LLD FDN Feature

Project	TCS 3.x
Document Type	Technical Documentation
Title	LLD FDN Feature
Author	Liyi Yu
Creation Date	11.11.2003
Last Modified	
ID and Version	
Status	Being Processed

0 Document Control

0.1 Document History

ID	Author	Date	Status
	Liyi Yu	02.12.2003	Being processed

0.2 References, Abbreviations, Terms

Ref 1	[TI 7010.801]	7010.801, References and Vocabulary, Texas Instruments.
Ref 2	[TS 02.07]	ETSI TS 100 906, April 2000 (GSM 02.07 version 7.1.0 Release 1998).
Ref 3		GSMA PRD TW.11, Version 3.5.0, September 2002.

0.3 Table of Contents

0.4 Table of Figures

1 Introduction

G23 is a software package implementing Layers 2 and 3 of the ETSI-defined GSM air interface signalling protocol, and as such represents that part of a GSM mobile station's protocol software which is both, platform and manufacturer independent. Therefore, G23 can be viewed as a building block providing standardised functionality through generic interfaces for easy integration.

The G23 suite of products consists of the following items:

- Layers 2 and 3 for speech & short message services,
- Layers 2 and 3 for fax & data services,
- Application Control Interface,
- Slim MMI [02.30] and
- Test and integration support tools.

2 Overview

2.1 General

3GPP TS 02.07 (**Ref 2**) does not clearly define how SMS and GPRS are to be handled in case FDN (fixed dialling numbers) is activated. The implementation chosen in TCS 2.1 (and all previous releases) allows access to SMS and GPRS regardless of the FDN state. This is not in line with operator's expectations and should be modified in such a way that the FDN tests specified in **Ref 3** will be passed.

In the new release TCS 3.x, the SMS and GPRS should be handled in a proper way when FDN is activated. More specific, it means that if FDN is activated, before the sending of an SMS, the destination address will be checked against the FD phonebook. If and only if a match found in the FD phonebook will the SMS be sent. The destination address checking for the SMS includes text mode and PDU mode. Moreover, if FDN is activated, the cell phone is not allowed to connect to the GPRS network if the code "*99#" is not found in the FD phone book. The adding of an entry to FD phonebook is controlled by PIN2.

For the GPRS attachment, the implementation here doesn't care about the GPRS attachment state before the FDN state is changed. The implementation here blocks all possible future attempts to attach the GPRS network.

2.2 Feature List

The new features about the FDN support is listed below:

SMS:

- Destination address will be checked in both text mode and PDU mode if FDN is activated. If a match of the destination number is found in FDN list, the request will be executed. Otherwise the request will be refused and an error message of "operation not allowed" will be returned.
- Destination address will be checked when sending an SMS from storage in both modes if FDN is activated. If a match of the destination number found in FDN list, the request will be performed. Otherwise the request will be refused and an error message of "operation not allowed" will be returned.

GPRS:

- The FDN feature will be checked when the user is attempting to attach the GPRS network. If the FDN feature is on and no entry as "*99#" is found in FD phonebook, the attachment will fail and an error message of "operation not allowed" will be returned.

