
       SHOW_COMP ("data_bits")
       SHOW_COMP ("parity")
       SHOW_COMP ("flow_control")
/*     SET_COMP ("stop_bits",     STOP_1_BIT)
       SET_COMP ("data_bits",     DATA_8_BIT)
       SET_COMP ("parity",        PARITY_NONE)
       SET_COMP ("flow_control",  NO_FLOW_CONTROL)*/
       SET_COMP ("modem_type",    MT_NONE)
ENDSTRUCT

BEGIN_PSTRUCT ("bcpara", BC_PARA_SPEECH_CTM)
       SET_COMP ("rate",          UR_NOT_PRES)
       SET_COMP ("bearer_serv",   BEARER_SERV_SPEECH_CTM)
       SET_COMP ("conn_elem",     CONN_ELEM_NOT_PRES)
       SHOW_COMP ("stop_bits")
       SHOW_COMP ("data_bits")
       SHOW_COMP ("parity")
       SHOW_COMP ("flow_control")
/*     SET_COMP ("stop_bits",     STOP_1_BIT)
       SET_COMP ("data_bits",     DATA_8_BIT)
       SET_COMP ("parity",        PARITY_NONE)
       SET_COMP ("flow_control",  NO_FLOW_CONTROL)*/
       SET_COMP ("modem_type",    MT_NONE)
ENDSTRUCT

/* --- bearer capability info, speech only ----*/

BEGIN_PSTRUCT ("bcpara", BC_PARA_SPEECH2)
       SET_COMP ("rate",    UR_NOT_PRES)
       SET_COMP ("bearer_serv",     BEARER_SERV_SPEECH)
       SET_COMP ("conn_elem",       CONN_ELEM_NOT_PRES)
       SET_COMP ("stop_bits",       STOP_1_BIT)
       SET_COMP ("data_bits",       DATA_8_BIT)
       SET_COMP ("parity",  PARITY_NONE)
       SET_COMP ("flow_control",    NO_FLOW_CONTROL)
       SET_COMP ("modem_type",      MT_NONE)
ENDSTRUCT
```

```
/* --- bearer capability info, no service ----*/

BEGIN_PSTRUCT ("bcpara2", BC_PARA_NO_SERVICE)
        SET_COMP ("rate",       UR_NOT_PRES)
        SET_COMP ("bearer_serv",        BEARER_SERV_NOT_PRES)
        SET_COMP ("conn_elem",          CONN_ELEM_NOT_PRES)
        SET_COMP ("stop_bits",          STOP_1_BIT)
        SET_COMP ("data_bits",          DATA_8_BIT)
        SET_COMP ("parity",     PARITY_NONE)
        SET_COMP ("flow_control",       NO_FLOW_CONTROL)
        SET_COMP ("modem_type",         MT_NONE)
ENDSTRUCT


/* --- SIMTOOLKIT profile --- */
BEGINARRAY (STK_PRO_FILE, 10)
        0x0C,
        0x00,
        0x00,
        0x00,
        0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00
ENDARRAY

/* --- emergency call codes --- */
BEGINARRAY (EC_CODES, 16)
        0x11, 0xF2, 0xFF,
        0x99, 0xF9, 0xFF,
        0xFF,0xFF,0xFF,0xFF,0xFF,
        0xFF,0xFF,0xFF,0xFF
ENDARRAY

/* --- preferd language ---*/
BEGINARRAY (NO_PREF_LANG, 5)
        0xFF,0xFF,0xFF,0xFF,0xFF
ENDARRAY

/* --- international mobil subscriber identity --- */
BEGINARRAY (IMSI_FIELD_DATA, 9)
        0x02,0x21,0x30,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF
ENDARRAY

/* --- SIM service table - only ADN supported --- */
BEGINARRAY (SIM_SERV_ADN, 10)
        0xCF,  //     1100 1111
        0x3C,  //     0011 1100
        0x3C,  //     0011 1100
        0x03,  //     0000 0011
        0xF3,  //     1111 0011
        0x00,  //     0000 0000
        0x00,  //     0000 0000
        0x00,  //     0000 0000
        0x00,  //     0000 0000
        0x00   //     0000 0000
ENDARRAY

/* --- prefered PLMN --- */
BEGINARRAY (PREF_PLMN_DATA, 96)
        0x21, 0xF3, 0x89,     0x89, 0xF7, 0x21 ,   0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,    0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,    0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,    0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,    0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,    0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,    0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,
        0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF,    0xFF, 0xFF, 0xFF ,    0xFF, 0xFF, 0xFF
ENDARRAY

/* --- ADN record data --- */
BEGINARRAY(DATA_ADN_01, 24)
        0x4D, 0x65, 0x69, 0x65, 0x72, 0x2C, 0x20, 0x4D, 0x61, 0x78,          // "Meier, Max"
        0x08,
        0xC9,                                                        // 1100 1001
        0x00, 0x94, 0x21, 0x93, 0x78, 0x56, 0x34, 0xFF, 0xFF, 0xFF,                // "0049
123 9876543"
        0xFF,
        0xFF
ENDARRAY
BEGINARRAY(DATA_ADN_02, 15)
        0xFF,
        0xFF,
        0xFF,
        0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF,
        0xFF,
        0xFF
ENDARRAY

/*--------------------------------------- arrays -------------------------------
---------------------- */
/* --- phone number  "01234567"---*/
BEGINARRAY_PART (PHN_NUM0, 8)
        0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07
ENDARRAY
/* --- phone number  "4901234567"---*/
```

```
BEGINARRAY_PART (PHN_NUM1, 10)
      0x04, 0x09, 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07
ENDARRAY
/* --- empty phone number ---*/
BEGINARRAY_PART (EMPTY_PHN_NUM, 1)
      0x00
ENDARRAY

/* EF ECC field array */
BEGINARRAY (A_ECC_FIELD,12) 0x11, 0xF2, 0xFF, 0x99, 0xF9, 0xFF, 0x21, 0x43, 0x65, 0xFF, 0xFF,
0xFF ENDARRAY

/*--------------------------------------- structures ----------------------------
------------------------ */

/* called_party (Called party BCD number) */
BEGIN_PSTRUCT ("called_party", CLED_PARTY0)
      SET_COMP ("ton",     TON_UNKNOWN)
      SET_COMP ("npi",     NPI_ISDN_TEL_NUMB_PLAN)
      SET_COMP ("c_called_num",  NUM_8)
      SET_COMP ("called_num",    PHN_NUM0)
ENDSTRUCT
BEGIN_PSTRUCT ("called_party", CLED_PARTY1)
      SET_COMP ("ton",     TON_INT_NUMB)
      SET_COMP ("npi",     NPI_ISDN_TEL_NUMB_PLAN)
      SET_COMP ("c_called_num",  NUM_10)
      SET_COMP ("called_num",    PHN_NUM1)
ENDSTRUCT
/* called_party_sub (Called party subaddress) */
 BEGIN_PSTRUCT ("called_party_sub", CLED_PARTY_SUB_NONE)
      SET_COMP ("tos",     TOS_NOT_PRES)
      SET_COMP ("odd_even",      OE_EVEN)
      SET_COMP ("c_subaddr",     NUM_0)
      SET_COMP ("subaddr",       EMPTY_PHN_NUM)
ENDSTRUCT

/* calling_party (Calling party BCD number) */
BEGIN_PSTRUCT ("calling_party", CLING_PARTY0)
      SET_COMP ("ton",     TON_UNKNOWN)
      SET_COMP ("npi",     NPI_ISDN_TEL_NUMB_PLAN)
      SET_COMP ("present",PRES_PRES_ALLOW)
      SET_COMP ("screen", SCREEN_USER_PROV_VER_PASS)
      SET_COMP ("c_num", NUM_8)
      SET_COMP ("num",   PHN_NUM0)
ENDSTRUCT
```

```
BEGIN_PSTRUCT ("calling_party", CLING_PARTY1)
        SET_COMP ("ton",      TON_INT_NUMB)
        SET_COMP ("npi",      NPI_ISDN_TEL_NUMB_PLAN)
        SET_COMP ("present",PRES_PRES_ALLOW)
        SET_COMP ("screen", SCREEN_USER_PROV_VER_PASS)
        SET_COMP ("c_num", NUM_8)
        SET_COMP ("num",      PHN_NUM0)
ENDSTRUCT
/* calling_party_sub (Calling party subaddress) */
 BEGIN_PSTRUCT ("calling_party_sub", CLING_PARTY_SUB_NONE)
        SET_COMP ("tos",      TOS_NOT_PRES)
        SET_COMP ("odd_even",        OE_EVEN)
        SET_COMP ("c_subaddr",       NUM_0)
        SET_COMP ("subaddr",         EMPTY_PHN_NUM)
ENDSTRUCT

/* redirecting_party (redirecting party BCD number) */
BEGIN_PSTRUCT ("redirecting_party", REDIR_PARTY)
        SET_COMP ("ton",      TON_UNKNOWN)
        SET_COMP ("npi",      NPI_ISDN_TEL_NUMB_PLAN)
        SET_COMP ("present",NUM_0)
        SET_COMP ("screen", NUM_0)
        SET_COMP ("c_redir_num",     NUM_8)
        SET_COMP ("redir_num",       PHN_NUM0)
ENDSTRUCT
/* redirecting_party_sub (redirecting party subaddress) */
 BEGIN_PSTRUCT ("redirecting_party_sub", REDIR_PARTY_SUB_NONE)
        SET_COMP ("tos",      TOS_NOT_PRES)
        SET_COMP ("odd_even",        OE_EVEN)
        SET_COMP ("c_subaddr",       NUM_0)
        SET_COMP ("subaddr",         EMPTY_PHN_NUM)
ENDSTRUCT

/* connected_number (Connected number) */
BEGIN_PSTRUCT ("connected_number", CONNECTED_NUMBER0)
        SET_COMP ("ton",      TON_UNKNOWN)
        SET_COMP ("npi",      NPI_ISDN_TEL_NUMB_PLAN)
        SET_COMP ("present",TOS_NOT_PRES)
        SET_COMP ("screen", SCREEN_USER_PROV_NOT_SCREEN)
        SET_COMP ("c_num", NUM_8)
        SET_COMP ("num",      PHN_NUM0)
ENDSTRUCT
BEGIN_PSTRUCT ("connected_number", CONNECTED_NUMBER1)
        SET_COMP ("ton",      TON_INT_NUMB)
        SET_COMP ("npi",      NPI_ISDN_TEL_NUMB_PLAN)
        SET_COMP ("present",TOS_NOT_PRES)
        SET_COMP ("screen", SCREEN_USER_PROV_NOT_SCREEN)
        SET_COMP ("c_num", NUM_10)
        SET_COMP ("num",   PHN_NUM1)
ENDSTRUCT

/*--- chanel using mode (voice only) */
BEGIN_PSTRUCT("chm", CHM_VOICE)
        SET_COMP("ch_type", CH_SDCCH)
        SET_COMP("ch_mode",        CHM_SPEECH)
ENDSTRUCT
/* --- IMSI --- */
BEGIN_PSTRUCT ("imsi_field", IMSI_FIELD)
        SET_COMP ("c_field", 0x09)
        SET_COMP ("field",   IMSI_FIELD_DATA)
ENDSTRUCT
/* --- preferd PLMN --- */
BEGIN_PSTRUCT ("pref_plmn", PREF_PLMN)
        SET_COMP ("c_pref", 0x60)
        SET_COMP ("pref",   PREF_PLMN_DATA)
ENDSTRUCT

/* ------------------------------------Transparent structs--------------------------*/
/* bearer service DATA 300 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_300)
        SET_COMP ("rate",   UR_0_3_KBIT)
        SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
        SET_COMP ("conn_elem",      CONN_ELEM_TRANS)
        SET_COMP ("stop_bits",      STOP_1_BIT)
        SET_COMP ("data_bits",      DATA_8_BIT)
        SET_COMP ("parity", PARITY_NONE)
        SET_COMP ("flow_control",   NO_FLOW_CONTROL)
        SET_COMP ("modem_type",     MT_V21)
ENDSTRUCT

/* bearer service DATA 300 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_300_TM_NONE)
        SET_COMP ("rate",   UR_0_3_KBIT)
        SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
        SET_COMP ("conn_elem",      CONN_ELEM_TRANS)
        SET_COMP ("stop_bits",      STOP_1_BIT)
        SET_COMP ("data_bits",      DATA_8_BIT)
        SET_COMP ("parity", PARITY_NONE)
        SET_COMP ("flow_control",   NO_FLOW_CONTROL)
        SET_COMP ("modem_type",     MT_NONE)
ENDSTRUCT

/* bearer service DATA 1200 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_1200)
```

```
      SET_COMP ("rate",    UR_1_2_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",     MT_V22)
ENDSTRUCT

/* bearer service DATA 1200/75 bps */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_1200_75)
      SET_COMP ("rate",    UR_1_2_KBIT_V23)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",     MT_V23)
ENDSTRUCT
/* bearer service DATA 2400 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_2400)
      SET_COMP ("rate",    UR_2_4_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",     MT_V22_BIS)
ENDSTRUCT
/* bearer service DATA 2400 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_2400_V26)
      SET_COMP ("rate",    UR_2_4_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",     MT_V26_TER)
ENDSTRUCT
/* bearer service DATA 4800 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_4800)
      SET_COMP ("rate",    UR_4_8_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",   PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",     MT_V32)
ENDSTRUCT
/* bearer service transparent data 9600 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_9600)
      SET_COMP ("rate",     UR_9_6_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",   PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",     MT_V32)
ENDSTRUCT
/* bearer service transparent data 9600 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_9600_V34)
      SET_COMP ("rate",     UR_9_6_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",   PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",     MT_V34)
ENDSTRUCT
/* bearer service transparent data 14400 V34 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_14400_V34)
      SET_COMP ("rate",     UR_14_4_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",   PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",     MT_V34)
ENDSTRUCT
/* bearer service DATA 1200 with speed Nr. 34 (GSM 7.07)*/
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_1200_TM_NONE)
      SET_COMP ("rate",     UR_1_2_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
```

```
        SET_COMP ("parity",  PARITY_NONE)
        SET_COMP ("flow_control",    NO_FLOW_CONTROL)
        SET_COMP ("modem_type",      MT_NONE)
ENDSTRUCT
/* bearer service DATA 2400 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_2400_V120)
        SET_COMP ("rate",    UR_2_4_KBIT)
        SET_COMP ("bearer_serv",     BEARER_SERV_ASYNC)
        SET_COMP ("conn_elem",       CONN_ELEM_TRANS)
        SET_COMP ("stop_bits",       STOP_1_BIT)
        SET_COMP ("data_bits",       DATA_8_BIT)
        SET_COMP ("parity",  PARITY_NONE)
        SET_COMP ("flow_control",    NO_FLOW_CONTROL)
        SET_COMP ("modem_type",      MT_NONE)
ENDSTRUCT
/* bearer service DATA 4800 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_4800_V120)
        SET_COMP ("rate",    UR_4_8_KBIT)
        SET_COMP ("bearer_serv",     BEARER_SERV_ASYNC)
        SET_COMP ("conn_elem",       CONN_ELEM_TRANS)
        SET_COMP ("stop_bits",       STOP_1_BIT)
        SET_COMP ("data_bits",       DATA_8_BIT)
        SET_COMP ("parity",  PARITY_NONE)
        SET_COMP ("flow_control",    NO_FLOW_CONTROL)
        SET_COMP ("modem_type",      MT_NONE)
ENDSTRUCT
/* bearer service transparent data 9600 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_9600_V120)
        SET_COMP ("rate",    UR_9_6_KBIT)
        SET_COMP ("bearer_serv",     BEARER_SERV_ASYNC)
        SET_COMP ("conn_elem",       CONN_ELEM_TRANS)
        SET_COMP ("stop_bits",       STOP_1_BIT)
        SET_COMP ("data_bits",       DATA_8_BIT)
        SET_COMP ("parity",  PARITY_NONE)
        SET_COMP ("flow_control",    NO_FLOW_CONTROL)
        SET_COMP ("modem_type",      MT_NONE)
ENDSTRUCT

/* bearer service transparent data 9600 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_AUTO)
        SET_COMP ("rate",    UR_9_6_KBIT)
        SET_COMP ("bearer_serv",     BEARER_SERV_ASYNC)
        SET_COMP ("conn_elem",       CONN_ELEM_TRANS)
        SET_COMP ("stop_bits",       STOP_1_BIT)
        SET_COMP ("data_bits",       DATA_8_BIT)
        SET_COMP ("parity",  PARITY_NONE)
        SET_COMP ("flow_control",    NO_FLOW_CONTROL)
```

```
      SET_COMP ("modem_type",   MT_AUTOBAUD)
ENDSTRUCT

/* bearer service transparent data 14400 V120 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_TRA_14400_V120)
      SET_COMP ("rate",   UR_14_4_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",   MT_NONE)
ENDSTRUCT

/* --------------------------------Nontransparent structs--------------------------*/
/* bearer service DATA 300 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_300)
      SET_COMP ("rate",   UR_0_3_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_NON_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",   MT_V21)
ENDSTRUCT

/* bearer service DATA 300 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_300_TM_NONE)
      SET_COMP ("rate",   UR_0_3_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_NON_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",   MT_NONE)
ENDSTRUCT

/* bearer service DATA 1200 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_1200)
      SET_COMP ("rate",   UR_1_2_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_NON_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
```

```
        SET_COMP ("parity",  PARITY_NONE)
        SET_COMP ("flow_control",    NO_FLOW_CONTROL)
        SET_COMP ("modem_type",      MT_V22)
ENDSTRUCT

/* bearer service DATA 1200/75 bps */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_1200_75)
        SET_COMP ("rate",    UR_1_2_KBIT_V23)
        SET_COMP ("bearer_serv",     BEARER_SERV_ASYNC)
        SET_COMP ("conn_elem",       CONN_ELEM_NON_TRANS)
        SET_COMP ("stop_bits",       STOP_1_BIT)
        SET_COMP ("data_bits",       DATA_8_BIT)
        SET_COMP ("parity",  PARITY_NONE)
        SET_COMP ("flow_control",    NO_FLOW_CONTROL)
        SET_COMP ("modem_type",      MT_V23)
ENDSTRUCT
/* bearer service DATA 2400 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_2400)
        SET_COMP ("rate",    UR_2_4_KBIT)
        SET_COMP ("bearer_serv",     BEARER_SERV_ASYNC)
        SET_COMP ("conn_elem",       CONN_ELEM_NON_TRANS)
        SET_COMP ("stop_bits",       STOP_1_BIT)
        SET_COMP ("data_bits",       DATA_8_BIT)
        SET_COMP ("parity",  PARITY_NONE)
        SET_COMP ("flow_control",    NO_FLOW_CONTROL)
        SET_COMP ("modem_type",      MT_V22_BIS)
ENDSTRUCT
/* bearer service DATA 2400 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_2400_V26)
        SET_COMP ("rate",    UR_2_4_KBIT)
        SET_COMP ("bearer_serv",     BEARER_SERV_ASYNC)
        SET_COMP ("conn_elem",       CONN_ELEM_NON_TRANS)
        SET_COMP ("stop_bits",       STOP_1_BIT)
        SET_COMP ("data_bits",       DATA_8_BIT)
        SET_COMP ("parity",  PARITY_NONE)
        SET_COMP ("flow_control",    NO_FLOW_CONTROL)
        SET_COMP ("modem_type",      MT_V26_TER)
ENDSTRUCT
/* bearer service DATA 4800 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_4800)
        SET_COMP ("rate",    UR_4_8_KBIT)
        SET_COMP ("bearer_serv",     BEARER_SERV_ASYNC)
        SET_COMP ("conn_elem",       CONN_ELEM_NON_TRANS)
        SET_COMP ("stop_bits",       STOP_1_BIT)
        SET_COMP ("data_bits",       DATA_8_BIT)
        SET_COMP ("parity",  PARITY_NONE)
        SET_COMP ("flow_control",    NO_FLOW_CONTROL)
```

```
      SET_COMP ("modem_type",   MT_V32)
ENDSTRUCT
/* bearer service transparent data 9600 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_9600)
      SET_COMP ("rate",   UR_9_6_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_NON_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",     MT_V32)
ENDSTRUCT

/* bearer service transparent data 9600 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_9600_V34)
      SET_COMP ("rate",   UR_9_6_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_NON_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",     MT_V34)
ENDSTRUCT

/* bearer service transparent data 14400 V34 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_14400_V34)
      SET_COMP ("rate",   UR_14_4_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_NON_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",     MT_V34)
ENDSTRUCT
/* bearer service DATA 1200 with speed Nr. 34 (GSM 7.07)*/
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_1200_TM_NONE)
      SET_COMP ("rate",   UR_1_2_KBIT)
      SET_COMP ("bearer_serv",    BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",      CONN_ELEM_NON_TRANS)
      SET_COMP ("stop_bits",      STOP_1_BIT)
      SET_COMP ("data_bits",      DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",   NO_FLOW_CONTROL)
      SET_COMP ("modem_type",   MT_NONE)
ENDSTRUCT
/* bearer service DATA 2400 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_2400_V120)
      SET_COMP ("rate",   UR_2_4_KBIT)
      SET_COMP ("bearer_serv",   BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",     CONN_ELEM_NON_TRANS)
      SET_COMP ("stop_bits",     STOP_1_BIT)
      SET_COMP ("data_bits",     DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",  NO_FLOW_CONTROL)
      SET_COMP ("modem_type",    MT_NONE)
ENDSTRUCT
/* bearer service DATA 4800 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_4800_V120)
      SET_COMP ("rate",   UR_4_8_KBIT)
      SET_COMP ("bearer_serv",   BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",     CONN_ELEM_NON_TRANS)
      SET_COMP ("stop_bits",     STOP_1_BIT)
      SET_COMP ("data_bits",     DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",  NO_FLOW_CONTROL)
      SET_COMP ("modem_type",    MT_NONE)
ENDSTRUCT
/* bearer service transparent data 9600 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_9600_V120)
      SET_COMP ("rate",   UR_9_6_KBIT)
      SET_COMP ("bearer_serv",   BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",     CONN_ELEM_NON_TRANS)
      SET_COMP ("stop_bits",     STOP_1_BIT)
      SET_COMP ("data_bits",     DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",  NO_FLOW_CONTROL)
      SET_COMP ("modem_type",    MT_NONE)
ENDSTRUCT
/* bearer service transparent data 14400 V120 */
 BEGIN_PSTRUCT ("bcpara", S_BS_DAT_NTRA_14400_V120)
      SET_COMP ("rate",   UR_14_4_KBIT)
      SET_COMP ("bearer_serv",   BEARER_SERV_ASYNC)
      SET_COMP ("conn_elem",     CONN_ELEM_NON_TRANS)
      SET_COMP ("stop_bits",     STOP_1_BIT)
      SET_COMP ("data_bits",     DATA_8_BIT)
      SET_COMP ("parity",  PARITY_NONE)
      SET_COMP ("flow_control",  NO_FLOW_CONTROL)
      SET_COMP ("modem_type",    MT_NONE)
ENDSTRUCT
```

4 TEST CASES

4.1 Initialisation (ACICC001 - ACICC10)

4.1.1 ACICC001: Initialisierung

Description:

Preamble:

None

```
    APL                          ACI                          PS
     |                            |                            |
COMMAND (TAP RESET)
COMMAND (CC RESET)
COMMAND (MM RESET)
COMMAND (SIM RESET)
COMMAND (SS RESET)
COMMAND (MMI RESET)
COMMAND (SMS RESET)
COMMAND (PL RESET)
      |                           |                            |
COMMAND (TAP REDIRECT CLEAR)
COMMAND (CC REDIRECT CLEAR)
COMMAND (MM REDIRECT CLEAR)
COMMAND (SIM REDIRECT CLEAR)
COMMAND (SS REDIRECT CLEAR)
COMMAND (MMI REDIRECT CLEAR)
COMMAND (SMS REDIRECT CLEAR)
COMMAND (PL REDIRECT CLEAR)
      |                           |                            |
COMMAND (MMI REDIRECT CC TAP)
COMMAND (MMI REDIRECT MM TAP)
COMMAND (MMI REDIRECT SIM TAP)
COMMAND (MMI REDIRECT SS TAP)
COMMAND (MMI REDIRECT MMI TAP)
COMMAND (MMI REDIRECT SMS TAP)
COMMAND (MMI REDIRECT T30 TAP)
COMMAND (MMI REDIRECT L2R TAP)
COMMAND (MMI REDIRECT RA TAP)
COMMAND (PL REDIRECT MMI NULL)
     |                            |                            |
COMMAND (TAP REDIRECT TAP MMI)
COMMAND (MMI REDIRECT MMI TAP)
     |                            |                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value

History: 14.12.98 AK Initial

4.1.2 ACICC002: use verbose <err> values

Description:

Preamble:

ACICC001

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |       (cmd: +CMEE=2)       |                            |
     *===========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<===========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CMEE_S
	cmd_seq	C_CMEE_S2
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 11.11.99 DAK Initial

4.1.3 ACICC003: initialize phonebook

Description:

initialize phonebook - ADN only

Preamble:

```
             ACICC002
       APL                        ACI                          PS
        |                          |                           |
(1)     |        ACI_CMD_REQ       |                           |
        |      (cmd: +CFUN=1)      |                           |
        *==========================>*                          |
(2)     |                          |     SIM_ACTIVATE_REQ      |
        |                          *==========================>*
(3)     |                          |     SIM_ACTIVATE_CNF      |
        |                          *<==========================*
(4)     |                          |    SIM_MMI_INSERT_IND     |
        |                          *<==========================*
(5)     |                          |       SIM_READ_REQ        |
        |                          *==========================>*
(6)     |                          |       SIM_READ_CNF        |
        |                          *<==========================*
(7)     |        ACI_CMD_IND       |                           |
        |        (cmd: OK)         |                           |
        *<==========================*                          |
(8)     |                          |     MNSMS_REPORT_IND      |
        |                          *<==========================*
(9)     |                          |    SIM_READ_RECORD_REQ    |
        |                          *==========================>*
(10)    |                          |    SIM_READ_RECORD_CNF    |
        |                          *<==========================*
(11)    |                          |    SIM_READ_RECORD_REQ    |
        |                          *==========================>*
(12)    |                          |    SIM_READ_RECORD_CNF    |
        |                          *<==========================*
        |                          |                           |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CFUN_S
	cmd_seq	C_CFUN_S
(2) SIM_ACTIVATE_REQ		
	proc	SIM_INITIALISATION
	mmi_pro_file	MMI_PRO_FILE
	stk_pro_file	NOT_USED
(3) SIM_ACTIVATE_CNF		
	cause	SIM_NO_ERROR
	pin_cnt	NUM_3
	puk_cnt	NUM_3
	pin2_cnt	NUM_3
	puk2_cnt	NUM_3

	ec_code	EC_CODES
	pref_lang	NO_PREF_LANG
(4) SIM_MMI_INSERT_IND		
	func	SIM_ADN_ENABLED
	sim_serv	SIM_SERV_ADN
	imsi_field	IMSI_FIELD
	pref_plmn	PREF_PLMN
	phase	PHASE_2_SIM
	access_acm	ACCESS_ALWAYS
	access_acmmax	ACCESS_ALWAYS
	access_puct	ACCESS_ALWAYS
(5) SIM_READ_REQ		
	source	SRC_MMI
	offset	NUM_0
	datafield	SIM_ECC
	length	NOT_PRESENT_8BIT
	max_length	NUM_0
(6) SIM_READ_CNF		
	datafield	SIM_ECC
	cause	SIM_NO_ERROR
	length	NUM_12
	trans_data	A_ECC_FIELD
(7) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(8) MNSMS_REPORT_IND		
	state	SMS_STATE_READY
(9) SIM_READ_RECORD_REQ		
	source	SRC_MMI
	datafield	SIM_ADN
	record	NUM_1
	length	MAX_DATA
(10) SIM_READ_RECORD_CNF		
	datafield	SIM_ADN
	cause	SIM_NO_ERROR
	record	NUM_1
	max_record	NUM_2
	length	LDATA_ADN_01
	linear_data	DATA_ADN_01
(11) SIM_READ_RECORD_REQ		
	source	SRC_MMI
	datafield	SIM_ADN

	record	NUM_2
	length	MAX_ADN_DATA
(12) SIM_READ_RECORD_CNF		
	datafield	SIM_ADN
	cause	SIM_NO_ERROR
	record	NUM_2
	max_record	NUM_2
	length	LDATA_ADN_02
	linear_data	DATA_ADN_02

History:	03.02.2000	DAK	Initial
	15-Mar-2002	HM	MNSMS_REPORT_IND changed

4.2 Select type of address "+CSTA"(ACICC011 - ACICC020)

4.2.1 ACICC011: list of supported modes

Description:

Type of Address, list of supported types

Preamble:

ACICC002

```
    APL                        ACI                        PS
     |                          |                          |
(1)  |        ACI_CMD_REQ       |                          |
     |       (cmd: +CSTA=?)     |                          |
     *=========================>*                          |
(2)  |        ACI_CMD_IND       |                          |
     |   (cmd: +CSTA: 129,145)  |                          |
     *<=========================*                          |
(3)  |        ACI_CMD_IND       |                          |
     |         (cmd: OK)        |                          |
     *<=========================*                          |
     |                          |                          |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CSTA_T
	cmd_seq	C_CSTA_T
(2) ACI_CMD_IND		
	cmd_len	LM_CSTA_T
	cmd_seq	M_CSTA_T

(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.10.99 DAK Initial

4.2.2 ACICC012: testing initial settings

Description:

Type of Address, test of initial settings

Preamble:

ACICC002

```
    APL                           ACI                           PS
     |                             |                             |
(1)  |         ACI_CMD_REQ         |                             |
     |        (cmd: +CSTA?)        |                             |
     *===========================>*                             |
(2)  |         ACI_CMD_IND         |                             |
     |      (cmd: +CSTA: 129)      |                             |
     *<===========================*                             |
(3)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<===========================*                             |
     |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CSTA_Q
	cmd_seq	C_CSTA_Q
(2) ACI_CMD_IND		
	cmd_len	LM_CSTA_Q
	cmd_seq	M_CSTA_Q1
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.10.99 DAK Initial

4.2.3 ACICC013: setting values and test whether they were setted

Description:

Type of Address, setting types and test whether type was setted

Preamble:

ACICC002

Variants: <A>....<B>

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |         ACI_CMD_REQ        |                            |
     |   (cmd: +CSTA (145||129)   |                            |
     *===========================>*                            |
(2)  |         ACI_CMD_IND        |                            |
     |          (cmd: OK)         |                            |
     *<===========================*                            |
(3)  |         ACI_CMD_REQ        |                            |
     |        (cmd: +CSTA?)       |                            |
     *===========================>*                            |
(4)  |         ACI_CMD_IND        |                            |
     |  (cmd: +CSTA:(145||129))   |                            |
     *<===========================*                            |
(5)  |         ACI_CMD_IND        |                            |
     |          (cmd: OK)         |                            |
     *<===========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CSTA_S
<A>	cmd_seq	C_CSTA_S0
<B>	cmd_seq	C_CSTA_S1
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(3) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CSTA_Q
	cmd_seq	C_CSTA_Q
(4) ACI_CMD_IND		
	cmd_len	LM_CSTA_Q
<A>	cmd_seq	M_CSTA_Q0
<B>	cmd_seq	M_CSTA_Q1

(5) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.10.99 DAK Initial

4.2.4 ACICC014: trying to set illegal values

Description:

Select Type of Address, set illegal mode

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |         ACI_CMD_REQ        |                            |
     |      (cmd: +CSTA=255)      |                            |
     *==========================>*                             |
(2)  |         ACI_CMD_IND        |                            |
     |        (cmd: ERROR)        |                            |
     *<==========================*                             |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CSTA_S
	cmd_seq	C_CSTA_S9
(2) ACI_CMD_IND		
	cmd_len	LM_CME_ERR_INV_OPP
	cmd_seq	M_CME_ERR_INV_OPP

History: 06.10.99 DAK Initial

4.3 Ordinate call "D"(ACICC021 - ACICC035)

4.3.1 ACICC021: setting calling line id present. and connection line id restriction

Description:

setting calling line id present. and connection line id restriction

Preamble:

ACICC002

```
    APL                        ACI                        PS
     |                          |                          |
(1)  |        ACI_CMD_REQ       |                          |
     |   (cmd: +CLIR=,+COLP=)   |                          |
     *=========================>*                          |
(2)  |        ACI_CMD_IND       |                          |
     |         (cmd: OK)        |                          |
     *<=========================*                          |
     |                          |                          |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CLIR_COLP_S
	cmd_seq	C_CLIR_COLP_S
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 17.11.99 DAK Initial

4.3.2 ACICC022: establish successful MO voice call

Description:

establish successful MO voice call

Variants:

<A>...<B>

Preamble:

ACICC021

```
     APL                          ACI                          PS
      |                            |                            |
(1)   |        ACI_CMD_REQ         |                            |
      |      (cmd: D0123456)       |                            |
      *===========================>*                            |
(2)   |                            |       MNCC_SETUP_REQ       |
      |                            *===========================>*
   MUTE (500)
(3)   |                            |   MNCC_CALL_PROCEED_IND    |
      |                            *<===========================*
(4)   |                            |       MNCC_ALERT_IND       |
      |                            *<===========================*
(5)   |                            |       MNCC_SETUP_CNF       |
      |                            *<===========================*
(6)   |                            |        SIM_SYNC_REQ        |
      |                            *===========================>*
(7)   |                            |        SIM_SYNC_CNF        |
      |                            *<===========================*
(8)   |        ACI_CMD_IND         |                            |
      |     (cmd: +COPL: ...)      |                            |
      *<===========================*                            |
(9)   |        ACI_CMD_IND         |                            |
      |         (cmd: OK)          |                            |
      *<===========================*                            |
      |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_D0
<A>	cmd_seq	C_D0
<B>	cmd_len	LC_D1
<B>	cmd_seq	C_D1
(2) MNCC_SETUP_REQ		
	ti	NUM_0
	prio	PRIO_NORM_CALL
	ri	RI_NOT_PRES
	bcpara	BC_PARA_SPEECH
	bcpara2	BC_PARA_NO_SERVICE

<A>	called_party	CLED_PARTY0
<B>	called_party	CLED_PARTY1
	called_party_sub	CLED_PARTY_SUB_NONE
	clir_sup	CLR_SUP
	fac_inf	NOT_USED
(3) MNCC_CALL_PROCEED_IND		
	ti	NUM_0
	progress_desc	NOT_SPEC
	ri	RI_NOT_PRES
	bcpara	BC_PARA_SPEECH
	bcpara2	BC_PARA_NO_SERVICE
(4) MNCC_ALERT_IND		
	ti	NUM_0
	progress_desc	NOT_SPEC
(5) MNCC_SETUP_CNF		
	ti	NUM_0
	cause	MNCC_CAUSE_SUCCESS
	progress_desc	NOT_SPEC
<A>	connected_number	CONNECTED_NUMBER0
<B>	connected_number	CONNECTED_NUMBER1
	connected_number_sub	NOT_USED
(6) SIM_SYNC_REQ		
	synccs	0x01
(7) SIM_SYNC_CNF		
	cause	NOT_SPEC
(8) ACI_CMD_IND		
<A>	cmd_len	LM_D0
<A>	cmd_seq	M_D0
<B>	cmd_len	LM_D1
<B>	cmd_seq	M_D1
(9) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 12.01.2000 DAK Initial

4.3.3 ACICC023: dialing number using phonebook

Description:

performe successful dail using phonebook

Preamble:

ACICC003

Variants:

<A>...<C>

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |      (cmd: +CBPS=...)      |                            |
     *==========================>*                             |
(2)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<==========================*                             |
(3)  |        ACI_CMD_REQ         |                            |
     |   (cmd: D>"Meier,Max")     |                            |
     *==========================>*                             |
(4)  |        ACI_CMD_IND         |                            |
     |      (cmd: CME ERROR)      |                            |
     *<==========================*                             |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CPBS_S
	cmd_seq	C_CPBS_S
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(3) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_PHB_D1
<A>	cmd_seq	C_PHB_D1
<B>	cmd_len	LC_PHB_D2
<B>	cmd_seq	C_PHB_D2
<C>	cmd_len	LC_PHB_D3
<C>	cmd_seq	C_PHB_D3
(4) ACI_CMD_IND		
<A>	cmd_len	LM_OK
<A>	cmd_seq	M_OK
<B>	cmd_len	LM_OK
<B>	cmd_seq	M_OK
<C>	cmd_len	LM_CME_ERR_INV_OPP
<C>	cmd_seq	M_CME_ERR_INV_OPP

History: 03.02.2000 DAK Initial

4.3.4 ACICC024: try to establish MO voice call – no connection (REJECT_IND)

Description:

try to establish MO voice call – no conncetion

Preamble:

ACICC021

Variants:

<A>...<E>

```
    APL                             ACI                             PS
     |                               |                               |
(1)  |         ACI_CMD_REQ           |                               |
     |       (cmd: D0123456)         |                               |
     *==========================>*                               |
(2)  |                               |       MNCC_SETUP_REQ          |
     |                               *===========================>*
(3)  |                               |       MNCC_REJECT_IND         |
     |                               *<===========================*
(4)  |                               |        SIM_SYNC_REQ           |
     |                               *===========================>*
(5)  |                               |        SIM_SYNC_CNF           |
     |                               *<===========================*
(6)  |                               |        SIM_SYNC_REQ           |
     |                               *===========================>*
(7)  |                               |        SIM_SYNC_CNF           |
     |                               *<===========================*
(8)  |         ACI_CMD_IND           |                               |
     |      (cmd: +COLP: ...)        |                               |
     *<===========================*                               |
     |                               |                               |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_D0
	cmd_seq	C_D0
(2) MNCC_SETUP_REQ		
	ti	NUM_0
	prio	PRIO_NORM_CALL
	ri	RI_NOT_PRES
	bcpara	BC_PARA_SPEECH
	bcpara2	BC_PARA_NO_SERVICE
	called_party	CLED_PARTY0
	called_party_sub	CLED_PARTY_SUB_NONE
	clir_sup	CLR_SUP
	fac_inf	NOT_USED

(3) MNCC_REJECT_IND

	ti	NUM_0
<A>	cause	MNCC_CAUSE_MS_TIMER
<B>	cause MNCC_CAUSE_MS_INCOMPAT_DEST	
<C>	cause MMCS_AUTHENTICATION_REJECTED	
<D>	cause	MMCS_UNSPECIFIED
<E>	cause	MMCS_INT_NOT_PRESENT

(4) SIM_SYNC_REQ

	synccs	NUM_1

(5) SIM_SYNC_CNF

	cause	NOT_SPEC

(6) SIM_SYNC_REQ

	synccs	NUM_2

(7) SIM_SYNC_CNF

	cause	NOT_SPEC

(8) ACI_CMD_IND

<A>	cmd_len	LM_NO_CARRIER
<A>	cmd_seq	M_NO_CARRIER
<B>	cmd_len	LM_NO_CARRIER
<B>	cmd_seq	M_NO_CARRIER
<C>	cmd_len	LM_NO_CARRIER
<C>	cmd_seq	M_NO_CARRIER
<D>	cmd_len	LM_NO_CARRIER
<D>	cmd_seq	M_NO_CARRIER
<E>	cmd_len	LM_NO_CARRIER
<E>	cmd_seq	M_NO_CARRIER

History: 12.01.2000 DAK Initial

4.3.5 ACICC025: try to establish MO voice call – no connection (RELEASE_IND)

Description:

try to establish MO voice call – no conncetion

Preamble:

ACICC021

Variants:

<A>...<H>

```
    APL                          ACI                          PS
      |                           |                           |
(1)   |        ACI_CMD_REQ        |                           |
      |      (cmd: D0123456)      |                           |
      *==========================>*                           |
(2)   |                           |      MNCC_SETUP_REQ       |
      |                           *==========================>*
(3)   |                           |     MNCC_RELEASE_IND      |
      |                           *<==========================*
(4)   |                           |       SIM_SYNC_REQ        |
      |                           *==========================>*
(5)   |                           |       SIM_SYNC_CNF        |
      |                           *<==========================*
(6)   |                           |       SIM_SYNC_REQ        |
      |                           *==========================>*
(7)   |                           |       SIM_SYNC_CNF        |
      |                           *<==========================*
(8)   |        ACI_CMD_IND        |                           |
      |     (cmd: +COLP: ...)     |                           |
      *<==========================*                           |
      |                           |                           |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_D0
	cmd_seq	C_D0
(2) MNCC_SETUP_REQ		
	ti	NUM_0
	prio	PRIO_NORM_CALL
	ri	RI_NOT_PRES
	bcpara	BC_PARA_SPEECH
	bcpara2	BC_PARA_NO_SERVICE
	called_party	CLED_PARTY0
	called_party_sub	CLED_PARTY_SUB_NONE
	clir_sup	CLR_SUP
	fac_inf	NOT_USED
(3) MNCC_RELEASE_IND		
	ti	NUM_0
<A>	cause	MNCC_CAUSE_USER_BUSY
<B>	cause MNCC_CAUSE_ALERT_NO_ANSWER	
<C>	cause	MNCC_CAUSE_UNASSIGN
<D>	cause	MNCC_CAUSE_NO_ROUTE
<E>	cause	MNCC_CAUSE_NO_RESPONSE
<F>	cause	MNCC_CAUSE_DEST_ORDER

<G>	cause	MNCC_CAUSE_NUM_FORMAT
<H>	cause	MNCC_CAUSE_NO_NET_CAUSE
(4) SIM_SYNC_REQ		
	synccs	NUM_1
(5) SIM_SYNC_CNF		
	cause	NOT_SPEC
(6) SIM_SYNC_REQ		
	synccs	NUM_2
(7) SIM_SYNC_CNF		
	cause	NOT_SPEC
(8) ACI_CMD_IND		
<A>	cmd_len	LM_BUSY
<A>	cmd_seq	M_BUSY
<B>	cmd_len	LM_NO_ANSWER
<B>	cmd_seq	M_NO_ANSWER
<C>	cmd_len	LM_NO_CARRIER
<C>	cmd_seq	M_NO_CARRIER
<D>	cmd_len	LM_NO_CARRIER
<D>	cmd_seq	M_NO_CARRIER
<E>	cmd_len	LM_NO_ANSWER
<E>	cmd_seq	M_NO_ANSWER
<F>	cmd_len	LM_NO_CARRIER
<F>	cmd_seq	M_NO_CARRIER
<G>	cmd_len	LM_NO_CARRIER
<G>	cmd_seq	M_NO_CARRIER
<H>	cmd_len	LM_NO_CARRIER
<H>	cmd_seq	M_NO_CARRIER

History:	12.01.2000	DAK	Initial

4.3.6 ACICC026: Voice Call with no answer by Subscriber, no in-band tones

Description:

try to establish MO voice call - no conncetion

Preamble:

ACICC021