2.3 Architecture

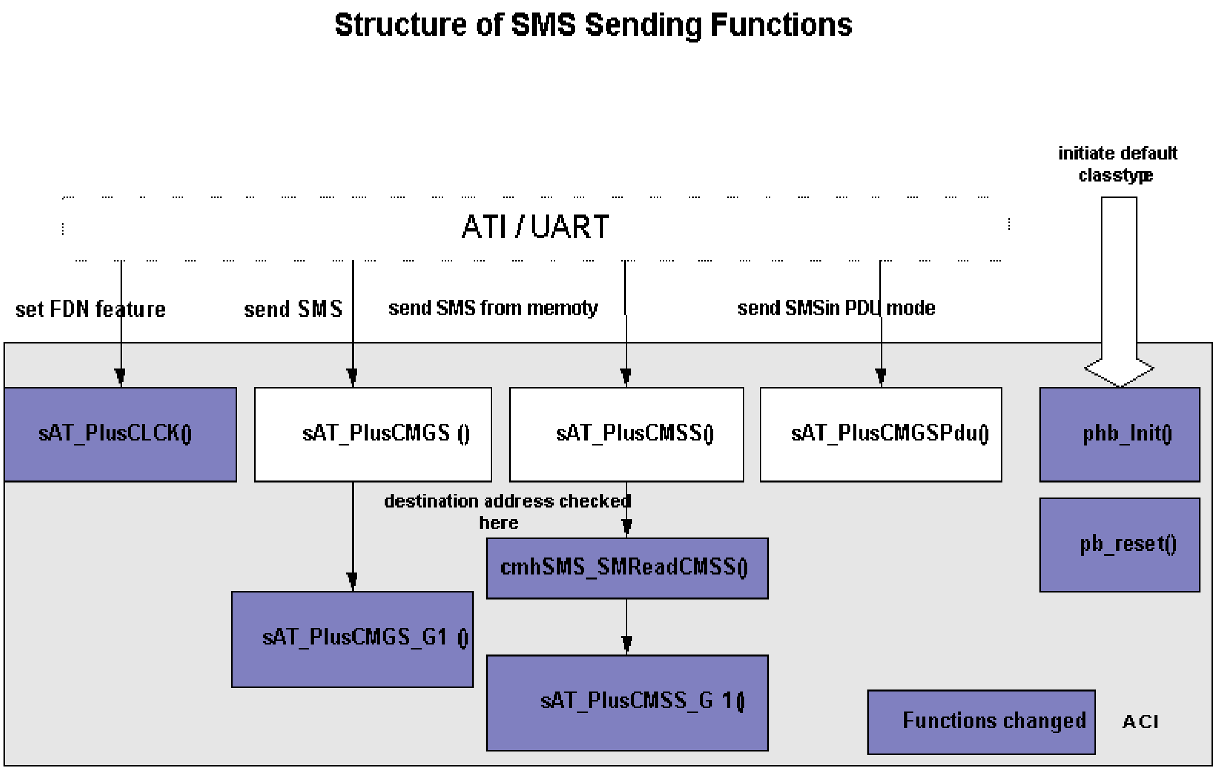

Figure 1 Related functions in SMS

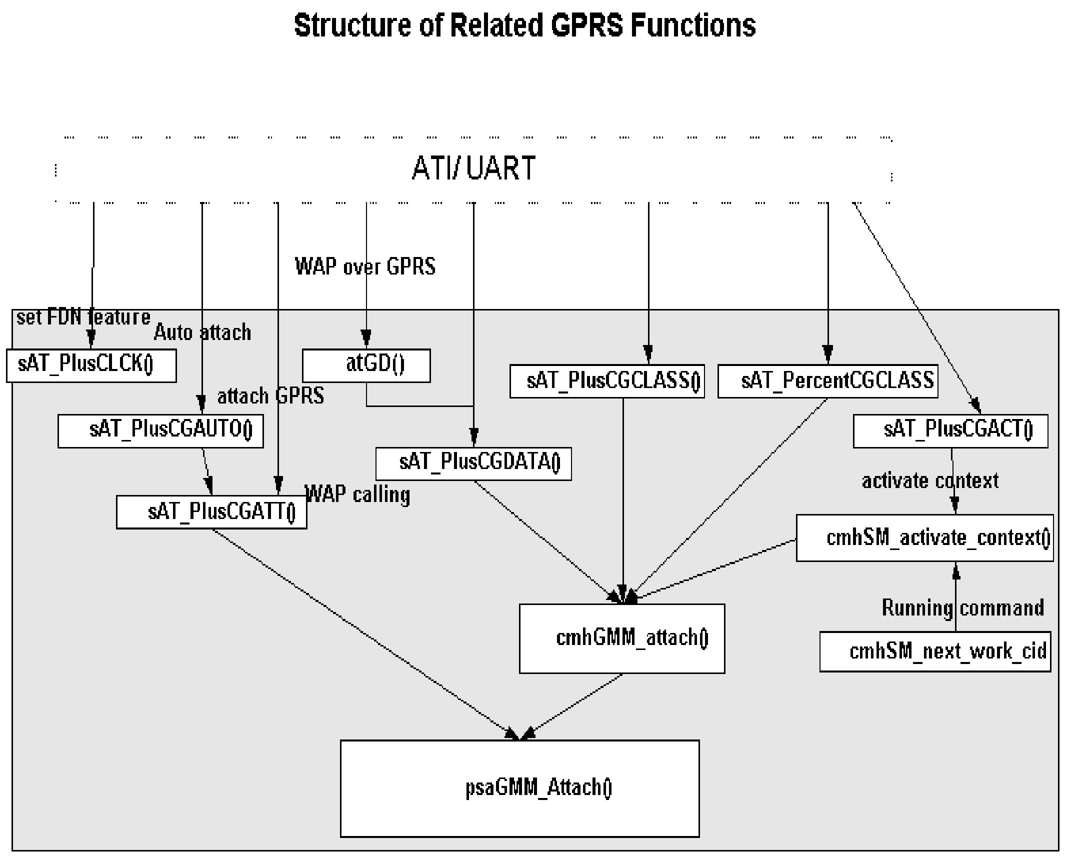

Figure 2 Related functions in GPRS

3 Changes in SMS

3.1 Basic Principle

The FDN related feature in SMS is implemented in a way that the old functions and functionalities are kept and the new feature is added to the existing functions. The changed functions are marked in purple in Figure 1. The different services are defined as class type in the phonebook module. In the existing SMS module, the services of voice, data and fax (corresponding to class type CLASS_VceDatFax) are not allowed when the FDN feature is activated. For the new release, the services of voice, data, fax and SMS (corresponding to class type CLASS_VceDatFaxSms) should be blocked if the FDN feature is activated and no match is found in the FD phonebook. If the sending of the SMS fails, an error of "operation not allowed" will be returned.

3.2 Variables Added

3.2.1 fdn_classtype and fdn_input_classtype

Definition:

static T_ACI_CLASS fdn_classtype;

static T_ACI_CLASS fdn_input_classtype;

Use:

Variable fdn_classtype is used to indicate the class type of the services that will be blocked when the FDN feature is enabled. It is initialised in functions phb_Init() and pb_reset(), reset in function pb_switch_and_fdn(). When power up, fdn_classtype will be updated to the last state saved in FFS. This is done in function pb_build_req(). This variable is globally managed in the new added functions UBYTE pb_get_fdn_mode() and void pb_set_fdn_mode(). Variable fdn_input_classtype is set each time when fdn_classtype is set plus when the AT command +CLCK is invoked. And in the call back of +CLCK, fdn_input_classtype will be used as a reference for fdn_classtype. If they are not the same, fdn_classtype will be set to the value of fdn_input_classtype.

3.3 Functions Changed

3.3.1 phb_init ()

Prototype:

void phb_init (void);