```
      APL                         ACI                          PS
       |                           |                            |
(1)    |         ACI_CMD_REQ       |                            |
       |       (cmd: D0123456)     |                            |
       *==========================>*                            |
(2)    |                           |       MNCC_SETUP_REQ       |
       |                           *==========================>*
(3)    |                           |        SIM_SYNC_REQ        |
       |                           *==========================>*
(4)    |                           |   MNCC_CALL_PROCEED_IND    |
       |                           *<==========================*
(5)    |                           |     MNCC_PROGRESS_IND      |
       |                           *<==========================*
(6)    |                           |       MNCC_ALERT_IND       |
       |                           *<==========================*
(7)    |                           |       MNCC_SYNC_IND        |
       |                           *<==========================*
(8)    |                           |    MNCC_DISCONNECT_IND     |
       |                           *<==========================*
(9)    |                           |        SIM_SYNC_REQ        |
       |                           *==========================>*
(10)   |         ACI_CMD_IND       |                            |
       |      (msg: NO ANSWER)     |                            |
       *<==========================*                            |
(11)   |                           |      MNCC_RELEASE_CNF      |
       |                           *<==========================*
       |                           |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_D0
	cmd_seq	C_D0
(2) MNCC_SETUP_REQ		
	ti	NUM_0
	prio	PRIO_NORM_CALL
	ri	RI_NOT_PRES
	bcpara	BC_PARA_SPEECH
	bcpara2	BC_PARA_NO_SERVICE
	called_party	CLED_PARTY0
	called_party_sub	CLED_PARTY_SUB_NONE
	clir_sup	CLR_SUP
	fac_inf	NOT_USED
(3) SIM_SYNC_REQ		
	synccs	SYNC_START_CALL
(4) MNCC_CALL_PROCEED_IND		
	ti	NUM_0

	progress_desc	PROG_NOT_PRES
	ri	RI_NOT_PRES
	bcpara	BC_PARA_SPEECH
	bcpara2	BC_PARA_NO_SERVICE
(5) MNCC_PROGRESS_IND		
	ti	NUM_0
	progress_desc	PROG_NOT_PRES
(6) MNCC_ALERT_IND		
	ti	NUM_0
	progress_desc	PROG_NOT_PRES
(7) MNCC_SYNC_IND		
	ti	NOT_PRESENT_8BIT
	cause	MNCC_CAUSE_REEST_FINISHED
	chm	CHM_VOICE
(8) MNCC_DISCONNECT_IND		
	ti	NUM_0
	cause	
	MNCC_CAUSE_ALERT_NO_ANSWER	
	diagnostic	NOT_PRESENT_8BIT
	progress_desc	PROG_NOT_PRES
(9) SIM_SYNC_REQ		
	synccs	SYNC_STOP_CALL
(10) ACI_CMD_IND		
	cmd_len	LM_NO_ANSWER
	cmd_seq	M_NO_ANSWER
(11) MNCC_RELEASE_CNF		
	ti	NUM_0
	cause	
	MNCC_CAUSE_ALERT_NO_ANSWER	

History: 23.10.2002 KGT Initial

4.4 Call Termination (ACICC036 - ACICC040)

4.4.1 ACICC036: Call Termination without in-band tones

Description:

call termination procedure without in-band tones active.

Preamble:

ACICC022A

```
    APL                         ACI                         PS
     |                           |                           |
(1)  |                           |    MNCC_DISCONNECT_IND    |
     |                           *<==========================*
(2)  |                           |       SIM_SYNC_REQ        |
     |                           *==========================>*
(3)  |        ACI_CMD_IND        |                           |
     |     (msg: NO CARRIER)     |                           |
     *<==========================*                           |
(4)  |                           |     MNCC_RELEASE_CNF      |
     |                           *<==========================*
     |                           |                           |
```

Parametrization:

	Primitive	Parameter	Value
(1)	MNCC_DISCONNECT_IND		
		ti	NUM_0
		cause	MNCC_CAUSE_CALL_CLEAR
		diagnostic	NOT_PRESENT_8BIT
		progress_desc	PROG_NOT_PRES
(2)	SIM_SYNC_REQ		
		synccs	SYNC_STOP_CALL
(3)	ACI_CMD_IND		
		cmd_len	
		NUM_ELEMENTS(M_NO_CARRIER)	
		cmd_seq	M_NO_CARRIER
(4)	MNCC_RELEASE_CNF		
		ti	NUM_0
		cause	MNCC_CAUSE_CALL_CLEAR

History: 23.10.2002 KGT Initial

4.5 Call mode "+CMOD" (ACICC041 - ACICC050)

4.5.1 ACICC041: listin of supported call modes

Description:

Call Mode, test of supported call modes

Preamble:

ACICC002

```
    APL                          ACI                           PS
     |                            |                             |
(1)  |        ACI_CMD_REQ         |                             |
     |       (cmd: +CMOD=?)       |                             |
     *===========================>*                             |
(2)  |        ACI_CMD_IND         |                             |
     |    (cmd: +CMOD: (0-3))     |                             |
     *<===========================*                             |
(3)  |        ACI_CMD_IND         |                             |
     |         (cmd: OK)          |                             |
     *<===========================*                             |
     |                            |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CMOD_T
	cmd_seq	C_CMOD_T
(2) ACI_CMD_IND		
	cmd_len	LM_CMOD_T
	cmd_seq	M_CMOD_T
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 05.10.99 DAK Initial

4.5.2 ACICC042: getting initial call mode settings

Description:

Call Mode, test of initial mode settings

Preamble:

ACICC002

```
    APL                        ACI                        PS
     |                          |                          |
(1)  |        ACI_CMD_REQ       |                          |
     |       (cmd: +CMOD?)      |                          |
     *=========================>*                          |
(2)  |        ACI_CMD_IND       |                          |
     |      (cmd: +CMOD: 0)     |                          |
     *<=========================*                          |
(3)  |        ACI_CMD_IND       |                          |
     |         (cmd: OK)        |                          |
     *<=========================*                          |
     |                          |                          |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CMOD_Q
	cmd_seq	C_CMOD_Q
(2) ACI_CMD_IND		
	cmd_len	LM_CMOD_Q
	cmd_seq	M_CMOD_Q0
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 05.10.99 DAK Initial

4.5.3 ACICC043: setting several legal call modes, an test whether they are setted

Description:

Call Mode, setting call mode and test whether mode was setted

Preamble:

ACICC002

Variants: <A>....<D>

```
    APL                          ACI                           PS
     |                            |                             |
(1)  |        ACI_CMD_REQ         |                             |
     |     (cmd: +CMOD(0-3))      |                             |
     *==========================>*                             |
(2)  |        ACI_CMD_IND         |                             |
     |         (cmd: OK)          |                             |
     *<==========================*                             |
(3)  |        ACI_CMD_REQ         |                             |
     |       (cmd: +CMOD?)        |                             |
     *==========================>*                             |
(4)  |        ACI_CMD_IND         |                             |
     |    (cmd: +CMOD:(0-3))      |                             |
     *<==========================*                             |
(5)  |        ACI_CMD_IND         |                             |
     |         (cmd: OK)          |                             |
     *<==========================*                             |
     |                            |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CMOD_S
<A>	cmd_seq	C_CMOD_S0
<B>	cmd_seq	C_CMOD_S1
<C>	cmd_seq	C_CMOD_S2
<D>	cmd_seq	C_CMOD_S3
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(3) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CMOD_Q
	cmd_seq	C_CMOD_Q
(4) ACI_CMD_IND		
	cmd_len	LM_CMOD_Q
<A>	cmd_seq	M_CMOD_Q0
<B>	cmd_seq	M_CMOD_Q1
<C>	cmd_seq	M_CMOD_Q2
<D>	cmd_seq	M_CMOD_Q3
(5) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 05.10.99 DAK Initial

4.5.4 ACICC044: trying to set an illegal call mode

Description:

Call Mode, set illegal mode

Preamble:

ACICC002

```
    APL                        ACI                        PS
     |                          |                          |
(1)  |        ACI_CMD_REQ       |                          |
     |      (cmd: +CMOD=4)      |                          |
     *=========================>*                          |
(2)  |        ACI_CMD_IND       |                          |
     |       (cmd: ERROR)       |                          |
     *<=========================*                          |
     |                          |                          |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CMOD_S
	cmd_seq	C_CMOD_S9
(2) ACI_CMD_IND		
	cmd_len	
	NUM_ELEMENTS(M_CME_ERR_INV_PARAM)	
	cmd_seq	M_CME_ERR_INV_PARAM

History: 05.10.99 DAK Initial

4.6 Hang up call "+CHUP" (ACICC051 - ACICC060)

4.6.1 ACICC051: performe test & read command

Description:

performe test & read command

Preamble:

ACICC002

Variants:

<A>...<B>

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |         ACI_CMD_REQ        |                            |
     |    (cmd: +CHUP=?,+CHUP?)   |                            |
     *==========================> *                            |
(2)  |         ACI_CMD_IND        |                            |
     |       (cmd: OK,ERROR)      |                            |
     *<========================== *                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_CHUP_T
<A>	cmd_seq	C_CHUP_T
<B>	cmd_len	LC_CHUP_Q
<B>	cmd_seq	C_CHUP_Q
(2) ACI_CMD_IND		
<A>	cmd_len	LM_OK
<A>	cmd_seq	M_OK
<B>	cmd_len	LM_CME_ERR_INV_OPP
<B>	cmd_seq	M_CME_ERR_INV_OPP

History: 05.01.2000 DAK Initial

4.6.2 ACICC052: hang up an open call

Description:

hang up an open call

Preamble:

ACICC022A

Variants:

<A>...<B>

```
     APL                         ACI                         PS
      |                           |                           |
(1)   |        ACI_CMD_REQ        |                           |
      |       (cmd: +CHUP)        |                           |
      *=========================>*                            |
(4)   |                           |       SIM_SYNC_REQ        |
      |                           *=========================>*
(5)   |                           |       SIM_SYNC_CNF        |
      |                           *<==========================*
(6)   |                           |    MNCC_DISCONNECT_REQ    |
      |                           *=========================>*
(7)   |                           |    MNCC_DISCONNECT_IND    |
      |                           *<==========================*
(8)   |        ACI_CMD_IND        |                           |
      |         (cmd: OK)         |                           |
      *<==========================*                           |
      |                           |                           |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CHUP_S
	cmd_seq	C_CHUP_S
(2) SIM_SYNC_REQ		
	synccs	NUM_2
(3) SIM_SYNC_CNF		
	cause	NOT_SPEC
(4) MNCC_DISCONNECT_REQ		
	ti	NUM_0
	cause	MNCC_CAUSE_CALL_CLEAR
	fac_inf	NOT_USED
	ss_version	SS_VER_NOT_PRES
(5) MNCC_DISCONNECT_IND		
	ti	NUM_0
<A>	cause	MNCC_CAUSE_CALL_CLEAR
<B>	cause	MNCC_CAUSE_MS_TIMER
	diagnostic	DIAG_UNKNOWN_CUG_INDEX
	progress_desc	PROG_NO_END_TO_END_PLMN
(6) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 13.01.2000 DAK Initial

4.7 Select bearer service type "CBST=?" (ACICC061 - ACICC070)

4.7.1 ACICC061: getting list of supported modes

Description:

Select Bearer Service Type, listing of supported modes

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |       (cmd: +CBST=?)       |                            |
     *===========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |     (cmd: +CSTA: ...)      |                            |
     *<===========================*                            |
(3)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<===========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CBST_T
	cmd_seq	C_CBST_T
(2) ACI_CMD_IND		
	cmd_len	LM_CBST_T
	cmd_seq	M_CBST_T
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.10.99 DAK Initial

4.7.2 ACICC062: getting initial bcap settings

Description:

Select Bearer Service Type, test of initial settings

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
      |        (cmd: +CBST?)        |                             |
      *===========================>*                             |
(2)   |         ACI_CMD_IND         |                             |
      |     (cmd: +CBST: 7,0,1)     |                             |
      *<===========================*                             |
(3)   |         ACI_CMD_IND         |                             |
      |          (cmd: OK)          |                             |
      *<===========================*                             |
      |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CBST_Q
	cmd_seq	C_CBST_Q
(2) ACI_CMD_IND		
	cmd_len	LM_CBST_Q0
	cmd_seq	M_CBST_Q14
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 05.10.99 DAK Initial

4.7.3 ACICC063: setting bcap mode, and test settings – PART I

Description:

Select Bearer Service Type, setting types and test whether type was setted

Preamble:

ACICC002

Variants: <A>....<T>

```
       APL                         ACI                         PS
        |                           |                           |
(1)     |        ACI_CMD_REQ        |                           |
        |    (cmd: +CBST x,x,x)     |                           |
        *==========================>*                           |
(2)     |                           |    MNCC_CONFIGURE_REQ     |
        |                           *==========================>*
(3)     |        ACI_CMD_IND        |                           |
        |         (cmd: OK)         |                           |
        *<==========================*                           |
(4)     |        ACI_CMD_REQ        |                           |
        |      (cmd: +CBST?)        |                           |
        *==========================>*                           |
(5)     |        ACI_CMD_IND        |                           |
        |   (cmd: +CSTA: x,x,x)     |                           |
        *<==========================*                           |
(6)     |        ACI_CMD_IND        |                           |
        |         (cmd: OK)         |                           |
        *<==========================*                           |
        |                           |                           |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_CBST_S0
<A>	cmd_seq	C_CBST_S01
<B>	cmd_len	LC_CBST_S0
<B>	cmd_seq	C_CBST_S03
<C>	cmd_len	LC_CBST_S0
<C>	cmd_seq	C_CBST_S05
<D>	cmd_len	LC_CBST_S0
<D>	cmd_seq	C_CBST_S07
<E>	cmd_len	LC_CBST_S0
<E>	cmd_seq	C_CBST_S09
<F>	cmd_len	LC_CBST_S0
<F>	cmd_seq	C_CBST_S11
<G>	cmd_len	LC_CBST_S0
<G>	cmd_seq	C_CBST_S13
<H>	cmd_len	LC_CBST_S1
<H>	cmd_seq	C_CBST_S15
<I>	cmd_len	LC_CBST_S1
<I>	cmd_seq	C_CBST_S17
<J>	cmd_len	LC_CBST_S1
<J>	cmd_seq	C_CBST_S19
<K>	cmd_len	LC_CBST_S1
<K>	cmd_seq	C_CBST_S21
<L>	cmd_len	LC_CBST_S1

<L>	cmd_seq	C_CBST_S23
<M>	cmd_len	LC_CBST_S1
<M>	cmd_seq	C_CBST_S25
<N>	cmd_len	LC_CBST_S1
<N>	cmd_seq	C_CBST_S27
<O>	cmd_len	LC_CBST_S1
<O>	cmd_seq	C_CBST_S29
<P>	cmd_len	LC_CBST_S1
<P>	cmd_seq	C_CBST_S31
<Q>	cmd_len	LC_CBST_S1
<Q>	cmd_seq	C_CBST_S33
<R>	cmd_len	LC_CBST_S1
<R>	cmd_seq	C_CBST_S35
<S>	cmd_len	LC_CBST_S1
<S>	cmd_seq	C_CBST_S37
<T>	cmd_len	LC_CBST_S1
<T>	cmd_seq	C_CBST_S39

(2) MNCC_CONFIGURE_REQ

	called_party_sub	NOT_USED
<A>	bcpara	S_BS_DAT_TRA_300
<B>	bcpara	S_BS_DAT_TRA_1200
<C>	bcpara	S_BS_DAT_TRA_1200_75
<D>	bcpara	S_BS_DAT_TRA_2400
<E>	bcpara	S_BS_DAT_TRA_2400_V26
<F>	bcpara	S_BS_DAT_TRA_4800
<G>	bcpara	S_BS_DAT_TRA_9600
<H>	bcpara	S_BS_DAT_TRA_9600_V34
<I>	bcpara	S_BS_DAT_TRA_14400_V34
<J>	bcpara	S_BS_DAT_TRA_1200_TM_NONE
<K>	bcpara	S_BS_DAT_TRA_2400_V120
<L>	bcpara	S_BS_DAT_TRA_4800_V120
<M>	bcpara	S_BS_DAT_TRA_9600_V120
<N>	bcpara	S_BS_DAT_TRA_14400_V120
<O>	bcpara	S_BS_DAT_TRA_300_TM_NONE
<P>	bcpara	S_BS_DAT_TRA_1200_TM_NONE
<Q>	bcpara	S_BS_DAT_TRA_2400_V120
<R>	bcpara	S_BS_DAT_TRA_4800_V120
<S>	bcpara	S_BS_DAT_TRA_9600_V120
<T>	bcpara	S_BS_DAT_TRA_14400_V120
	sns_mode	SNS_MODE_VOICE
	ctm_ena	CTM_DISABLED

(3) ACI_CMD_IND

	cmd_len	LM_OK
	cmd_seq	M_OK

(4) ACI_CMD_REQ

	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CBST_Q
	cmd_seq	C_CBST_Q

(5) ACI_CMD_IND

<A>	cmd_len	LM_CBST_Q0
<A>	cmd_seq	M_CBST_Q01
<B>	cmd_len	LM_CBST_Q0
<B>	cmd_seq	M_CBST_Q03
<C>	cmd_len	LM_CBST_Q0
<C>	cmd_seq	M_CBST_Q05
<D>	cmd_len	LM_CBST_Q0
<D>	cmd_seq	M_CBST_Q07
<E>	cmd_len	LM_CBST_Q0
<E>	cmd_seq	M_CBST_Q09
<F>	cmd_len	LM_CBST_Q0
<F>	cmd_seq	M_CBST_Q11
<G>	cmd_len	LM_CBST_Q0
<G>	cmd_seq	M_CBST_Q13
<H>	cmd_len	LM_CBST_Q1
<H>	cmd_seq	M_CBST_Q15
<I>	cmd_len	LM_CBST_Q1
<I>	cmd_seq	M_CBST_Q17
<J>	cmd_len	LM_CBST_Q1
<J>	cmd_seq	M_CBST_Q19
<K>	cmd_len	LM_CBST_Q1
<K>	cmd_seq	M_CBST_Q21
<L>	cmd_len	LM_CBST_Q1
<L>	cmd_seq	M_CBST_Q23
<M>	cmd_len	LM_CBST_Q1
<M>	cmd_seq	M_CBST_Q25
<N>	cmd_len	LM_CBST_Q1
<N>	cmd_seq	M_CBST_Q27
<O>	cmd_len	LM_CBST_Q1
<O>	cmd_seq	M_CBST_Q29
<P>	cmd_len	LM_CBST_Q1
<P>	cmd_seq	M_CBST_Q31
<Q>	cmd_len	LM_CBST_Q1
<Q>	cmd_seq	M_CBST_Q33
<R>	cmd_len	LM_CBST_Q1
<R>	cmd_seq	M_CBST_Q35
<S>	cmd_len	LM_CBST_Q1
<S>	cmd_seq	M_CBST_Q37
<T>	cmd_len	LM_CBST_Q1
<T>	cmd_seq	M_CBST_Q39

(6) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.10.99 DAK Initial
07.01.2003 FK Change of primitive MNCC_CONFIGURE_REQ

4.7.4 ACICC064: setting bcap mode, and test settings – PART II

Description:

Select Bearer Service Type, setting types and test whether type was set

Preamble:

ACICC002

Variants: <A>....<T>

```
    APL                              ACI                              PS
     |                                |                                |
(1)  |          ACI_CMD_REQ           |                                |
     |      (cmd: +CBST x,x,x)        |                                |
     *==============================> *                                |
(2)  |                                |      MNCC_CONFIGURE_REQ        |
     |                                *==============================> *
(3)  |          ACI_CMD_IND           |                                |
     |           (cmd: OK)            |                                |
     *<============================== *                                |
(4)  |          ACI_CMD_REQ           |                                |
     |         (cmd: +CBST?)          |                                |
     *==============================> *                                |
(5)  |          ACI_CMD_IND           |                                |
     |     (cmd: +CSTA: x,x,x)        |                                |
     *<============================== *                                |
(6)  |          ACI_CMD_IND           |                                |
     |           (cmd: OK)            |                                |
     *<============================== *                                |
     |                                |                                |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_CBST_S0
<A>	cmd_seq	C_CBST_S02
<B>	cmd_len	LC_CBST_S0
<B>	cmd_seq	C_CBST_S04
<C>	cmd_len	LC_CBST_S0
<C>	cmd_seq	C_CBST_S06
<D>	cmd_len	LC_CBST_S0
<D>	cmd_seq	C_CBST_S08

<E>	cmd_len	LC_CBST_S0
<E>	cmd_seq	C_CBST_S10
<F>	cmd_len	LC_CBST_S0
<F>	cmd_seq	C_CBST_S12
<G>	cmd_len	LC_CBST_S0
<G>	cmd_seq	C_CBST_S14
<H>	cmd_len	LC_CBST_S1
<H>	cmd_seq	C_CBST_S16
<I>	cmd_len	LC_CBST_S1
<I>	cmd_seq	C_CBST_S18
<J>	cmd_len	LC_CBST_S1
<J>	cmd_seq	C_CBST_S20
<K>	cmd_len	LC_CBST_S1
<K>	cmd_seq	C_CBST_S22
<L>	cmd_len	LC_CBST_S1
<L>	cmd_seq	C_CBST_S24
<M>	cmd_len	LC_CBST_S1
<M>	cmd_seq	C_CBST_S26
<N>	cmd_len	LC_CBST_S1
<N>	cmd_seq	C_CBST_S28
<O>	cmd_len	LC_CBST_S1
<O>	cmd_seq	C_CBST_S30
<P>	cmd_len	LC_CBST_S1
<P>	cmd_seq	C_CBST_S32
<Q>	cmd_len	LC_CBST_S1
<Q>	cmd_seq	C_CBST_S34
<R>	cmd_len	LC_CBST_S1
<R>	cmd_seq	C_CBST_S36
<S>	cmd_len	LC_CBST_S1
<S>	cmd_seq	C_CBST_S38
<T>	cmd_len	LC_CBST_S1
<T>	cmd_seq	C_CBST_S40

(2) MNCC_CONFIGURE_REQ

	called_party_sub	NOT_USED
<A>	bcpara	S_BS_DAT_NTRA_300
<B>	bcpara	S_BS_DAT_NTRA_1200
<C>	bcpara	S_BS_DAT_NTRA_1200_75
<D>	bcpara	S_BS_DAT_NTRA_2400
<E>	bcpara	S_BS_DAT_NTRA_2400_V26
<F>	bcpara	S_BS_DAT_NTRA_4800
<G>	bcpara	S_BS_DAT_NTRA_9600
<H>	bcpara	S_BS_DAT_NTRA_9600_V34
<I>	bcpara	S_BS_DAT_NTRA_14400_V34
<J>	bcpara	S_BS_DAT_NTRA_1200_TM_NONE
<K>	bcpara	S_BS_DAT_NTRA_2400_V120

<L>	bcpara	S_BS_DAT_NTRA_4800_V120
<M>	bcpara	S_BS_DAT_NTRA_9600_V120
<N>	bcpara	S_BS_DAT_NTRA_14400_V120
<O>	bcpara	S_BS_DAT_NTRA_300_TM_NONE
<P>	bcpara S_BS_DAT_NTRA_1200_TM_NONE	
<Q>	bcpara	S_BS_DAT_NTRA_2400_V120
<R>	bcpara	S_BS_DAT_NTRA_4800_V120
<S>	bcpara	S_BS_DAT_NTRA_9600_V120
<T>	bcpara	S_BS_DAT_NTRA_14400_V120
	sns_mode	SNS_MODE_VOICE
	ctm_ena	CTM_DISABLED
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(4) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CBST_Q
	cmd_seq	C_CBST_Q
(5) ACI_CMD_IND		
<A>	cmd_len	LM_CBST_Q0
<A>	cmd_seq	M_CBST_Q02
<B>	cmd_len	LM_CBST_Q0
<B>	cmd_seq	M_CBST_Q04
<C>	cmd_len	LM_CBST_Q0
<C>	cmd_seq	M_CBST_Q06
<D>	cmd_len	LM_CBST_Q0
<D>	cmd_seq	M_CBST_Q08
<E>	cmd_len	LM_CBST_Q0
<E>	cmd_seq	M_CBST_Q10
<F>	cmd_len	LM_CBST_Q0
<F>	cmd_seq	M_CBST_Q12
<G>	cmd_len	LM_CBST_Q0
<G>	cmd_seq	M_CBST_Q14
<H>	cmd_len	LM_CBST_Q1
<H>	cmd_seq	M_CBST_Q16
<I>	cmd_len	LM_CBST_Q1
<I>	cmd_seq	M_CBST_Q18
<J>	cmd_len	LM_CBST_Q1
<J>	cmd_seq	M_CBST_Q20
<K>	cmd_len	LM_CBST_Q1
<K>	cmd_seq	M_CBST_Q22
<L>	cmd_len	LM_CBST_Q1
<L>	cmd_seq	M_CBST_Q24
<M>	cmd_len	LM_CBST_Q1

<M>	cmd_seq	M_CBST_Q26
<N>	cmd_len	LM_CBST_Q1
<N>	cmd_seq	M_CBST_Q28
<O>	cmd_len	LM_CBST_Q1
<O>	cmd_seq	M_CBST_Q30
<P>	cmd_len	LM_CBST_Q1
<P>	cmd_seq	M_CBST_Q32
<Q>	cmd_len	LM_CBST_Q1
<Q>	cmd_seq	M_CBST_Q34
<R>	cmd_len	LM_CBST_Q1
<R>	cmd_seq	M_CBST_Q36
<S>	cmd_len	LM_CBST_Q1
<S>	cmd_seq	M_CBST_Q38
<T>	cmd_len	LM_CBST_Q1
<T>	cmd_seq	M_CBST_Q40
(6) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.10.99 DAK Initial
07.01.2003 FK Change of primitive MNCC_CONFIGURE_REQ

4.7.5 ACICC065: trying to set an illegal service type

Description:

Select Bearer Service Type, test illegal settings

Preamble:

ACICC002

```
     APL                           ACI                           PS
      |                             |                             |
(1)   |         ACI_CMD_REQ         |                             |
      |      (cmd: +CBST=0,0,0)     |                             |
      *===========================> *                             |
(2)   |                             |     MNCC_CONFIGURE_REQ      |
      |                             *===========================> *
(3)   |         ACI_CMD_IND         |                             |
      |          (cmd: OK)          |                             |
      *<=========================== *                             |
      |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT

	cmd_len	LC_CBST_S0
	cmd_seq	C_CBST_S99
(2) MNCC_CONFIGURE_REQ		
	called_party_sub	NOT_USED
	bcpara	S_BS_DAT_TRA_AUTO
	sns_mode	SNS_MODE_VOICE
	ctm_ena	CTM_DISABLED
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.10.99 DAK Initial
07.01.2003 FK Change of primitive
MNCC_CONFIGURE_REQ

4.8 Radio link Protocol "+CRLP"(ACICC071 - ACICC080)

4.8.1 ACICC071: getting list supported modes

Description:

Radio Link Protocol, performe test command

Preamble:

ACICC002

```
    APL                         ACI                         PS
     |                           |                           |
(1)  |        ACI_CMD_REQ        |                           |
     |       (cmd: +CRLP=?)      |                           |
     *==========================>*                           |
(2)  |        ACI_CMD_IND        |                           |
     |     (cmd: +CRLP: ...)     |                           |
     *<==========================*                           |
(3)  |        ACI_CMD_IND        |                           |
     |         (cmd: OK)         |                           |
     *<==========================*                           |
     |                           |                           |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CRLP_T
	cmd_seq	C_CRLP_T
(2) ACI_CMD_IND		
	cmd_len	LM_CRLP_T
	cmd_seq	M_CRLP_T

(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.01.2000 DAK Initial

4.8.2 ACICC072: reading initial settings

Description:

Radio Link Protocol, getting initial settings

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |         ACI_CMD_REQ        |                            |
     |        (cmd: +CRLP?)       |                            |
     *==========================>*                            |
(2)  |         ACI_CMD_IND        |                            |
     |      (cmd: +CRLP: ...)     |                            |
     *<==========================*                            |
(3)  |         ACI_CMD_IND        |                            |
     |          (cmd: OK)         |                            |
     *<==========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CRLP_Q
	cmd_seq	C_CRLP_Q
(2) ACI_CMD_IND		
	cmd_len	LM_CRLP_Q0
	cmd_seq	M_CRLP_Q0
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.01.2000 DAK Initial

4.8.3 ACICC073: setting modes and check after changes

Description:

Radio Link Protocol, setting modes and check, whether they are setted

Preamble:

ACICC002

Variants:

<A>...<H>

```
    APL                           ACI                           PS
     |                             |                             |
(1)  |         ACI_CMD_REQ         |                             |
     |       (cmd: +CRLP=...)      |                             |
     *============================>*                             |
(2)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<============================*                             |
(3)  |         ACI_CMD_REQ         |                             |
     |        (cmd: +CRLP?)        |                             |
     *============================>*                             |
(4)  |         ACI_CMD_IND         |                             |
     |      (cmd: +CRLP: ...)      |                             |
     *<============================*                             |
(5)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<============================*                             |
     |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_CRLP_S0
<A>	cmd_seq	C_CRLP_S0
<B>	cmd_len	LC_CRLP_S1
<B>	cmd_seq	C_CRLP_S1
<C>	cmd_len	LC_CRLP_S2
<C>	cmd_seq	C_CRLP_S2
<D>	cmd_len	LC_CRLP_S3
<D>	cmd_seq	C_CRLP_S3
<E>	cmd_len	LC_CRLP_S4
<E>	cmd_seq	C_CRLP_S4
<F>	cmd_len	LC_CRLP_S5
<F>	cmd_seq	C_CRLP_S5
<G>	cmd_len	LC_CRLP_S6
<G>	cmd_seq	C_CRLP_S6
<H>	cmd_len	LC_CRLP_S7
<H>	cmd_seq	C_CRLP_S7

(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(3) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CRLP_Q
	cmd_seq	C_CRLP_Q
(4) ACI_CMD_IND		
<A>	cmd_len	LM_CRLP_Q1
<A>	cmd_seq	M_CRLP_Q1
<B>	cmd_len	LM_CRLP_Q2
<B>	cmd_seq	M_CRLP_Q2
<C>	cmd_len	LM_CRLP_Q3
<C>	cmd_seq	M_CRLP_Q3
<D>	cmd_len	LM_CRLP_Q4
<D>	cmd_seq	M_CRLP_Q4
<E>	cmd_len	LM_CRLP_Q5
<E>	cmd_seq	M_CRLP_Q5
<F>	cmd_len	LM_CRLP_Q6
<F>	cmd_seq	M_CRLP_Q6
<G>	cmd_len	LM_CRLP_Q7
<G>	cmd_seq	M_CRLP_Q7
<H>	cmd_len	LM_CRLP_Q8
<H>	cmd_seq	M_CRLP_Q8
(5) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.01.2000 DAK Initial

4.8.4 ACICC074: trying to set illegal modes

Description:

Radio Link Protocol, trying to set illegal modes

Preamble:

ACICC002

Variants:

<A>...<F>

```
    APL                           ACI                            PS
     |                             |                              |
(1)  |         ACI_CMD_REQ         |                              |
     |      (cmd: +CRLP=...)       |                              |
     *============================>*                              |
(2)  |         ACI_CMD_IND         |                              |
     |          (cmd: OK)          |                              |
     *<============================*                              |
     |                             |                              |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_CRLP_S10
<A>	cmd_seq	C_CRLP_S10
<B>	cmd_len	LC_CRLP_S11
<B>	cmd_seq	C_CRLP_S11
<C>	cmd_len	LC_CRLP_S12
<C>	cmd_seq	C_CRLP_S12
<D>	cmd_len	LC_CRLP_S13
<D>	cmd_seq	C_CRLP_S13
<E>	cmd_len	LC_CRLP_S14
<E>	cmd_seq	C_CRLP_S14
<F>	cmd_len	LC_CRLP_S15
<F>	cmd_seq	C_CRLP_S15
(2) ACI_CMD_IND		
	cmd_len	LM_EXT_ERR_PRM_NOT_ALWD
	cmd_seq	M_EXT_ERR_PRM_NOT_ALWD

History: 06.01.2000 DAK Initial

4.9 Service reporting control "+CR"(ACICC081 - ACICC090)

4.9.1 ACICC081: getting list of supported modes

Description:

Service Report Control, listing of supported modes

Preamble:

ACICC002

```
    APL                         ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |        (cmd: +CR=?)        |                            |
     *===========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |       (cmd: +CR: 0,1)      |                            |
     *<===========================*                            |
(3)  |        ACI_CMD_IND         |                            |
     |          (cmd: OK)         |                            |
     *<===========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CR_T
	cmd_seq	C_CR_T
(2) ACI_CMD_IND		
	cmd_len	LM_CR_T
	cmd_seq	M_CR_T
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.10.99 DAK Initial

4.9.2 ACICC082: testing initial settings

Description:

Service Report Control, testing initial settings

Preamble:

ACICC002

```
    APL                          ACI                           PS
     |                            |                             |
(1)  |        ACI_CMD_REQ         |                             |
     |        (cmd: +CR?)         |                             |
     *===========================>*                             |
(2)  |        ACI_CMD_IND         |                             |
     |       (cmd: +CR: 0)        |                             |
     *<===========================*                             |
(3)  |        ACI_CMD_IND         |                             |
     |         (cmd: OK)          |                             |
     *<===========================*                             |
     |                            |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CR_Q
	cmd_seq	C_CR_Q
(2) ACI_CMD_IND		
	cmd_len	LM_CR_Q
	cmd_seq	M_CR_Q0
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.10.99 DAK Initial

4.9.3 ACICC083: setting modes and check whether done

Description:

Service Report Control, setting types and test whether type was setted

Preamble:

ACICC002

Variants:

<A>....<B>

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |         ACI_CMD_REQ        |                            |
     |      (cmd: +CR (0||1)      |                            |
     *==========================> *                            |
(2)  |         ACI_CMD_IND        |                            |
     |          (cmd: OK)         |                            |
      *<==========================              *                |
(3)  |         ACI_CMD_REQ        |                            |
     |         (cmd: +CR?)        |                            |
     *==========================> *                            |
(4)  |         ACI_CMD_IND        |                            |
     |     (cmd: +CR:(0||1))      |                            |
     *<========================== *                            |
(5)  |         ACI_CMD_IND        |                            |
     |          (cmd: OK)         |                            |
     *<========================== *                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CR_S
<A>	cmd_seq	C_CR_S0
<B>	cmd_seq	C_CR_S1
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(3) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CR_Q
	cmd_seq	C_CR_Q
(4) ACI_CMD_IND		
	cmd_len	LM_CR_Q
<A>	cmd_seq	M_CR_Q0
<B>	cmd_seq	M_CR_Q1
(5) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.10.99 DAK Initial

4.9.4 ACICC084: trying to set illegal modes

Description:

Service Report Control, set illegal mode

Preamble:

ACICC002

```
    APL                          ACI                           PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |        (cmd: +CR=2)        |                            |
     *===========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |        (cmd: ERROR)        |                            |
     *<===========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CR_S
	cmd_seq	C_CR_S9
(2) ACI_CMD_IND		
	cmd_len	LM_CME_ERR_INV_OPP
	cmd_seq	M_CME_ERR_INV_OPP

History: 06.10.99 DAK Initial

4.10 Cellular result codes "+CRC"(ACICC091 - ACICC100)

4.10.1 ACICC091: listing of supported modes

Description:

Cellular Result Codes, listing of supported modes

Preamble:

ACICC002

```
     APL                          ACI                           PS
      |                            |                             |
(1)   |         ACI_CMD_REQ        |                             |
      |        (cmd: +CRC=?)       |                             |
      *==========================>*                              |
(2)   |         ACI_CMD_IND        |                             |
      |      (cmd: +CRC: 0,1)      |                             |
      *<==========================*                              |
(3)   |         ACI_CMD_IND        |                             |
      |         (cmd: OK)          |                             |
      *<==========================*                              |
      |                            |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CRC_T
	cmd_seq	C_CRC_T
(2) ACI_CMD_IND		
	cmd_len	LM_CRC_T
	cmd_seq	M_CRC_T
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.10.99 DAK Initial

4.10.2 ACICC092: checking initial settings

Description:

Cellular Result Codes, testing initial settings

Preamble:

ACICC002

```
    APL                           ACI                            PS
     |                             |                             |
(1)  |         ACI_CMD_REQ         |                             |
     |        (cmd: +CRC?)         |                             |
     *==========================>*                             |
(2)  |         ACI_CMD_IND         |                             |
     |       (cmd: +CRC: 0)        |                             |
     *<==========================*                             |
(3)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<==========================*                             |
     |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CRC_Q
	cmd_seq	C_CRC_Q
(2) ACI_CMD_IND		
	cmd_len	LM_CRC_Q
	cmd_seq	M_CRC_Q0
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.10.99 DAK Initial

4.10.3 ACICC093: setting several modes and check whether setted

Description:

Cellular Result Codes, setting types and test whether type was setted

Preamble:

ACICC002

Variants:

<A>....<B>

```
    APL                           ACI                           PS
     |                             |                             |
(1)  |         ACI_CMD_REQ         |                             |
     |      (cmd: +CRC (0||1)      |                             |
     *===========================>*                             |
(2)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<===========================*                             |
(3)  |         ACI_CMD_REQ         |                             |
     |        (cmd: +CRC?)         |                             |
     *===========================>*                             |
(4)  |         ACI_CMD_IND         |                             |
     |     (cmd: +CRC:(0||1))      |                             |
     *<===========================*                             |
(5)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<===========================*                             |
     |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CRC_S
<A>	cmd_seq	C_CRC_S0
<B>	cmd_seq	C_CRC_S1
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(3) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CRC_Q
	cmd_seq	C_CRC_Q
(4) ACI_CMD_IND		
	cmd_len	LM_CRC_Q
<A>	cmd_seq	M_CRC_Q0
<B>	cmd_seq	M_CRC_Q1
(5) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 06.10.99 DAK Initial

4.10.4 ACICC094: trying to set illegal modes

Description:

Cellular Result Codes, set illegal mode

Preamble:

ACICC002

```
    APL                           ACI                           PS
     |                             |                             |
(1)  |         ACI_CMD_REQ         |                             |
     |        (cmd: +CRC=2)        |                             |
     *===========================>*                              |
(2)  |         ACI_CMD_IND         |                             |
     |         (cmd: ERROR)        |                             |
     *<===========================*                              |
     |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CRC_S
	cmd_seq	C_CRC_S9
(2) ACI_CMD_IND		
	cmd_len	LM_CME_ERR_INV_OPP
	cmd_seq	M_CME_ERR_INV_OPP

History: 06.10.99 DAK Initial

4.11 Closed user group "+CCUG"(ACICC101 - ACICC110)

4.11.1 ACICC101: listing of supported modes

Description:

Closed user group, listing of supported modes

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |       (cmd: +CCUG=?)       |                            |
     *==========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |     (cmd: +CCUG: ...)      |                            |
     *<==========================*                            |
(3)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<==========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CCUG_T
	cmd_seq	C_CCUG_T
(2) ACI_CMD_IND		
	cmd_len	LM_CCUG_T
	cmd_seq	M_CCUG_T
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 07.10.99 DAK Initial

4.11.2 ACICC102: checking initial settings

Description:

Closed user group, testing initial settings

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |         ACI_CMD_REQ        |                            |
     |        (cmd: +CCUG?)       |                            |
     *==========================> *                            |
(2)  |         ACI_CMD_IND        |                            |
     |     (cmd: +CCUG: 0,0,0)    |                            |
     *<========================== *                            |
(3)  |         ACI_CMD_IND        |                            |
     |          (cmd: OK)         |                            |
     *<========================== *                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CCUG_Q
	cmd_seq	C_CCUG_Q
(2) ACI_CMD_IND		
	cmd_len	LM_CCUG_Q0
	cmd_seq	M_CCUG_Q00
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 07.10.99 DAK Initial

4.11.3 ACICC103: setting modes and check changes – Part I

Description:

Closed user group, setting types and test whether type was setted

Preamble:

ACICC002

Variants:

<A>....<V>

```
    APL                           ACI                            PS
     |                             |                             |
(1)  |         ACI_CMD_REQ         |                             |
     |       (cmd: +CCUG=*)        |                             |
     *===========================> *                             |
(2)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<=========================== *                             |
(3)  |         ACI_CMD_REQ         |                             |
     |        (cmd: +CCUG?)        |                             |
     *===========================> *                             |
(4)  |         ACI_CMD_IND         |                             |
     |      (cmd: +CSTA: *)        |                             |
     *<=========================== *                             |
(5)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<=========================== *                             |
     |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_CCUG_S0
<A>	cmd_seq	C_CCUG_S00
<B>	cmd_len	LC_CCUG_S0
<B>	cmd_seq	C_CCUG_S01
<C>	cmd_len	LC_CCUG_S0
<C>	cmd_seq	C_CCUG_S02
<D>	cmd_len	LC_CCUG_S0
<D>	cmd_seq	C_CCUG_S03
<E>	cmd_len	LC_CCUG_S0
<E>	cmd_seq	C_CCUG_S04
<F>	cmd_len	LC_CCUG_S0
<F>	cmd_seq	C_CCUG_S05
<G>	cmd_len	LC_CCUG_S0
<G>	cmd_seq	C_CCUG_S06
<H>	cmd_len	LC_CCUG_S0
<H>	cmd_seq	C_CCUG_S07
<I>	cmd_len	LC_CCUG_S0
<I>	cmd_seq	C_CCUG_S08
<J>	cmd_len	LC_CCUG_S0
<J>	cmd_seq	C_CCUG_S09
<K>	cmd_len	LC_CCUG_S1
<K>	cmd_seq	C_CCUG_S10
<L>	cmd_len	LC_CCUG_S0
<L>	cmd_seq	C_CCUG_S11
<M>	cmd_len	LC_CCUG_S0

<M>	cmd_seq	C_CCUG_S12
<N>	cmd_len	LC_CCUG_S0
<N>	cmd_seq	C_CCUG_S13
<O>	cmd_len	LC_CCUG_S0
<O>	cmd_seq	C_CCUG_S14
<P>	cmd_len	LC_CCUG_S0
<P>	cmd_seq	C_CCUG_S15
<Q>	cmd_len	LC_CCUG_S0
<Q>	cmd_seq	C_CCUG_S16
<R>	cmd_len	LC_CCUG_S0
<R>	cmd_seq	C_CCUG_S17
<S>	cmd_len	LC_CCUG_S0
<S>	cmd_seq	C_CCUG_S18
<T>	cmd_len	LC_CCUG_S0
<T>	cmd_seq	C_CCUG_S19
<U>	cmd_len	LC_CCUG_S0
<U>	cmd_seq	C_CCUG_S20
<V>	cmd_len	LC_CCUG_S1
<V>	cmd_seq	C_CCUG_S21

(2) ACI_CMD_IND

	cmd_len	LM_OK
	cmd_seq	M_OK

(3) ACI_CMD_REQ

	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CCUG_Q
	cmd_seq	C_CCUG_Q

(4) ACI_CMD_IND

<A>	cmd_len	LM_CCUG_Q0
<A>	cmd_seq	M_CCUG_Q00
<B>	cmd_len	LM_CCUG_Q0
<B>	cmd_seq	M_CCUG_Q01
<C>	cmd_len	LM_CCUG_Q0
<C>	cmd_seq	M_CCUG_Q02
<D>	cmd_len	LM_CCUG_Q0
<D>	cmd_seq	M_CCUG_Q03
<E>	cmd_len	LM_CCUG_Q0
<E>	cmd_seq	M_CCUG_Q04
<F>	cmd_len	LM_CCUG_Q0
<F>	cmd_seq	M_CCUG_Q05
<G>	cmd_len	LM_CCUG_Q0
<G>	cmd_seq	M_CCUG_Q06
<H>	cmd_len	LM_CCUG_Q0
<H>	cmd_seq	M_CCUG_Q07
<I>	cmd_len	LM_CCUG_Q0

<I>	cmd_seq	M_CCUG_Q08
<J>	cmd_len	LM_CCUG_Q0
<J>	cmd_seq	M_CCUG_Q09
<K>	cmd_len	LM_CCUG_Q1
<K>	cmd_seq	M_CCUG_Q10
<L>	cmd_len	LM_CCUG_Q0
<L>	cmd_seq	M_CCUG_Q11
<M>	cmd_len	LM_CCUG_Q0
<M>	cmd_seq	M_CCUG_Q12
<N>	cmd_len	LM_CCUG_Q0
<N>	cmd_seq	M_CCUG_Q13
<O>	cmd_len	LM_CCUG_Q0
<O>	cmd_seq	M_CCUG_Q14
<P>	cmd_len	LM_CCUG_Q0
<P>	cmd_seq	M_CCUG_Q15
<Q>	cmd_len	LM_CCUG_Q0
<Q>	cmd_seq	M_CCUG_Q16
<R>	cmd_len	LM_CCUG_Q0
<R>	cmd_seq	M_CCUG_Q17
<S>	cmd_len	LM_CCUG_Q0
<S>	cmd_seq	M_CCUG_Q18
<T>	cmd_len	LM_CCUG_Q0
<T>	cmd_seq	M_CCUG_Q19
<U>	cmd_len	LM_CCUG_Q0
<U>	cmd_seq	M_CCUG_Q20
<V>	cmd_len	LM_CCUG_Q1
<V>	cmd_seq	M_CCUG_Q21

(5)

ACI_CMD_IND

	cmd_len	LM_OK
	cmd_seq	M_OK

History: 07.10.99 DAK Initial

4.11.4 ACICC104: setting modes and check changes - Part II

Description:

Closed user group, setting types and test whether type was setted

Preamble:

ACICC002

Variants:

<A>....<V>

```
    APL                           ACI                           PS
     |                             |                             |
(1)  |         ACI_CMD_REQ         |                             |
     |       (cmd: +CCUG=*)        |                             |
     *===========================>*                              |
(2)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<===========================*                              |
(3)  |         ACI_CMD_REQ         |                             |
     |        (cmd: +CCUG?)        |                             |
     *===========================>*                              |
(4)  |         ACI_CMD_IND         |                             |
     |      (cmd: +CSTA: *)        |                             |
     *<===========================*                              |
(5)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<===========================*                              |
     |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_CCUG_S0
<A>	cmd_seq	C_CCUG_S22
<B>	cmd_len	LC_CCUG_S0
<B>	cmd_seq	C_CCUG_S23
<C>	cmd_len	LC_CCUG_S0
<C>	cmd_seq	C_CCUG_S24
<D>	cmd_len	LC_CCUG_S0
<D>	cmd_seq	C_CCUG_S25
<E>	cmd_len	LC_CCUG_S0
<E>	cmd_seq	C_CCUG_S26
<F>	cmd_len	LC_CCUG_S0
<F>	cmd_seq	C_CCUG_S27
<G>	cmd_len	LC_CCUG_S0
<G>	cmd_seq	C_CCUG_S28
<H>	cmd_len	LC_CCUG_S0
<H>	cmd_seq	C_CCUG_S29
<I>	cmd_len	LC_CCUG_S0
<I>	cmd_seq	C_CCUG_S30
<J>	cmd_len	LC_CCUG_S0
<J>	cmd_seq	C_CCUG_S31
<K>	cmd_len	LC_CCUG_S1
<K>	cmd_seq	C_CCUG_S32
<L>	cmd_len	LC_CCUG_S0
<L>	cmd_seq	C_CCUG_S33
<M>	cmd_len	LC_CCUG_S0

<M>	cmd_seq	C_CCUG_S34
<N>	cmd_len	LC_CCUG_S0
<N>	cmd_seq	C_CCUG_S35
<O>	cmd_len	LC_CCUG_S0
<O>	cmd_seq	C_CCUG_S36
<P>	cmd_len	LC_CCUG_S0
<P>	cmd_seq	C_CCUG_S37
<Q>	cmd_len	LC_CCUG_S0
<Q>	cmd_seq	C_CCUG_S38
<R>	cmd_len	LC_CCUG_S0
<R>	cmd_seq	C_CCUG_S39
<S>	cmd_len	LC_CCUG_S0
<S>	cmd_seq	C_CCUG_S40
<T>	cmd_len	LC_CCUG_S0
<T>	cmd_seq	C_CCUG_S41
<U>	cmd_len	LC_CCUG_S0
<U>	cmd_seq	C_CCUG_S42
<V>	cmd_len	LC_CCUG_S1
<V>	cmd_seq	C_CCUG_S43

(2) ACI_CMD_IND

	cmd_len	LM_OK
	cmd_seq	M_OK

(3) ACI_CMD_REQ

	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CCUG_Q
	cmd_seq	C_CCUG_Q

(4) ACI_CMD_IND

<A>	cmd_len	LM_CCUG_Q0
<A>	cmd_seq	M_CCUG_Q22
<B>	cmd_len	LM_CCUG_Q0
<B>	cmd_seq	M_CCUG_Q23
<C>	cmd_len	LM_CCUG_Q0
<C>	cmd_seq	M_CCUG_Q24
<D>	cmd_len	LM_CCUG_Q0
<D>	cmd_seq	M_CCUG_Q25
<E>	cmd_len	LM_CCUG_Q0
<E>	cmd_seq	M_CCUG_Q26
<F>	cmd_len	LM_CCUG_Q0
<F>	cmd_seq	M_CCUG_Q27
<G>	cmd_len	LM_CCUG_Q0
<G>	cmd_seq	M_CCUG_Q28
<H>	cmd_len	LM_CCUG_Q0
<H>	cmd_seq	M_CCUG_Q29
<I>	cmd_len	LM_CCUG_Q0

<I>	cmd_seq	M_CCUG_Q30
<J>	cmd_len	LM_CCUG_Q0
<J>	cmd_seq	M_CCUG_Q31
<K>	cmd_len	LM_CCUG_Q1
<K>	cmd_seq	M_CCUG_Q32
<L>	cmd_len	LM_CCUG_Q0
<L>	cmd_seq	M_CCUG_Q33
<M>	cmd_len	LM_CCUG_Q0
<M>	cmd_seq	M_CCUG_Q34
<N>	cmd_len	LM_CCUG_Q0
<N>	cmd_seq	M_CCUG_Q35
<O>	cmd_len	LM_CCUG_Q0
<O>	cmd_seq	M_CCUG_Q36
<P>	cmd_len	LM_CCUG_Q0
<P>	cmd_seq	M_CCUG_Q37
<Q>	cmd_len	LM_CCUG_Q0
<Q>	cmd_seq	M_CCUG_Q38
<R>	cmd_len	LM_CCUG_Q0
<R>	cmd_seq	M_CCUG_Q39
<S>	cmd_len	LM_CCUG_Q0
<S>	cmd_seq	M_CCUG_Q40
<T>	cmd_len	LM_CCUG_Q0
<T>	cmd_seq	M_CCUG_Q41
<U>	cmd_len	LM_CCUG_Q0
<U>	cmd_seq	M_CCUG_Q42
<V>	cmd_len	LM_CCUG_Q1
<V>	cmd_seq	M_CCUG_Q43

(5)

ACI_CMD_IND

cmd_len	LM_OK
cmd_seq	M_OK

History: 07.10.99 DAK Initial

4.11.5 ACICC105: setting modes and check changes - Part III

Description:

Closed user group, setting types and test whether type was setted

Preamble:

ACICC002

Variants:

<A>....<V>

```
    APL                            ACI                            PS
     |                              |                              |
(1)  |         ACI_CMD_REQ          |                              |
     |       (cmd: +CCUG=*)         |                              |
     *============================>*                              |
(2)  |         ACI_CMD_IND          |                              |
     |          (cmd: OK)           |                              |
     *<============================*                              |
(3)  |         ACI_CMD_REQ          |                              |
     |        (cmd: +CCUG?)         |                              |
     *============================>*                              |
(4)  |         ACI_CMD_IND          |                              |
     |      (cmd: +CSTA: *)         |                              |
     *<============================*                              |
(5)  |         ACI_CMD_IND          |                              |
     |          (cmd: OK)           |                              |
     *<============================*                              |
     |                              |                              |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_CCUG_S0
<A>	cmd_seq	C_CCUG_S44
<B>	cmd_len	LC_CCUG_S0
<B>	cmd_seq	C_CCUG_S45
<C>	cmd_len	LC_CCUG_S0
<C>	cmd_seq	C_CCUG_S46
<D>	cmd_len	LC_CCUG_S0
<D>	cmd_seq	C_CCUG_S47
<E>	cmd_len	LC_CCUG_S0
<E>	cmd_seq	C_CCUG_S48
<F>	cmd_len	LC_CCUG_S0
<F>	cmd_seq	C_CCUG_S49
<G>	cmd_len	LC_CCUG_S0
<G>	cmd_seq	C_CCUG_S50
<H>	cmd_len	LC_CCUG_S0
<H>	cmd_seq	C_CCUG_S51
<I>	cmd_len	LC_CCUG_S0
<I>	cmd_seq	C_CCUG_S52
<J>	cmd_len	LC_CCUG_S0
<J>	cmd_seq	C_CCUG_S53
<K>	cmd_len	LC_CCUG_S1
<K>	cmd_seq	C_CCUG_S54
<L>	cmd_len	LC_CCUG_S0
<L>	cmd_seq	C_CCUG_S55
<M>	cmd_len	LC_CCUG_S0

<M>	cmd_seq	C_CCUG_S56
<N>	cmd_len	LC_CCUG_S0
<N>	cmd_seq	C_CCUG_S57
<O>	cmd_len	LC_CCUG_S0
<O>	cmd_seq	C_CCUG_S58
<P>	cmd_len	LC_CCUG_S0
<P>	cmd_seq	C_CCUG_S59
<Q>	cmd_len	LC_CCUG_S0
<Q>	cmd_seq	C_CCUG_S60
<R>	cmd_len	LC_CCUG_S0
<R>	cmd_seq	C_CCUG_S61
<S>	cmd_len	LC_CCUG_S0
<S>	cmd_seq	C_CCUG_S62
<T>	cmd_len	LC_CCUG_S0
<T>	cmd_seq	C_CCUG_S63
<U>	cmd_len	LC_CCUG_S0
<U>	cmd_seq	C_CCUG_S64
<V>	cmd_len	LC_CCUG_S1
<V>	cmd_seq	C_CCUG_S65

(2) ACI_CMD_IND

	cmd_len	LM_OK
	cmd_seq	M_OK

(3) ACI_CMD_REQ

	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CCUG_Q
	cmd_seq	C_CCUG_Q

(4) ACI_CMD_IND

<A>	cmd_len	LM_CCUG_Q0
<A>	cmd_seq	M_CCUG_Q44
<B>	cmd_len	LM_CCUG_Q0
<B>	cmd_seq	M_CCUG_Q45
<C>	cmd_len	LM_CCUG_Q0
<C>	cmd_seq	M_CCUG_Q46
<D>	cmd_len	LM_CCUG_Q0
<D>	cmd_seq	M_CCUG_Q47
<E>	cmd_len	LM_CCUG_Q0
<E>	cmd_seq	M_CCUG_Q48
<F>	cmd_len	LM_CCUG_Q0
<F>	cmd_seq	M_CCUG_Q49
<G>	cmd_len	LM_CCUG_Q0
<G>	cmd_seq	M_CCUG_Q50
<H>	cmd_len	LM_CCUG_Q0
<H>	cmd_seq	M_CCUG_Q51
<I>	cmd_len	LM_CCUG_Q0

<I>	cmd_seq	M_CCUG_Q52
<J>	cmd_len	LM_CCUG_Q0
<J>	cmd_seq	M_CCUG_Q53
<K>	cmd_len	LM_CCUG_Q1
<K>	cmd_seq	M_CCUG_Q54
<L>	cmd_len	LM_CCUG_Q0
<L>	cmd_seq	M_CCUG_Q55
<M>	cmd_len	LM_CCUG_Q0
<M>	cmd_seq	M_CCUG_Q56
<N>	cmd_len	LM_CCUG_Q0
<N>	cmd_seq	M_CCUG_Q57
<O>	cmd_len	LM_CCUG_Q0
<O>	cmd_seq	M_CCUG_Q58
<P>	cmd_len	LM_CCUG_Q0
<P>	cmd_seq	M_CCUG_Q59
<Q>	cmd_len	LM_CCUG_Q0
<Q>	cmd_seq	M_CCUG_Q60
<R>	cmd_len	LM_CCUG_Q0
<R>	cmd_seq	M_CCUG_Q61
<S>	cmd_len	LM_CCUG_Q0
<S>	cmd_seq	M_CCUG_Q62
<T>	cmd_len	LM_CCUG_Q0
<T>	cmd_seq	M_CCUG_Q63
<U>	cmd_len	LM_CCUG_Q0
<U>	cmd_seq	M_CCUG_Q64
<V>	cmd_len	LM_CCUG_Q1
<V>	cmd_seq	M_CCUG_Q65

(5)

ACI_CMD_IND

	cmd_len	LM_OK
	cmd_seq	M_OK

History: 07.10.99 DAK Initial

4.11.6 ACICC106: setting modes and check changes – Part IV

Description:

Closed user group, setting types and test whether type was setted

Preamble:

ACICC002

Variants:

<A>....<V>

```
    APL                        ACI                         PS
     |                          |                           |
(1)  |        ACI_CMD_REQ       |                           |
     |      (cmd: +CCUG=*)      |                           |
     *=========================>*                           |
(2)  |        ACI_CMD_IND       |                           |
     |         (cmd: OK)        |                           |
     *<=========================*                           |
(3)  |        ACI_CMD_REQ       |                           |
     |      (cmd: +CCUG?)       |                           |
     *=========================>*                           |
(4)  |        ACI_CMD_IND       |                           |
     |     (cmd: +CSTA: *)      |                           |
     *<=========================*                           |
(5)  |        ACI_CMD_IND       |                           |
     |         (cmd: OK)        |                           |
     *<=========================*                           |
     |                          |                           |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_CCUG_S0
<A>	cmd_seq	C_CCUG_S66
<B>	cmd_len	LC_CCUG_S0
<B>	cmd_seq	C_CCUG_S67
<C>	cmd_len	LC_CCUG_S0
<C>	cmd_seq	C_CCUG_S68
<D>	cmd_len	LC_CCUG_S0
<D>	cmd_seq	C_CCUG_S69
<E>	cmd_len	LC_CCUG_S0
<E>	cmd_seq	C_CCUG_S70
<F>	cmd_len	LC_CCUG_S0
<F>	cmd_seq	C_CCUG_S71
<G>	cmd_len	LC_CCUG_S0
<G>	cmd_seq	C_CCUG_S72
<H>	cmd_len	LC_CCUG_S0
<H>	cmd_seq	C_CCUG_S73
<I>	cmd_len	LC_CCUG_S0
<I>	cmd_seq	C_CCUG_S74
<J>	cmd_len	LC_CCUG_S0
<J>	cmd_seq	C_CCUG_S75
<K>	cmd_len	LC_CCUG_S1
<K>	cmd_seq	C_CCUG_S76
<L>	cmd_len	LC_CCUG_S0
<L>	cmd_seq	C_CCUG_S77
<M>	cmd_len	LC_CCUG_S0

<M>	cmd_seq	C_CCUG_S78
<N>	cmd_len	LC_CCUG_S0
<N>	cmd_seq	C_CCUG_S79
<O>	cmd_len	LC_CCUG_S0
<O>	cmd_seq	C_CCUG_S80
<P>	cmd_len	LC_CCUG_S0
<P>	cmd_seq	C_CCUG_S81
<Q>	cmd_len	LC_CCUG_S0
<Q>	cmd_seq	C_CCUG_S82
<R>	cmd_len	LC_CCUG_S0
<R>	cmd_seq	C_CCUG_S83
<S>	cmd_len	LC_CCUG_S0
<S>	cmd_seq	C_CCUG_S84
<T>	cmd_len	LC_CCUG_S0
<T>	cmd_seq	C_CCUG_S85
<U>	cmd_len	LC_CCUG_S0
<U>	cmd_seq	C_CCUG_S86
<V>	cmd_len	LC_CCUG_S1
<V>	cmd_seq	C_CCUG_S87

(2) ACI_CMD_IND

cmd_len	LM_OK
cmd_seq	M_OK

(3) ACI_CMD_REQ

cmd_src	CMD_SRC_EXT
cmd_len	LC_CCUG_Q
cmd_seq	C_CCUG_Q

(4) ACI_CMD_IND

<A>	cmd_len	LM_CCUG_Q0
<A>	cmd_seq	M_CCUG_Q66
<B>	cmd_len	LM_CCUG_Q0
<B>	cmd_seq	M_CCUG_Q67
<C>	cmd_len	LM_CCUG_Q0
<C>	cmd_seq	M_CCUG_Q68
<D>	cmd_len	LM_CCUG_Q0
<D>	cmd_seq	M_CCUG_Q69
<E>	cmd_len	LM_CCUG_Q0
<E>	cmd_seq	M_CCUG_Q70
<F>	cmd_len	LM_CCUG_Q0
<F>	cmd_seq	M_CCUG_Q71
<G>	cmd_len	LM_CCUG_Q0
<G>	cmd_seq	M_CCUG_Q72
<H>	cmd_len	LM_CCUG_Q0
<H>	cmd_seq	M_CCUG_Q73
<I>	cmd_len	LM_CCUG_Q0

<I>	cmd_seq	M_CCUG_Q74
<J>	cmd_len	LM_CCUG_Q0
<J>	cmd_seq	M_CCUG_Q75
<K>	cmd_len	LM_CCUG_Q1
<K>	cmd_seq	M_CCUG_Q76
<L>	cmd_len	LM_CCUG_Q0
<L>	cmd_seq	M_CCUG_Q77
<M>	cmd_len	LM_CCUG_Q0
<M>	cmd_seq	M_CCUG_Q78
<N>	cmd_len	LM_CCUG_Q0
<N>	cmd_seq	M_CCUG_Q79
<O>	cmd_len	LM_CCUG_Q0
<O>	cmd_seq	M_CCUG_Q80
<P>	cmd_len	LM_CCUG_Q0
<P>	cmd_seq	M_CCUG_Q81
<Q>	cmd_len	LM_CCUG_Q0
<Q>	cmd_seq	M_CCUG_Q82
<R>	cmd_len	LM_CCUG_Q0
<R>	cmd_seq	M_CCUG_Q83
<S>	cmd_len	LM_CCUG_Q0
<S>	cmd_seq	M_CCUG_Q84
<T>	cmd_len	LM_CCUG_Q0
<T>	cmd_seq	M_CCUG_Q85
<U>	cmd_len	LM_CCUG_Q0
<U>	cmd_seq	M_CCUG_Q86
<V>	cmd_len	LM_CCUG_Q1
<V>	cmd_seq	M_CCUG_Q87

(5)

ACI_CMD_IND

	cmd_len	LM_OK
	cmd_seq	M_OK

(6) History: 07.10.99 DAK Initial

4.11.7 ACICC107: trying to set illegal modes

Description:

Closed user group, testing illegal settings

Preamble:

ACICC002

Variants:

<A>....<C>

```
    APL                        ACI                        PS
     |                          |                          |
(1)  |        ACI_CMD_REQ       |                          |
     |       (cmd: +CCUG=*)     |                          |
     *=========================>*                          |
(2)  |        ACI_CMD_IND       |                          |
     |        (cmd: ERROR)      |                          |
     *<=========================*                          |
     |                          |                          |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_CCUG_S0
<A>	cmd_seq	C_CCUG_S97
<B>	cmd_len	LC_CCUG_S1
<B>	cmd_seq	C_CCUG_S98
<C>	cmd_len	LC_CCUG_S0
<C>	cmd_seq	C_CCUG_S99
(2) ACI_CMD_IND		
	cmd_len	LM_EXT_ERR_PRM_NOT_ALWD
	cmd_seq	M_EXT_ERR_PRM_NOT_ALWD

History: 07.10.99 DAK Initial

4.12 Call related supplementary services "+CHLD" (ACICC111 - ACICC120)

4.12.1 ACICC111: listing of supported modes

Description:

call related suupplementary services, listing of supported modes

Preamble:

ACICC002

```
    APL                        ACI                        PS
     |                          |                          |
(1)  |        ACI_CMD_REQ       |                          |
     |       (cmd: +CHLD=?)     |                          |
     *=========================>*                          |
(2)  |        ACI_CMD_IND       |                          |
     |       (cmd: +CHLD: )     |                          |
     *<=========================*                          |
(3)  |        ACI_CMD_IND       |                          |
     |         (cmd: OK)        |                          |
     *<=========================*                          |
     |                          |                          |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CHLD_T
	cmd_seq	C_CHLD_T
(2) ACI_CMD_IND		
	cmd_len	LM_CHLD_T
	cmd_seq	M_CHLD_T
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 11.01.2000 DAK Initial

4.12.2 ACICC112: trying to performe a read command

Description:

call related suupplementary services, trying to performe a read command

Preamble:

ACICC002

```
    APL                          ACI                           PS
     |                            |                             |
(1)  |        ACI_CMD_REQ         |                             |
     |       (cmd: +CHLD?)        |                             |
     *===========================>*                             |
(2)  |        ACI_CMD_IND         |                             |
     |     (cmd: CME ERROR)       |                             |
     *<===========================*                             |
     |                            |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CHLD_Q
	cmd_seq	C_CHLD_Q
(2) ACI_CMD_IND		
	cmd_len	LM_CME_ERR_INV_OPP
	cmd_seq	M_CME_ERR_INV_OPP

History: 11.01.2000 DAK Initial

4.12.3 ACICC113:

Description:

call related suupplementary services,

Preamble:

ACICC022A

```
     APL                                 ACI                                  PS
      |                                   |                                    |
(1)   |          ACI_CMD_REQ              |                                    |
      |         (cmd: +CHLD=)             |                                    |
      *==================================>*                                    |
(2)   |          ACI_CMD_IND              |                                    |
      |       (cmd: CME ERROR)            |                                    |
      *<==================================*                                    |
      |                                   |                                    |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CHLD_S0
	cmd_seq	C_CHLD_S0
(2) ACI_CMD_IND		
	cmd_len	LM_CME_ERR_INV_OPP
	cmd_seq	M_CME_ERR_INV_OPP

History: 11.01.2000 DAK Initial

4.13 Call deflection "+CTRF" (ACICC121 - ACICC130)

4.13.1 ACICC121: performe test & read command

Description:

Call deflection, performe test & read command

Preamble:

ACICC002

Variants:

<A>...<B>

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |       (cmd: +CTFR=?)       |                            |
     *===========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |    (cmd: OK, CME ERROR)    |                            |
     *<===========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_CTFR_T
<A>	cmd_seq	C_CTFR_T
<B>	cmd_len	LC_CTFR_Q
<B>	cmd_seq	C_CTFR_Q
(2) ACI_CMD_IND		
<A>	cmd_len	LM_OK
<A>	cmd_seq	M_OK
<B>	cmd_len	LM_CME_ERR_INV_OPP
<B>	cmd_seq	M_CME_ERR_INV_OPP

History: 13.01.2000 DAK Initial

4.14 Advice of charge "+CAOC"(ACICC131 - ACICC140)

4.14.1 ACICC131: getting list of supported modes

Description:

Advisce of charge, listing of supported modes

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |       (cmd: +CAOC=?)       |                            |
     *===========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |     (cmd: +CAOC: ...)      |                            |
     *<===========================*                            |
(3)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<===========================*                            |
     |                            |                            |
```

Parametrization:

	Primitive	Parameter	Value
(1)	ACI_CMD_REQ		
		cmd_src	CMD_SRC_EXT
		cmd_len	LC_CAOC_T
		cmd_seq	C_CAOC_T
(2)	ACI_CMD_IND		
		cmd_len	LM_CAOC_T
		cmd_seq	M_CAOC_T
(3)	ACI_CMD_IND		
		cmd_len	LM_OK
		cmd_seq	M_OK

History:	07.10.99	DAK	Initial

4.14.2 ACICC132: checking initial settings

Description:

Advisce of charge, testing initial settings

Preamble:

ACICC002

```
    APL                         ACI                         PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |       (cmd: +CAOC?)        |                            |
     *===========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |      (cmd: +CAOC: 2)       |                            |
     *<===========================*                            |
(3)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<===========================*                            |
     |                            |                            |
```

Parametrization:

	Primitive	Parameter	Value
(1)	ACI_CMD_REQ		
		cmd_src	CMD_SRC_EXT
		cmd_len	LC_CAOC_Q
		cmd_seq	C_CAOC_Q
(2)	ACI_CMD_IND		
		cmd_len	LM_CAOC_Q
		cmd_seq	M_CAOC_Q1

(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History:	07.10.99	DAK	Initial

4.14.3 ACICC133: setting modes and test whether they are changed

Description:

Advice of Charge, setting call mode and test whether mode was setted

Preamble:

ACICC002

Variants:

<A>....<B>

```
    APL                          ACI                           PS
     |                            |                             |
(1)  |        ACI_CMD_REQ         |                             |
     |       (cmd: +CAOC=*)       |                             |
     *==========================>*                              |
(2)  |        ACI_CMD_IND         |                             |
     |         (cmd: OK)          |                             |
     *<==========================*                              |
(3)  |        ACI_CMD_REQ         |                             |
     |       (cmd: +CAOC?)        |                             |
     *==========================>*                              |
(4)  |        ACI_CMD_IND         |                             |
     |      (cmd: +CAOC: *)       |                             |
     *<==========================*                              |
(5)  |        ACI_CMD_IND         |                             |
     |         (cmd: OK)          |                             |
     *<==========================*                              |
     |                            |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CAOC_S
<A>	cmd_seq	C_CAOC_S1
<B>	cmd_seq	C_CAOC_S2
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(3) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT

	cmd_len	LC_CAOC_Q
	cmd_seq	C_CAOC_Q
(4) ACI_CMD_IND		
	cmd_len	LM_CAOC_Q
<A>	cmd_seq	M_CAOC_Q1
<B>	cmd_seq	M_CAOC_Q2
(5) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 05.10.99 DAK Initial

4.14.4 ACICC134: trying to set illegal modes

Description:

Advisce of charge, test illegal settings

Preamble:

ACICC002

```
    APL                        ACI                        PS
     |                          |                          |
(1)  |        ACI_CMD_REQ       |                          |
     |       (cmd: +CAOC=4)     |                          |
     *=========================>*                          |
(2)  |        ACI_CMD_IND       |                          |
     |        (cmd: ERROR)      |                          |
     *<=========================*                          |
     |                          |                          |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CAOC_S
	cmd_seq	C_CAOC_S9
(2) ACI_CMD_IND		
	cmd_len	LM_CME_ERR_INV_OPP
	cmd_seq	M_CME_ERR_INV_OPP

History: 07.10.99 DAK Initial

4.14.5 ACICC135: setting modes and test whether they are changed

Description:

Advice of Charge, setting call mode and test whether mode was setted

Preamble:

ACICC002

```
        APL                            ACI                            PS
         |                              |                              |
(1)      |         ACI_CMD_REQ          |                              |
         |       (cmd: +CAOC=*)         |                              |
         *==========================>*                                 |
(2)      |         ACI_CMD_IND          |                              |
         |      (cmd: +CAOC: *)         |                              |
         *<==========================*                                 |
(3)      |         ACI_CMD_IND          |                              |
         |          (cmd: OK)           |                              |
         *<==========================*                                 |
(4)      |         ACI_CMD_REQ          |                              |
         |        (cmd: +CAOC?)         |                              |
         *==========================>*                                 |
(5)      |         ACI_CMD_IND          |                              |
         |      (cmd: +CAOC: *)         |                              |
         *<==========================*                                 |
(6)      |         ACI_CMD_IND          |                              |
         |          (cmd: OK)           |                              |
         *<==========================*                                 |
         |                              |                              |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CAOC_S
	cmd_seq	C_CAOC_S0
(2) ACI_CMD_IND		
	cmd_len	LM_CCM
	cmd_seq	M_CCM
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(4) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CAOC_Q
	cmd_seq	C_CAOC_Q

(5) ACI_CMD_IND		
	cmd_len	LM_CAOC_Q
	cmd_seq	M_CAOC_Q1
(6) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 05.10.99 DAK Initial

4.15 Accumulated call meter "+CACM"(ACICC141 - ACICC150)

4.15.1 ACICC141: getting list of supported modes

Description:

accumulated call meter, listing of supported modes

Preamble:

ACICC002

```
    APL                         ACI                         PS
     |                           |                           |
(1)  |        ACI_CMD_REQ        |                           |
     |       (cmd: +CACM=?)      |                           |
     *==========================>*                           |
(2)  |        ACI_CMD_IND        |                           |
     |         (cmd: OK)         |                           |
     *<==========================*                           |
     |                           |                           |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CACM_T
	cmd_seq	C_CACM_T
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 12.10.99 DAK Initial

4.15.2 ACICC142: test initial settings

Description:

accumulated call meter, test of initial settings

Preamble:

ACICC002

```
    APL                          ACI                           PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |       (cmd: +CACM?)        |                            |
     *==========================>*                             |
(2)  |        ACI_CMD_IND         |                            |
     |      (cmd: +CACM: 0)       |                            |
     *<==========================*                             |
(3)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<==========================*                             |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CACM_Q
	cmd_seq	C_CACM_Q
(2) ACI_CMD_IND		
	cmd_len	LM_CACM_Q
	cmd_seq	M_CACM_Q
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 12.10.99 DAK Initial

4.15.3 ACICC143: performe a set command

Description:

accumulated call meter, setting

Preamble:

ACICC002

```
    APL                          ACI                          PS
      |                           |                            |
(1)   |        ACI_CMD_REQ        |                            |
      |  (cmd: +CACM=\"12345\")   |                            |
      *==========================>*                            |
(2)   |                           |   SIM_UPDATE_RECORD_REQ    |
      |                           *===========================>*
(3)   |                           |   SIM_UPDATE_RECORD_CNF    |
      |                           *<===========================*
(4)   |        ACI_CMD_IND        |                            |
      |         (cmd: OK)         |                            |
      *<==========================*                            |
      |                           |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CACM_S
	cmd_seq	C_CACM_S
(2) SIM_UPDATE_RECORD_REQ		
	source	SRC_MMI
	datafield	SIM_ACM
	record	NUM_0
	length	LDATA_ACM
	linear_data	NOT_USED
(3) SIM_UPDATE_RECORD_CNF		
	datafield	SIM_ACM
	record	NUM_0
	cause	SIM_NO_ERROR
(4) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 12.10.99 DAK Initial

4.16 Accumulated call meter maximum "+CAMM"(ACICC151 - ACICC160)

4.16.1 ACICC151: getting list of supported modes

Description:

accumulated call meter, listing of supported modes

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |       (cmd: +CAMM=?)       |                            |
     *===========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<===========================*                            |
     |                            |                            |
```

Parametrization:

	Primitive	Parameter	Value
(1)	ACI_CMD_REQ		
		cmd_src	CMD_SRC_EXT
		cmd_len	LC_CAMM_T
		cmd_seq	C_CAMM_T
(2)	ACI_CMD_IND		
		cmd_len	LM_OK
		cmd_seq	M_OK

History: 12.10.99 DAK Initial

4.16.2 ACICC152: check initial settings

Description:

accumulated call meter, check initial settings

Preamble:

ACICC002

```
    APL                           ACI                           PS
     |                             |                             |
(1)  |         ACI_CMD_REQ         |                             |
     |        (cmd: +CAMM?)        |                             |
     *===========================>*                             |
(2)  |         ACI_CMD_IND         |                             |
     |      (cmd: +CAMM "0")       |                             |
     *<===========================*                             |
(3)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<===========================*                             |
     |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CAMM_Q
	cmd_seq	C_CAMM_Q
(2) ACI_CMD_IND		
	cmd_len	LM_CAMM_Q
	cmd_seq	M_CAMM_Q
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 12.10.99 DAK Initial

4.17 List current calls "+CLCC"(ACICC161 - ACICC170)

4.17.1 ACICC161: getting list of supported modes

Description:

list current calls, listing of supported modes

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |       (cmd: +CLCC=?)       |                            |
     *==========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<==========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CLCC_T
	cmd_seq	C_CLCC_T
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 12.10.99 DAK Initial

4.17.2 ACICC162: performe read command

Description:

list current calls, performe read command

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |        (cmd: +CLCC)        |                            |
     *==========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<==========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CLCC_Q
	cmd_seq	C_CLCC_Q
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 12.10.99 DAK Initial

4.18 Select tone and pulse dialling "T" & "P"(ACICC171 - ACICC180)

4.18.1 ACICC171: select tone dialing

Description:

select tone dialling

Preamble:

ACICC002

```
    APL                            ACI                            PS
     |                              |                              |
(1)  |          ACI_CMD_REQ         |                              |
     |           (cmd: T)           |                              |
     *=============================>*                              |
(2)  |          ACI_CMD_IND         |                              |
     |          (cmd: OK)           |                              |
     *<=============================*                              |
     |                              |                              |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_T
	cmd_seq	C_T
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 12.10.99 DAK Initial

4.18.2 ACICC172: select pulse dialing

Description:

select pulse dialling

Preamble:

ACICC002

```
    APL                         ACI                          PS
     |                           |                           |
(1)  |        ACI_CMD_REQ        |                           |
     |          (cmd: P)         |                           |
     *==========================>*                           |
(2)  |        ACI_CMD_IND        |                           |
     |         (cmd: OK)         |                           |
     *<==========================*                           |
     |                           |                           |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_P
	cmd_seq	C_P
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History:	12.10.99	DAK	Initial

4.19 Rings before automatic answer "S0"(ACICC181 - ACICC190)

4.19.1 ACICC181: getting list of supported modes

Description:

rings before automatic answer, linsting of supported modes

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |        (cmd: S0=?)         |                            |
     *==========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |     (cmd: +S0: (0,1))      |                            |
     *<==========================*                            |
(3)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<==========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S0_T
	cmd_seq	C_S0_T
(2) ACI_CMD_IND		
	cmd_len	LM_S0_T
	cmd_seq	M_S0_T
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History:	12.10.99	DAK	Initial

4.19.2 ACICC182: checking initial settings

Description:

rings before automatic answer, testing of initial settings

Preamble:

ACICC002

```
    APL                           ACI                           PS
     |                             |                             |
(1)  |         ACI_CMD_REQ         |                             |
     |         (cmd: S0?)          |                             |
     *===========================>*                             |
(2)  |         ACI_CMD_IND         |                             |
     |        (cmd: +S0: 0)        |                             |
     *<===========================*                             |
(3)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<===========================*                             |
     |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S0_Q
	cmd_seq	C_S0_Q
(2) ACI_CMD_IND		
	cmd_len	LM_S0_Q
	cmd_seq	M_S0_Q0
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 12.10.99 DAK Initial

4.19.3 ACICC183: setting rings before auto answer

Description:

rings before automatic answer, setting rings

Preamble:

ACICC002

```
    APL                           ACI                           PS
     |                             |                             |
(1)  |         ACI_CMD_REQ         |                             |
     |         (cmd: S0=2)         |                             |
     *===========================>*                              |
(2)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<===========================*                              |
(3)  |         ACI_CMD_REQ         |                             |
     |          (cmd: S0?)         |                             |
     *===========================>*                              |
(4)  |         ACI_CMD_IND         |                             |
     |          (cmd: 002)         |                             |
     *<===========================*                              |
(5)  |         ACI_CMD_IND         |                             |
     |          (cmd: OK)          |                             |
     *<===========================*                              |
     |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S0_S
	cmd_seq	C_S0_S
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(3) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S0_Q
	cmd_seq	C_S0_Q
(4) ACI_CMD_IND		
	cmd_len	LM_S0_Q
	cmd_seq	M_S0_Q2
(5) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 12.10.99 DAK Initial

4.20 Pause before blind dialing "S6"(ACICC191 - ACICC200)

4.20.1 ACICC191: getting list of supported modes

Description:

Pause before blind dialing, linsting of supported modes

Preamble:

ACICC002

```
    APL                            ACI                             PS
     |                              |                              |
(1)  |         ACI_CMD_REQ          |                              |
     |         (cmd: S6=?)          |                              |
     *=============================>*                              |
(2)  |         ACI_CMD_IND          |                              |
     |     (cmd: +S6: (0-255))      |                              |
     *<=============================*                              |
(3)  |         ACI_CMD_IND          |                              |
     |          (cmd: OK)           |                              |
     *<=============================*                              |
     |                              |                              |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S6_T
	cmd_seq	C_S6_T
(2) ACI_CMD_IND		
	cmd_len	LM_S6_T
	cmd_seq	M_S6_T
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 13.10.99 DAK Initial

4.20.2 ACICC192: reading initial settings

Description:

Pause before blind dialing, test of initial settings

Preamble:

ACICC002

```
    APL                               ACI                                PS
     |                                 |                                  |
(1)  |          ACI_CMD_REQ            |                                  |
     |           (cmd: S6?)            |                                  |
     *=============================>*                                  |
(2)  |          ACI_CMD_IND            |                                  |
     |         (cmd: +S6: 2)           |                                  |
     *<=============================*                                  |
(3)  |          ACI_CMD_IND            |                                  |
     |           (cmd: OK)             |                                  |
     *<=============================*                                  |
     |                                 |                                  |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S6_Q
	cmd_seq	C_S6_Q
(2) ACI_CMD_IND		
	cmd_len	LM_S6_Q0
	cmd_seq	M_S6_Q0
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 13.10.99 DAK Initial

4.20.3 ACICC193: setting several modes

Description:

rings before automatic answer, setting rings and check whether setting was done

Preamble:

ACICC002

Variants:

<A>....<I>

```
     APL                           ACI                           PS
      |                             |                             |
(1)   |         ACI_CMD_REQ         |                             |
      |         (cmd: S6=*)         |                             |
      *===========================> *                             |
(2)   |         ACI_CMD_IND         |                             |
      |          (cmd: OK)          |                             |
      *<=========================== *                             |
(3)   |         ACI_CMD_REQ         |                             |
      |         (cmd: S6?)          |                             |
      *===========================> *                             |
(4)   |         ACI_CMD_IND         |                             |
      |        (cmd: S6: *)         |                             |
      *<=========================== *                             |
(5)   |         ACI_CMD_IND         |                             |
      |          (cmd: OK)          |                             |
      *<=========================== *                             |
      |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_S6_S0
<A>	cmd_seq	C_S6_S0
<B>	cmd_len	LC_S6_S0
<B>	cmd_seq	C_S6_S1
<C>	cmd_len	LC_S6_S0
<C>	cmd_seq	C_S6_S2
<D>	cmd_len	LC_S6_S0
<D>	cmd_seq	C_S6_S3
<E>	cmd_len	LC_S6_S0
<E>	cmd_seq	C_S6_S4
<F>	cmd_len	LC_S6_S0
<F>	cmd_seq	C_S6_S5
<G>	cmd_len	LC_S6_S0
<G>	cmd_seq	C_S6_S6
<H>	cmd_len	LC_S6_S0
<H>	cmd_seq	C_S6_S7
<I>	cmd_len	LC_S6_S1
<I>	cmd_seq	C_S6_S8
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(3) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT

	cmd_len	LC_S6_Q
	cmd_seq	C_S6_Q
(4) ACI_CMD_IND		
<A>	cmd_len	LM_S6_Q0
<A>	cmd_seq	M_S6_Q0
<B>	cmd_len	LM_S6_Q0
<B>	cmd_seq	M_S6_Q1
<C>	cmd_len	LM_S6_Q0
<C>	cmd_seq	M_S6_Q2
<D>	cmd_len	LM_S6_Q0
<D>	cmd_seq	M_S6_Q3
<E>	cmd_len	LM_S6_Q0
<E>	cmd_seq	M_S6_Q4
<F>	cmd_len	LM_S6_Q0
<F>	cmd_seq	M_S6_Q5
<G>	cmd_len	LM_S6_Q0
<G>	cmd_seq	M_S6_Q6
<H>	cmd_len	LM_S6_Q0
<H>	cmd_seq	M_S6_Q7
<I>	cmd_len	LM_S6_Q0
<I>	cmd_seq	M_S6_Q8
(5) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

(6) History: 12.10.99 DAK Initial

4.20.4 ACICC194: trying to set illegal modes

Description:

Pause before blind dialing, test illegal settings

Preamble:

ACICC002

```
    APL                         ACI                          PS
     |                           |                            |
(1)  |        ACI_CMD_REQ        |                            |
     |        (cmd: S6=11)       |                            |
     *==========================>*                            |
(2)  |        ACI_CMD_IND        |                            |
     |     (cmd: CME ERROR)      |                            |
     *<==========================*                            |
     |                           |                            |
```