Parameters:

Void

Return:

Void

Description:

This function initializes the phonebook module when power on. The change in the function is to set the default class type and input class type to CLASS_VceDatFaxSms instead of CLASS_VceDatFax.

Changes:

Initializes the variable *fdn_input_classtype = fdn_classtype = (UBYTE) CLASS_VceDatFaxSms* when phonebook is initialized.

3.3.2 pb_reset()

Prototype:

void pb_reset (void);

Parameters:

Void

Return:

Void

Description:

This function resets the phonebook module. The change in the function is to set the default class type to CLASS_VceDatFaxSms instead of CLASS_VceDatFax when the phonebook is reset.

Changes:

Reset also the variable *fdn_classtype = (UBYTE) CLASS_VceDatFaxSms* when phonebook is reset.

3.3.3 pb_switch_adn_fdn()

Prototype:

T_PHB_RETURN pb_switch_adn_fdn(UBYTE mode)

Parameters:

UBYTE mode	FDN mode;

Description:

The function is called to set the fdn_mode in the stack after the +CLCK function is successfully handled.

Changes:

Compare the current fdn_classtype with the fdn_input_classtpye, if they are not the same, update the fdn_classtpye and the classtype in FFS.

3.3.4 sAT_PlusCLCK()

Prototype:

```
GLOBAL T_ACI_RETURN sAT_PlusCLCK  ( T_ACI_CMD_SRC srcId,
                                T_ACI_CLCK_FAC fac,
                                T_ACI_CLCK_MOD mode,
                                CHAR  *      passwd,
                                T_ACI_CLASS    class_type)
```

Parameters:

SrcId	source ID
Fac	CLCK facility
Mode	CLCK mode (Lock, unlock, query)
Passwd	Password
Class_type	class type

Return:

AT_FAIL	Execution of command failed
AT_CMPL	Execution of command completed
AT_EXCT	Execution of command is in progress
AT_BUSY	Execution of command is rejected due to a busy command handler

Description:

This is the implementation of the AT command AT+CLCK, which enables or disables the SIM lock for the corresponding facility.