Parametrization:

Primitive	Parameter	Value

(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S6_S1
	cmd_seq	C_S6_S9
(2) ACI_CMD_IND		
	cmd_len	LM_ERROR
	cmd_seq	M_ERROR

History: 13.10.99 DAK Initial

4.21 Wait for completion "S7"(ACICC201 - ACICC210)

4.21.1 ACICC201: getting list of supported modes

Description:

Wait for completion, linsting of supported modes

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |        (cmd: S7=?)         |                            |
     *==========================> *                            |
(2)  |        ACI_CMD_IND         |                            |
     |    (cmd: +S7: (0-255))     |                            |
     *<===========================*                            |
(3)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<===========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S7_T
	cmd_seq	C_S7_T
(2) ACI_CMD_IND		
	cmd_len	LM_S7_T
	cmd_seq	M_S7_T
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 13.10.99 DAK Initial

4.21.2 ACICC202: DOES NOT PASS !!! Bug to be fixed... Reading initial settings

Description:

Wait for completion, test of initial settings

Preamble:

ACICC002

```
    APL                           ACI                           PS
     |                             |                             |
(1)  |         ACI_CMD_REQ         |                             |
     |         (cmd: S7?)          |                             |
     *===========================> *                             |
(2)  |         ACI_CMD_IND         |                             |
     |       (cmd: +S7: 1)         |                             |
     *<=========================== *                             |
(3)  |         ACI_CMD_IND         |                             |
     |         (cmd: OK)           |                             |
     *<=========================== *                             |
     |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S7_Q
	cmd_seq	C_S7_Q
(2) ACI_CMD_IND		
	cmd_len	LM_S7_Q
	cmd_seq	M_S7_Q0
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History:	13.10.99	DAK	Initial

4.21.3 ACICC203: setting modes and check settings

Description:

Wait for completion, setting values and test whether they are setted

Preamble:

ACICC002

Variants:

<A>....<E>

```
    APL                          ACI                           PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |        (cmd: S7=*)         |                            |
     *==========================>*                             |
(2)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<==========================*                             |
(3)  |        ACI_CMD_REQ         |                            |
     |         (cmd: S7?)         |                            |
     *==========================>*                             |
(4)  |        ACI_CMD_IND         |                            |
     |        (cmd: ^???)         |                            |
     *<==========================*                             |
(5)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<==========================*                             |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_S7_S0
<A>	cmd_seq	C_S7_S0
<B>	cmd_len	LC_S7_S1
<B>	cmd_seq	C_S7_S1
<C>	cmd_len	LC_S7_S2
<C>	cmd_seq	C_S7_S2
<D>	cmd_len	LC_S7_S1
<D>	cmd_seq	C_S7_S3
<E>	cmd_len	LC_S7_S1
<E>	cmd_seq	C_S7_S4
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(3) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S7_Q
	cmd_seq	C_S7_Q
(4) ACI_CMD_IND		
	cmd_len	LM_S7_Q
<A>	cmd_seq	M_S7_Q0
<B>	cmd_seq	M_S7_Q1
<C>	cmd_seq	M_S7_Q2
<D>	cmd_seq	M_S7_Q3
<E>	cmd_seq	M_S7_Q4

(5) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History:	20.10.99	DAK	Initial

4.21.4 ACICC204: trying to set illegal modes

Description:

wait for completion, test illegal settings

Preamble:

ACICC002

Variants:

<A>....<B>

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |         ACI_CMD_REQ        |                            |
     |        (cmd: S7=260)       |                            |
     *==========================>*                            |
(2)  |         ACI_CMD_IND        |                            |
     |      (cmd: CME ERROR)      |                            |
     *<==========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_S7_S0
<A>	cmd_seq	C_S7_S8
<B>	cmd_len	LC_S7_S2
<B>	cmd_seq	C_S7_S9
(2) ACI_CMD_IND		
	cmd_len	LM_ERROR
	cmd_seq	M_ERROR

History:	13.10.99	DAK	Initial

4.22 Dial pause "S8"(ACICC211 - ACICC220)

4.22.1 ACICC211: getting list of supported modes

Description:

Dial pause, linsting of supported modes

Preamble:

ACICC002

```
    APL                            ACI                            PS
     |                              |                              |
(1)  |          ACI_CMD_REQ         |                              |
     |          (cmd: S8=?)         |                              |
     *=============================>*                              |
(2)  |          ACI_CMD_IND         |                              |
     |      (cmd: +S8: (1-255))     |                              |
     *<=============================*                              |
(3)  |          ACI_CMD_IND         |                              |
     |           (cmd: OK)          |                              |
     *<=============================*                              |
     |                              |                              |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S8_T
	cmd_seq	C_S8_T
(2) ACI_CMD_IND		
	cmd_len	LM_S8_T
	cmd_seq	M_S8_T
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 13.10.99 DAK Initial

4.22.2 ACICC212: reading initial settings

Description:

dial pause, test of initial settings

Preamble:

ACICC002

```
     APL                            ACI                            PS
      |                              |                              |
(1)   |         ACI_CMD_REQ          |                              |
      |          (cmd: S8?)          |                              |
      *============================>*                              |
(2)   |         ACI_CMD_IND          |                              |
      |        (cmd: +S8: 1)         |                              |
      *<============================*                              |
(3)   |         ACI_CMD_IND          |                              |
      |          (cmd: OK)           |                              |
      *<============================*                              |
      |                              |                              |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S8_Q
	cmd_seq	C_S8_Q
(2) ACI_CMD_IND		
	cmd_len	LM_S8_Q
	cmd_seq	M_S8_Q0
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 20.10.99 DAK Initial

4.22.3 ACICC213: setting modes and check settings

Description:

dial pause, setting values and test whether they are setted

Preamble:

ACICC002

Variants:

<A>....<E>

```
    APL                         ACI                          PS
     |                           |                            |
(1)  |        ACI_CMD_REQ        |                            |
     |        (cmd: S8=*)        |                            |
     *==========================>*                            |
(2)  |        ACI_CMD_IND        |                            |
     |         (cmd: OK)         |                            |
     *<==========================*                            |
(3)  |        ACI_CMD_REQ        |                            |
     |        (cmd: S8?)         |                            |
     *==========================>*                            |
(4)  |        ACI_CMD_IND        |                            |
     |        (cmd: ^???)        |                            |
     *<==========================*                            |
(5)  |        ACI_CMD_IND        |                            |
     |         (cmd: OK)         |                            |
     *<==========================*                            |
     |                           |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_S8_S0
<A>	cmd_seq	C_S8_S0
<B>	cmd_len	LC_S8_S1
<B>	cmd_seq	C_S8_S1
<C>	cmd_len	LC_S8_S2
<C>	cmd_seq	C_S8_S2
<D>	cmd_len	LC_S8_S1
<D>	cmd_seq	C_S8_S3
<E>	cmd_len	LC_S8_S1
<E>	cmd_seq	C_S8_S4
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(3) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S8_Q
	cmd_seq	C_S8_Q
(4) ACI_CMD_IND		
	cmd_len	LM_S8_Q
<A>	cmd_seq	M_S8_Q5
<B>	cmd_seq	M_S8_Q1
<C>	cmd_seq	M_S8_Q2
<D>	cmd_seq	M_S8_Q3
<E>	cmd_seq	M_S8_Q4

(5) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 20.10.99 DAK Initial

4.22.4 ACICC214: trying to set illegal modes

Description:

dial pause, test illegal settings

Preamble:

ACICC002

```
    APL                          ACI                           PS
     |                            |                             |
(1)  |        ACI_CMD_REQ         |                             |
     |       (cmd: S8=256)        |                             |
     *==========================>*                             |
(2)  |        ACI_CMD_IND         |                             |
     |     (cmd: CME ERROR)       |                             |
     *<==========================*                             |
     |                            |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S8_S2
	cmd_seq	C_S8_S9
(2) ACI_CMD_IND		
	cmd_len	LM_ERROR
	cmd_seq	M_ERROR

History: 20.10.99 DAK Initial

4.23 Hang up delay "S10"(ACICC221 - ACICC230)

4.23.1 ACICC221: getting list of supportes modes

Description:

Hang up delay, linsting of supported modes

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |        (cmd: S10=?)        |                            |
     *==========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |    (cmd: +S10: (1-255))    |                            |
     *<==========================*                            |
(3)  |        ACI_CMD_IND         |                            |
     |         (cmd: OK)          |                            |
     *<==========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S10_T
	cmd_seq	C_S10_T
(2) ACI_CMD_IND		
	cmd_len	LM_S10_T
	cmd_seq	M_S10_T
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 20.10.99 DAK Initial

4.23.2 ACICC222: reading initial settings

Description:

Hang up delay, test of initial settings

Preamble:

ACICC002

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |        (cmd: S10?)         |                            |
     *==========================>*                             |
(2)  |        ACI_CMD_IND         |                            |
     |      (cmd: +S10: 1)        |                            |
     *<==========================*                             |
(3)  |        ACI_CMD_IND         |                            |
     |        (cmd: OK)           |                            |
     *<==========================*                             |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S10_Q
	cmd_seq	C_S10_Q
(2) ACI_CMD_IND		
	cmd_len	LM_S10_Q
	cmd_seq	M_S10_Q0
(3) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 20.10.99 DAK Initial

4.23.3 ACICC223: setting modes and check whether done

Description:

Hang up delay, setting values and test whether they are setted

Preamble:

ACICC002

Variants:

<A>....<E>

```
    APL                         ACI                         PS
     |                           |                           |
(1)  |        ACI_CMD_REQ        |                           |
     |       (cmd: S10=*)        |                           |
     *==========================>*                           |
(2)  |        ACI_CMD_IND        |                           |
     |         (cmd: OK)         |                           |
     *<==========================*                           |
(3)  |        ACI_CMD_REQ        |                           |
     |        (cmd: S10?)        |                           |
     *==========================>*                           |
(4)  |        ACI_CMD_IND        |                           |
     |        (cmd: ^???)        |                           |
     *<==========================*                           |
(5)  |        ACI_CMD_IND        |                           |
     |         (cmd: OK)         |                           |
     *<==========================*                           |
     |                           |                           |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_S10_S0
<A>	cmd_seq	C_S10_S0
<B>	cmd_len	LC_S10_S1
<B>	cmd_seq	C_S10_S1
<C>	cmd_len	LC_S10_S2
<C>	cmd_seq	C_S10_S2
<D>	cmd_len	LC_S10_S1
<D>	cmd_seq	C_S10_S3
<E>	cmd_len	LC_S10_S1
<E>	cmd_seq	C_S10_S4
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(3) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_S10_Q
	cmd_seq	C_S10_Q
(4) ACI_CMD_IND		
	cmd_len	LM_S10_Q
<A>	cmd_seq	M_S10_Q0
<B>	cmd_seq	M_S10_Q1
<C>	cmd_seq	M_S10_Q2
<D>	cmd_seq	M_S10_Q3
<E>	cmd_seq	M_S10_Q4

(5) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History:	20.10.99	DAK	Initial

4.23.4 ACICC224: trying to set illegal modes

Description:

Hang up delay, test illegal settings

Preamble:

ACICC002

Variants:

<A>....<B>

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |     (cmd: S10=0||255)      |                            |
     *==========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |      (cmd: CME ERROR)      |                            |
     *<==========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_S10_S0
<A>	cmd_seq	C_S10_S8
<B>	cmd_len	LC_S10_S2
<B>	cmd_seq	C_S10_S9
(2) ACI_CMD_IND		
	cmd_len	LM_ERROR
	cmd_seq	M_ERROR

History:	20.10.99	DAK	Initial

4.24 Expected Error Reporting "+CEER"(ACICC231 - ACICC240)

4.24.1 ACICC231: getting list of supportes modes

Description:

expected error reporting, linsting of supported modes

Preamble:

ACICC002

Variants:

<A>...<B>

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |         ACI_CMD_REQ        |                            |
     |    (cmd: +CEER=?,+CEER?)   |                            |
     *==========================>*                            |
(2)  |         ACI_CMD_IND        |                            |
     |     (cmd: OK,CME ERROR)    |                            |
     *<==========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_CEER_T
<A>	cmd_seq	C_CEER_T
<B>	cmd_len	LC_CEER_Q
<B>	cmd_seq	C_CEER_Q
(2) ACI_CMD_IND		
<A>	cmd_len	LM_OK
<A>	cmd_seq	M_OK
<B>	cmd_len	LM_CME_ERR_INV_OPP
<B>	cmd_seq	M_CME_ERR_INV_OPP

History: 18.01.2000 DAK Initial

4.24.2 ACICC232: reading last error report

Description:

expected error reporting, reading last report

Variants:

<A>...<U>

Preamble:

<A>	ACICC002
<B>	ACICC024A
<C>	ACICC024B
<D>	ACICC024C
<E>	ACICC024D
<F>	ACICC024E
<G>	ACICC025A
<H>	ACICC025B
<I>	ACICC025C
<J>	ACICC025D
<K>	ACICC025E
<L>	ACICC025F
<M>	ACICC025G
<N>	ACICC025H
<O>	ACICC242A
<P>	ACICC242B
<Q>	ACICC052A
<R>	ACICC052B
<S>	ACICC252
<T>	ACICC026
<U>	ACICC036

```
    APL                           ACI                           PS
     |                             |                             |
(1)  |        ACI_CMD_REQ          |                             |
     |        (cmd: +CEER)         |                             |
     *==========================>*                             |
(2)  |        ACI_CMD_IND          |                             |
     |   (cmd: +CEER: <errmsg>)    |                             |
     *<==========================*                             |
(3)  |        ACI_CMD_IND          |                             |
     |         (cmd: OK)           |                             |
     *<==========================*                             |
     |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CEER_S
	cmd_seq	C_CEER_S
(2) ACI_CMD_IND		
<A>	cmd_len	LM_CEER_NO_ERR
<A>	cmd_seq	M_CEER_NO_ERR
<B>	cmd_len	LM_CEER_TIMER_RECOVERY
<B>	cmd_seq	M_CEER_TIMER_RECOVERY

<C>	cmd_len	LM_CEER_INCOMP_DEST
<C>	cmd_seq	M_CEER_INCOMP_DEST
<D>	cmd_len	LM_CEER_AUTH_REJ
<D>	cmd_seq	M_CEER_AUTH_REJ
<E>	cmd_len	LM_CEER_UNSPEC
<E>	cmd_seq	M_CEER_UNSPEC
<F>	cmd_len	LM_CEER_NO_ERR
<F>	cmd_seq	M_CEER_NO_ERR
<G>	cmd_len	LM_CEER_USR_BUSY
<G>	cmd_seq	M_CEER_USR_BUSY
<H>	cmd_len	LM_CEER_ALRT_NO_ANSW
<H>	cmd_seq	M_CEER_ALRT_NO_ANSW
<I>	cmd_len	LM_CEER_UNASSIGNED
<I>	cmd_seq	M_CEER_UNASSIGNED
<J>	cmd_len	LM_CEER_NO_ROUTE
<J>	cmd_seq	M_CEER_NO_ROUTE
<K>	cmd_len	LM_CEER_NO_USR_RESP
<K>	cmd_seq	M_CEER_NO_USR_RESP
<L>	cmd_len	LM_CEER_DEST_OOO
<L>	cmd_seq	M_CEER_DEST_OOO
<M>	cmd_len	LM_CEER_INV_FORMAT
<M>	cmd_seq	M_CEER_INV_FORMAT
<N>	cmd_len	LM_CEER_NO_ERR
<N>	cmd_seq	M_CEER_NO_ERR
<O>	cmd_len	LM_CEER_NO_ERR
<O>	cmd_seq	M_CEER_NO_ERR
<P>	cmd_len	LM_CEER_TIMER_RECOVERY
<P>	cmd_seq	M_CEER_TIMER_RECOVERY
<Q>	cmd_len	LM_CEER_CALL_CLEAR
<Q>	cmd_seq	M_CEER_CALL_CLEAR
<R>	cmd_len	LM_CEER_TIMER_RECOVERY
<R>	cmd_seq	M_CEER_TIMER_RECOVERY
<S>	cmd_len	LM_CEER_CALL_CLEAR
<S>	cmd_seq	M_CEER_CALL_CLEAR
<T>	cmd_len	LM_CEER_ALRT_NO_ANSW
<T>	cmd_seq	M_CEER_ALRT_NO_ANSW
<U>	cmd_len	LM_CEER_CALL_CLEAR
<U>	cmd_seq	M_CEER_CALL_CLEAR

(3) ACI_CMD_IND

	cmd_len	LM_OK
	cmd_seq	M_OK

History:	18.01.2000	DAK	Initial

4.25 Answer a call "A"(ACICC241 - ACICC250)

4.25.1 ACICC241: preamble for further testcases – mt voice call indicated –> ring

Description:

answer a call, preamble for further testcases – mt voice call indicated

Preamble:

ACICC002

Variants:

<A>...<B>

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |         ACI_CMD_REQ        |                            |
     |   (msg: +CLIP=1;+CRC=1")   |                            |
     *==========================> *                            |
(2)  |         ACI_CMD_IND        |                            |
     |         (msg: "OK")        |                            |
     *<========================== *                            |
(3)  |                            |       MNCC_SETUP_IND       |
     |                            *<========================== *
(4)  |         ACI_CMD_IND        |                            |
     |   (msg: "CRING: VOICE")    |                            |
     *<========================== *                            |
(5)  |         ACI_CMD_IND        |                            |
     |  (msg: "CLIR: 01234567")   |                            |
     *<========================== *                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_CLIP_CRC_S
	cmd_seq	C_CLIP_CRC_S
(2) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(3) MNCC_SETUP_IND		
	ti	NUM_8
	ri	RI_NOT_PRES
	bcpara	BC_PARA_SPEECH2
	bcpara2	BC_PARA_NO_SERVICE
	progress_desc	PROG_END_TO_END_PLMN
	sig	SIG_CONF_TONE_ON
<A>	calling_party	CLING_PARTY0

<B>	calling_party	CLING_PARTY1
	calling_party_sub	CLING_PARTY_SUB_NONE
	called_party	CLED_PARTY0
	called_party_sub	CLED_PARTY_SUB_NONE
	redirecting_party	REDIR_PARTY
	redirecting_party_sub	REDIR_PARTY_SUB_NONE
(4) ACI_CMD_IND		
	cmd_len	LM_CRING_VOICE
	cmd_seq	M_CRING_VOICE
(5) ACI_CMD_IND		
<A>	cmd_len	LM_CLIP_Q01
<A>	cmd_seq	M_CLIP_Q01
<B>	cmd_len	LM_CLIP_Q02
<B>	cmd_seq	M_CLIP_Q02

History:	19.01.2000	DAK	Initial
	27.01.2000	DAK	variants added

4.25.2 ACICC242: answer an mt voice call

Description:

answer a call, answer an incomming voice call

Preamble:

ACICC241A

Variants:

<A>...<B>

```
      APL                          ACI                          PS
       |                            |                            |
(1)    |                            |       MNCC_ALERT_REQ       |
       |                            *==========================> *
(2)    |         ACI_CMD_REQ        |                            |
       |         (cmd: "A")         |                            |
       *==========================> *                            |
(3)    |                            |       MNCC_SETUP_RES       |
       |                            *==========================> *
(4)    |                            |        MNCC_SYNC_IND       |
       |                            *<========================== *
(5)    |                            |    MNCC_SETUP_COMPL_IND    |
       |                            *<========================== *
(6)    |                            |        SIM_SYNC_REQ        |
       |                            *==========================> *
(7)    |         ACI_CMD_IND        |                            |
       |        (cmd: "OK")         |                            |
       *<========================== *                            |
       |                            |                            |
```

Parametrization:

Primitive	Parameter	Value

(1) MNCC_ALERT_REQ		
	ti	NUM_8
(2) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_A
	cmd_seq	C_A
(3) MNCC_SETUP_RES		
	ti	NUM_8
(4) MNCC_SYNC_IND		
	ti	NUM_8
	cause	MNCC_CAUSE_UNASSIGN
	chm	CHM_VOICE
(5) MNCC_SETUP_COMPL_IND		
	ti	NUM_8
<A>	cause	MNCC_CAUSE_SUCCESS
<B>	cause	MNCC_CAUSE_MS_TIMER
(6) SIM_SYNC_REQ		
	synccs	SYNC_START_CALL
(7) ACI_CMD_IND		
<A>	cmd_len	LM_OK
<B>	cmd_len	LM_NO_CARRIER
<A>	cmd_seq	M_OK
<B>	cmd_seq	M_NO_CARRIER

History: 19.01.2000 DAK Initial

4.25.3 ACICC243: trying to performe test and readcommand

Description:

answer a call, performe test and read command

Preamble:

ACICC002

Variants:

<A>...<B>

```
    APL                            ACI                            PS
     |                              |                              |
(1)  |          ACI_CMD_REQ         |                              |
     |        (cmd: "A=?,A?")       |                              |
     *============================>*                              |
(2)  |          ACI_CMD_IND         |                              |
     |        (cmd: "ERROR")        |                              |
     *<============================*                              |
     |                              |                              |
```

Parametrization:

Primitive	Parameter	Value

(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_A_T
<A>	cmd_seq	C_A_T
<B>	cmd_len	LC_A_Q
<B>	cmd_seq	C_A_Q
(2) ACI_CMD_IND		
	cmd_len	LM_ERROR
	cmd_seq	M_ERROR