Changed:

The change in this function is to set the input_fdn_classtype to the input class type (default is *CLASS_VceDatFaxSms*) for the later reference of the fdn_classtype when AT+CLCK command is successfully finished.

3.3.5 sAT_PlusCMGS_Gl()

Prototype:

```
GLOBAL T_ACI_RETURN sAT_PlusCMGS_Gl ( T_ACI_CMD_SRC   srcId,
                                      CHAR*          da,
                                      T_ACI_TOA*     toda,
                                      T_ACI_SM_DATA*  data,
                                     T_ACI_UDH_DATA* udh,
                                     CHAR*          sca,
                                     T_ACI_TOA*     tosca,
                                     SHORT          isReply,
                                     T_CMSS_FCT     rplyCB,
                                     T_ERROR_FCT    errorCB);
```

Parameters:

da	destination address
toda	type of destination address
data	message data
udh	user data header
sca	service centre address
tosca	type of service centre address
isReply	> 0: set TP-Reply-Path explicitly

	= 0: clear TP-Reply-Path explicitly
rplyCB	reply call-back
errorCB	error call-back

Return:

AT_FAIL	Execution of command failed
AT_CMPL	Execution of command completed
AT_EXCT	Execution of command is in progress
AT_BUSY	Execution of command is rejected due to a busy command handler

Description:

This is the functional counterpart of the +CMGS AT command, which is responsible for sending a short message.

Change:

After the parameters <da> and <toda> are checked, the FDN feature will be checked. If FDN is enabled and the classtype is CLASS_VceDatFaxSms, the destination number will be checked according to the FDN entries. If there is a match found in the entries, the SMS will be sent; otherwise the sending of the SMS will be blocked.

3.3.6 sAT_PlusCMGSPdu()

Prototype:

```
GLOBAL T_ACI_RETURN sAT_PlusCMGSPdu ( T_ACI_CMD_SRC srcId,
                                      T_ACI_SM_DATA *pdu );
```

Parameters:

SrcId	Source ID
Pdu	PDU data

Return:

AT_FAIL	Execution of command failed
AT_CMPL	Execution of command completed
AT_EXCT	Execution of command is in progress
AT_BUSY	Execution of command is rejected due to a busy command handler

Description:

This is the functional counterpart of the +CMGS AT command, which is responsible for sending a short message in PDU mode.

Change:

After the decoding of the message is done check the FDN feature. If FDN is enabled and the classtype is CLASS_VceDatFaxSms, the destination number will be checked according to the FDN entries. If there is a match found in the entries, the SMS will be sent; otherwise the sending of the SMS will be blocked.

3.3.7 sAT_PlusCMSS_Gl()

Prototype:

```
GLOBAL T_ACI_RETURN sAT_PlusCMSS_Gl ( T_ACI_CMD_SRC srcId,
                                      UBYTE         index,
                                      CHAR*         da,
                                      T_ACI_TOA*    toda,
                                      T_CMSS_FCT    rplyCB,
                                      T_ERROR_FCT   errorCB );
```

Parameters:

srcId	source ID
index	storage area index
da	destination address
toda	type of destination address
rplyCB	reply call-back
errorCB	error call-back

Return:

AT_FAIL	Execution of command failed
AT_CMPL	Execution of command completed
AT_EXCT	Execution of command is in progress
AT_BUSY	Execution of command is rejected due to a busy command handler

Description:

This is the functional counterpart of the +CMSS AT command, which is responsible for sending a short message from storage.

Changes:

After the parameters <srcID>, <da> and <toda> are checked no error, check also the FDN feature. If the FDN feature is on and the class type is CLASS_VceDatFaxSms, the <da> will be checked against the FDN entries. Because the SMS sent from storage can either be sent to the <da> specified by the AT command or to the <da> together with the stored message, the checking of the <da> is done in two ways. If the <da> is specified in the AT command, the checking is performed in the function itself. If the <da> is not present, the checking of the <da> will be done in the function cmhSMS_SMReadCMSS which is invoked by the read request function psaSMS_ReadReq().