History: 25.01.2000 DAK Initial

4.25.4 ACICC244: Invalid Answer Command

Description:

answer a call, perform test and read command

Preamble:

ACICC002

Variants:

<A>...<B>

```
     APL                           ACI                           PS
      |                             |                             |
(1)   |          ACI_CMD_REQ        |                             |
      |        (cmd: "A=?,A?")      |                             |
      *===========================>*                             |
(2)   |          ACI_CMD_IND        |                             |
      |         (cmd: "ERROR")      |                             |
      *<===========================*                             |
      |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_A_T
<A>	cmd_seq	C_A_T
<B>	cmd_len	LC_A_Q
<B>	cmd_seq	C_A_Q
(2) ACI_CMD_IND		
	cmd_len	LM_ERROR
	cmd_seq	M_ERROR

History: 25.01.2000 DAK Initial

4.26 Hook control "H"(ACICC251 - ACICC260)

4.26.1 ACICC251: trying to performe test and read command

Description:

hook control, performe test and read command

Preamble:

ACICC002

Variants:

<A>...<B>

```
    APL                          ACI                          PS
     |                            |                            |
(1)  |        ACI_CMD_REQ         |                            |
     |      (cmd: "H=?,H?")       |                            |
     *===========================>*                            |
(2)  |        ACI_CMD_IND         |                            |
     |      (cmd: "ERROR")        |                            |
     *<===========================*                            |
     |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_H_T
<A>	cmd_seq	C_H_T
<B>	cmd_len	LC_H_Q
<B>	cmd_seq	C_H_Q
(2) ACI_CMD_IND		
	cmd_len	LM_ERROR
	cmd_seq	M_ERROR

History: 25.01.2000 DAK Initial

4.26.2 ACICC252: hang up a mt voice call

Description:

hook control, hang up a mt voice call

Preamble:

ACICC242A

```
        APL                        ACI                        PS
         |                          |                          |
(1)      |        ACI_CMD_REQ       |                          |
         |        (cmd: "H")        |                          |
         *=========================>*                          |
(2)      |                          |       SIM_SYNC_REQ       |
         |                          *=========================>*
(3)      |                          |    MNCC_DISCONNECT_REQ   |
         |                          *=========================>*
(4)      |                          |       SIM_SYNC_CNF       |
         |                          *<=========================*
(5)      |                          |    MNCC_DISCONNECT_IND   |
         |                          *<=========================*
(6)      |        ACI_CMD_IND       |                          |
         |        (cmd: "OK")       |                          |
         *<=========================*                          |
         |                          |                          |
```

Parametrization:

Primitive	Parameter	Value
(1) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_H
	cmd_seq	C_H
(2) SIM_SYNC_REQ		
	synccs	SYNC_STOP_CALL
(7) MNCC_DISCONNECT_REQ		
	ti	NUM_8
	cause	MNCC_CAUSE_CALL_CLEAR
	fac_inf	NOT_USED
	ss_version	SS_VER_NOT_PRES
(3) SIM_SYNC_CNF		
	cause	NOT_SPEC
(4) MNCC_DISCONNECT_IND		
	ti	NUM_8
	cause	MNCC_CAUSE_CALL_CLEAR
	diagnostic	DIAG_UNKNOWN_CUG_INDEX
	progress_desc	PROG_NO_END_TO_END_PLMN
(5) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History:	25.01.2000	DAK	Initial

4.27 TTY Service (ACICC261 - ACICC270)

4.27.1 ACICC261: TTY Test Command

Description:

Perform TTY Test Command with AT Command %CTTY=?

Preamble:

ACICC002

```
     APL                         ACI                          PS
      |                           |                            |
(7)   |        ACI_CMD_REQ        |                            |
      |       (cmd: %CTTY=?)      |                            |
      *==========================>*                            |
(8)   |        ACI_CMD_IND        |                            |
      |    (res: %CTTY: (...))    |                            |
      *<==========================*                            |
(9)   |        ACI_CMD_IND        |                            |
      |         (res: OK)         |                            |
      *<==========================*                            |
      |                           |                            |
```

Parametrization:

Primitive	Parameter	Value
(7) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_TTY_TEST
	cmd_seq	C_TTY_TEST
(8) ACI_CMD_IND		
	cmd_len	LM_TTY_TEST
	cmd_seq	M_TTY_TEST
(9) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History:	08.01.2003	FK	Initial

4.27.2 ACICC262: Set TTY Service

Description:

Handle TTY Service with AT Command %CTTY=...

Variants:

<A>...<D>

Preamble:

ACICC261

```
        APL                           ACI                           PS
         |                             |                             |
(10)     |         ACI_CMD_REQ         |                             |
         |      (cmd: %CTTY=x,x)       |                             |
         *===========================> *                             |
(11)     |                             |     MNCC_CONFIGURE_REQ      |
         |                             *===========================> *
(12)     |         ACI_CMD_IND         |                             |
         |          (res: OK)          |                             |
         *<=========================== *                             |
         |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(10) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_TTY
<A>	cmd_seq	C_TTY_DIS_NOREQ
<B>	cmd_seq	C_TTY_DIS_REQ
<C>	cmd_seq	C_TTY_EN_NOREQ
<D>	cmd_seq	C_TTY_EN_REQ
(11) MNCC_CONFIGURE_REQ		
	called_party_sub	NOT_USED
	bcpara	NOT_USED
	sns_mode	SNS_MODE_VOICE
<A>	ctm_ena	CTM_DISABLED
<B>	ctm_ena	CTM_ENABLED
<C>	ctm_ena	CTM_DISABLED
<D>	ctm_ena	CTM_ENABLED
(12) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK

History: 08.01.2003 FK Initial

4.27.3 ACICC263: Query TTY Service

Description:

Query TTY Service with AT Command %CTTY?

Variants:

<A>...<N>

Preamble:

<A>	ACICC262A
<B>	ACICC262B
<C>	ACICC262C
<D>	ACICC262D
<E>	ACICC264A
<F>	ACICC264B
<G>	ACICC264C
<H>	ACICC265A
<I>	ACICC265B
<J>	ACICC265C
<K>	ACICC266A
<L>	ACICC266B
<M>	ACICC267A
<N>	ACICC267B

```
      APL                          ACI                          PS
       |                            |                            |
(13)   |        ACI_CMD_REQ         |                            |
       |       (cmd: %CTTY?)        |                            |
       *===========================>*                            |
(14)   |        ACI_CMD_IND         |                            |
       |     (res: %CTTY: ...)      |                            |
       *<===========================*                            |
(15)   |        ACI_CMD_IND         |                            |
       |         (res: OK)          |                            |
       *<===========================*                            |
       |                            |                            |
```

Parametrization:

Primitive	Parameter	Value
(13) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_TTY_QUERY
	cmd_seq	C_TTY_QUERY
(14) ACI_CMD_IND		
	cmd_len	LM_TTY
<A>	cmd_seq	M_TTY_DIS_NOREQ_UNKNOWN
<B>	cmd_seq	M_TTY_DIS_REQ_UNKNOWN
<C>	cmd_seq	M_TTY_EN_NOREQ_UNKNOWN
<D>	cmd_seq	M_TTY_EN_REQ_UNKNOWN

<E>	cmd_seq	M_TTY_DIS_REQ_OFF
<F>	cmd_seq	M_TTY_DIS_REQ_ON
<G>	cmd_seq	M_TTY_DIS_NOREQ_ON
<H>	cmd_seq	M_TTY_EN_REQ_OFF
<I>	cmd_seq	M_TTY_EN_REQ_ON
<J>	cmd_seq	M_TTY_EN_NOREQ_ON
<K>	cmd_seq	M_TTY_DIS_REQ_OFF
<L>	cmd_seq	M_TTY_DIS_REQ_ON
<M>	cmd_seq	M_TTY_EN_REQ_OFF
<N>	cmd_seq	M_TTY_EN_REQ_ON

(15) ACI_CMD_IND

	cmd_len	LM_OK
	cmd_seq	M_OK

History:	09.01.2003	FK	Initial

4.27.4 ACICC264: Setup MO Call with TTY Service Request (no Indication)

Description:

Setup a call with TTY Service request, but without unsolicited status indication.

Variants:

<A>...<C>

Preamble:

<A>	ACICC262B
<B>	ACICC262B
<C>	ACICC262A

```
       APL                           ACI                           PS
        |                             |                             |
(10)    |         ACI_CMD_REQ         |                             |
        |      (cmd: D0123456)        |                             |
        *===========================>*                             |
(11)    |                             |       MNCC_SETUP_REQ        |
        |                             *===========================>*
(12)    |         ACI_CMD_IND         |                             |
        |         (res: OK)           |                             |
        *<===========================*                             |
(13)    |                             |        SIM_SYNC_REQ         |
        |                             *===========================>*
    MUTE(500)
(14)    |                             |    MNCC_CALL_PROCEED_IND    |
        |                             *<===========================*
    MUTE(500)
(15)    |                             |       MNCC_ALERT_IND        |
        |                             *<===========================*
(16)    |                             |        MNCC_SYNC_IND        |
        |                             *<===========================*
    MUTE(500)
(17)    |                             |       MNCC_SETUP_CNF        |
        |                             *<===========================*
(18)    |                             |        SIM_SYNC_CNF         |
        |                             *<===========================*
    MUTE(1000)
        |                             |                             |
```

Parametrization:

Primitive	Parameter	Value
(10) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_D0
<B>	cmd_len	LC_D0
<C>	cmd_len	LC_D_TTY
<A>	cmd_seq	C_D0
<B>	cmd_seq	C_D0
<C>	cmd_seq	C_D_TTY_ON

(11) MNCC_SETUP_REQ

	ti	NUM_0
	prio	PRIO_NORM_CALL
	ri	RI_NOT_PRES
	bcpara	BC_PARA_SPEECH_CTM
	bcpara2	BC_PARA_NO_SERVICE
	called_party	CLED_PARTY0
	called_party_sub	CLED_PARTY_SUB_NONE
	clir_sup	NOT_USED
	fac_inf	NOT_USED

(12) ACI_CMD_IND

	cmd_len	LM_OK
	cmd_seq	M_OK

(13) SIM_SYNC_REQ

	synccs	SYNC_START_CALL

(14) MNCC_CALL_PROCEED_IND

	ti	NUM_0
	progress_desc	NOT_SPEC
	ri	RI_NOT_PRES
<A>	bcpara	BC_PARA_SPEECH
<B>	bcpara	BC_PARA_SPEECH_CTM
<C>	bcpara	BC_PARA_SPEECH_CTM
	bcpara2	BC_PARA_NO_SERVICE

(15) MNCC_ALERT_IND

	ti	NUM_0
	progress_desc	NOT_SPEC

(16) MNCC_SYNC_IND

	ti	NUM_0
	cause	MNCC_CAUSE_CHANNEL_SYNC
	chm	CHM_VOICE

(17) MNCC_SETUP_CNF

	ti	NUM_0
	cause	MNCC_CAUSE_SUCCESS
	progress_desc	NOT_SPEC
	connected_number	CONNECTED_NUMBER0
	connected_number_sub	NOT_USED

(18) SIM_SYNC_CNF

	cause	SIM_NO_ERROR

History:	09.01.2003	FK	Initial

4.27.5 ACICC265: Setup MO Call with TTY Service Request (with Indication)

Description:

Setup a call with TTY Service request and with unsolicited status indication.

Variants:

<A>...<B>

Preamble:

<A>	ACICC262D
<B>	ACICC262D
<C>	ACICC262C

```
      APL                             ACI                            PS
       |                               |                              |
(19)   |          ACI_CMD_REQ          |                              |
       |        (cmd: D0123456)        |                              |
       *===========================>*                              |
(20)   |                               |       MNCC_SETUP_REQ         |
       |                               *===========================>*
(21)   |          ACI_CMD_IND          |                              |
       |      (res: %CTYI: ...)        |                              |
       *<===========================*                              |
(22)   |          ACI_CMD_IND          |                              |
       |           (res: OK)           |                              |
       *<===========================*                              |
(23)   |                               |        SIM_SYNC_REQ          |
       |                               *===========================>*
   MUTE(500)
(24)   |                               |    MNCC_CALL_PROCEED_IND     |
       |                               *<===========================*
(25)   |                               |        SIM_SYNC_CNF          |
       |                               *<===========================*
   MUTE(500)
(26)   |                               |        MNCC_SYNC_IND         |
       |                               *<===========================*
(27)   |          ACI_CMD_IND          |                              |
       |      (res: %CTYI: ...)        |                              |
       *<===========================*                              |
   MUTE(500)
(28)   |                               |       MNCC_ALERT_IND         |
       |                               *<===========================*
(29)   |                               |       MNCC_SETUP_CNF         |
       |                               *<===========================*
   MUTE(1000)
       |                               |                              |
```

Parametrization:

Primitive	Parameter	Value
(19) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
<A>	cmd_len	LC_D0

<B>	cmd_len	LC_D0
<C>	cmd_len	LC_D_TTY
<A>	cmd_seq	C_D0
<B>	cmd_seq	C_D0
<C>	cmd_seq	C_D_TTY_ON
(20) MNCC_SETUP_REQ		
	ti	NUM_0
	prio	PRIO_NORM_CALL
	ri	RI_NOT_PRES
	bcpara	BC_PARA_SPEECH_CTM
	bcpara2	BC_PARA_NO_SERVICE
	called_party	CLED_PARTY0
	called_party_sub	CLED_PARTY_SUB_NONE
	clir_sup	NOT_USED
	fac_inf	NOT_USED
(21) ACI_CMD_IND		
	cmd_len	LM_TYI
	cmd_seq	M_TYI_REQ
(22) ACI_CMD_IND		
	cmd_len	LM_OK
	cmd_seq	M_OK
(23) SIM_SYNC_REQ		
	synccs	SYNC_START_CALL
(24) MNCC_CALL_PROCEED_IND		
	ti	NUM_0
	progress_desc	NOT_SPEC
	ri	RI_NOT_PRES
<A>	bcpara	BC_PARA_SPEECH
<B>	bcpara	BC_PARA_SPEECH_CTM
<C>	bcpara	BC_PARA_SPEECH_CTM
	bcpara2	BC_PARA_NO_SERVICE
(25) SIM_SYNC_CNF		
	cause	SIM_NO_ERROR
(26) MNCC_SYNC_IND		
	ti	NUM_0
	cause	MNCC_CAUSE_CHANNEL_SYNC
	chm	CHM_VOICE
(27) ACI_CMD_IND		
	cmd_len	LM_TYI
<A>	cmd_seq	M_TYI_NOGRANT
<B>	cmd_seq	M_TYI_GRANT
<C>	cmd_seq	M_TYI_GRANT

(28) MNCC_ALERT_IND

	ti	NUM_0
	progress_desc	NOT_SPEC

(29) MNCC_SETUP_CNF

	ti	NUM_0
	cause	MNCC_CAUSE_SUCCESS
	progress_desc	NOT_SPEC
	connected_number	CONNECTED_NUMBER0
	connected_number_sub	NOT_USED

History:	09.01.2003	FK	Initial

4.27.6 ACICC266: Incoming Call with TTY Service Request (no Indication)

Description:

Receive a call, TTY Service requested by MS, but without unsolicited status indication.

Variants:

<A>...<B>

Preamble:

ACICC262B

```
     APL                         ACI                         PS
      |                           |                           |
(8)   |                           |      MNCC_SETUP_IND       |
      |                           *<==========================*
(9)   |                           |      MNCC_ALERT_REQ       |
      |                           *==========================>*
(10)  |        ACI_CMD_IND        |                           |
      |        (res: RING)        |                           |
      *<==========================*                           |
  MUTE(500)
(11)  |        ACI_CMD_REQ        |                           |
      |         (cmd: "A")        |                           |
      *==========================>*                           |
(12)  |                           |      MNCC_SETUP_RES       |
      |                           *==========================>*
(13)  |                           |       SIM_SYNC_REQ        |
      |                           *==========================>*
  MUTE(500)
(14)  |                           |       MNCC_SYNC_IND       |
      |                           *<==========================*
  MUTE(500)
(15)  |                           |       SIM_SYNC_CNF        |
      |                           *<==========================*
  MUTE(500)
(16)  |                           |    MNCC_SETUP_COMPL_IND   |
      |                           *<==========================*
(17)  |        ACI_CMD_IND        |                           |
      |        (cmd: "OK")        |                           |
      *<==========================*                           |
  MUTE(1000)
     |                              |                              |
```

Parametrization:

Primitive	Parameter	Value
(8) MNCC_SETUP_IND		
	ti	NUM_8
	ri	RI_NOT_PRES
<A>	bcpara	BC_PARA_SPEECH
<B>	bcpara	BC_PARA_SPEECH_CTM
	bcpara2	BC_PARA_NO_SERVICE
	progress_desc	PROG_END_TO_END_PLMN
	sig	SIG_CONF_TONE_ON
	calling_party	CLING_PARTY0
	calling_party_sub	CLING_PARTY_SUB_NONE
	called_party	CLED_PARTY0
	called_party_sub	CLED_PARTY_SUB_NONE
	redirecting_party	REDIR_PARTY
	redirecting_party_sub	REDIR_PARTY_SUB_NONE
(9) MNCC_ALERT_REQ		
	ti	NUM_8
(10) ACI_CMD_IND		
	cmd_len	LM_RING
	cmd_seq	M_RING
(11) ACI_CMD_REQ		
	cmd_src	CMD_SRC_EXT
	cmd_len	LC_A
	cmd_seq	C_A
(12) MNCC_SETUP_RES		
	ti	NUM_8
(13) SIM_SYNC_REQ		
	synccs	SYNC_START_CALL
(14) MNCC_SYNC_IND		
	ti	NUM_8
	cause	MNCC_CAUSE_CHANNEL_SYNC
	chm	CHM_VOICE
(15) SIM_SYNC_CNF		
	cause	SIM_NO_ERROR
(16) MNCC_SETUP_COMPL_IND		
	ti	NUM_8
	cause	MNCC_CAUSE_SUCCESS

(17) ACI_CMD_IND			
		cmd_len	LM_OK
		cmd_seq	M_OK
History:	10.01.2003	FK	Initial

4.27.7 ACICC267: Incoming Call with TTY Service Request (with Indication)

Description:

Receive a call, TTY Service requested by MS, with unsolicited status indication.

Variants:

<A>...<B>

Preamble:

ACICC262D

```
    APL                         ACI                          PS
     |                           |                            |
(18) |                           |       MNCC_SETUP_IND       |
     |                           *<===========================*
(19) |                           |       MNCC_ALERT_REQ       |
     |                           *===========================>*
(20) |        ACI_CMD_IND        |                            |
     |        (res: RING)        |                            |
     *<==========================*                            |
  MUTE(500)
(21) |        ACI_CMD_REQ        |                            |
     |         (cmd: "A")        |                            |
     *==========================>*                            |
(22) |                           |       MNCC_SETUP_RES       |
     |                           *===========================>*
(23) |                           |        SIM_SYNC_REQ        |
     |                           *===========================>*
  MUTE(500)
(24) |                           |        SIM_SYNC_CNF        |
     |                           *<===========================*
  MUTE(500)
(25) |                           |        MNCC_SYNC_IND       |
     |                           *<===========================*
  MUTE(500)
(26) |                           |    MNCC_SETUP_COMPL_IND    |
     |                           *<===========================*
(27) |        ACI_CMD_IND        |                            |
     |     (res: %CTYI: ...)     |                            |
     *<==========================*                            |
(28) |        ACI_CMD_IND        |                            |
     |        (cmd: "OK")        |                            |
     *<==========================*                            |
  MUTE(1000)
     |                           |                            |
```

Parametrization:

Primitive	Parameter	Value

(18) MNCC_SETUP_IND

	ti	NUM_8
	ri	RI_NOT_PRES
<A>	bcpara	BC_PARA_SPEECH
<B>	bcpara	BC_PARA_SPEECH_CTM
	bcpara2	BC_PARA_NO_SERVICE
	progress_desc	PROG_END_TO_END_PLMN
	sig	SIG_CONF_TONE_ON
	calling_party	CLING_PARTY0
	calling_party_sub	CLING_PARTY_SUB_NONE
	called_party	CLED_PARTY0
	called_party_sub	CLED_PARTY_SUB_NONE
	redirecting_party	REDIR_PARTY
	redirecting_party_sub	REDIR_PARTY_SUB_NONE

(19) MNCC_ALERT_REQ

ti	NUM_8

(20) ACI_CMD_IND

cmd_len	LM_RING
cmd_seq	M_RING

(21) ACI_CMD_REQ

cmd_src	CMD_SRC_EXT
cmd_len	LC_A
cmd_seq	C_A

(22) MNCC_SETUP_RES

ti	NUM_8

(23) SIM_SYNC_REQ

synccs	SYNC_START_CALL

(24) SIM_SYNC_CNF

cause	SIM_NO_ERROR

(25) MNCC_SYNC_IND

ti	NUM_8
cause	MNCC_CAUSE_CHANNEL_SYNC
chm	CHM_VOICE

(26) MNCC_SETUP_COMPL_IND

ti	NUM_8
cause	MNCC_CAUSE_SUCCESS

(27) ACI_CMD_IND

	cmd_len	LM_TYI
<A>	cmd_seq	M_TYI_NOGRANT
<B>	cmd_seq	M_TYI_GRANT

(28) ACI_CMD_IND

		cmd_len	LM_OK
		cmd_seq	M_OK
History:	13.01.2003	FK	Initial