3.3.8 cmhSMS_SMReadCMSS()

Prototype:

```
GLOBAL void cmhSMS_SMReadCMSS ( T_MNSMS_READ_CNF * mnsms_read_cnf);
```

Parameters:

mnsms_read_cnf	confirmation of SMS read

Return:
Void

Description:

This function processes the mnsms_read_cnf () triggered by +CMSS.

Changes:

Check the <da> with the stored message if the FDN feature is enabled and the class type is set to CLASS_VceDatFaxSms.

3.3.9 pb_check_fdn ()

Prototype:

**UBYTE pb_check_fdn(UBYTE type, UBYTE *number, UBYTE *result,
T_PHB_RECORD *entry, UBYTE toa);**

Parameters:

type	type of phonebooks
Number	<da>being compared
Result	if there is match or not
Entry	entry in phonebook
Toa	type of address

Return:

PHB_FAIL	execution of command failed
PHB_OK	execution of command completed

Description:

This function checks if a match for <da> is found in the FDN phonebook. The parameter "result" shows the result.

Changes:

Check the <da> even if there is no <toa> present.

3.3.10 Pb_build_req ()

Prototype:

void pb_build_req(T_SIM_MMI_INSERT_IND *sim_mmi_insert_ind)

Parameters:

sim_mmi_insert_ind	Sim insert indication

Return:

void

Description:

This function requests to build phonebook.

Changes:

Read the last fdn_classtype from FFS since SIM will not memorize the class type.

3.4 Functions Added

3.4.1 pb_get_fdn_classtype()

Prototype:

T_ACI_CLASS pb_get_fdn_classtype (void)

Parameters:

Void

Return:

CLASS_VceDatFax	Voice, data and fax class type
CLASS_VceDatFaxSms	Voice, data, fax and SMS class type

Description:

This function returns the FDN classtype.

3.4.2 pb_set_fdn_classtype()

Prototype:

void pb_set_fdn_classtype (T_ACI_CLASS classtype)

Parameters:

Classtype	class type

Return:
void

Description:

This function sets the FDN classtype in the callback of +CLCK.

3.4.3 pb_get_fdn_input_classtype()

Prototype:

T_ACI_CLASS pb_get_fdn_input_classtype (void)

Parameters:

Void

Return:

CLASS_VceDatFax	Voice, data and fax class type
CLASS_VceDatFaxSms	Voice, data, fax and SMS class type

Description:

This function returns the FDN input classtype.

3.4.4 pb_set_fdn_input_classtype()

Prototype:

void pb_set_fdn_input_classtype (T_ACI_CLASS classtype)

Parameters:

Classtype	class type

Return:

void

Description:

This function sets the FDN classtype in the function pb_switch_and_fdn() invoked by a callback function of +CLCK.

4 Changes in GPRS

4.1 Basic Principle

In the existing GPRS module, the attachment to GPRS network is done despite of the FDN state. For the new release, the following feature requirement should be fulfilled. The GPRS network cannot be attached if the FDN feature is activated and the entry “*99#” is not found in FD phonebook. Since there are several ways through which the mobile can be attached to the GPRS network, the implementation here finds out all the possible ways and performs a FDN check there before each attachment. If the attachment fails, an error message of “+CME ERROR: operation not allowed” will be returned.

4.2 Variables

4.2.1 fdn_mode

Definition:

static UBYTE fdn_mode; This is a variable added for the FDN dialling support, which is also an indicator for the GSM and GPRS. The possible values of this variable are:

FDN_ENABLE: FDN feature is activated;
FDN_DISABLE: FDN feature is deactivated;
NO_OPERATION: No operation about FDN, the initial state of FDN.

Use:

This variable is used to indicate whether or not the FDN feature is enabled. It is initialised in functions phb_Init() and pb_reset(), reset in the related functions invoked by +CLCK AT command and will be used in the new added functions UBYTE gprs_get_fdn_mode() and void gprs_set_fdn_mode().

4.3 Functions Changed

4.3.1 sAT_PlusCLCK()

Prototype:

```
GLOBAL T_ACI_RETURN sAT_PlusCLCK ( T_ACI_CMD_SRC srcId,
                                   T_ACI_CLCK_FAC fac,
                                   T_ACI_CLCK_MOD mode,
                                   CHAR *        passwd,
                                   T_ACI_CLASS    class_type)
```

Parameters:

srcId	source ID
fac	CLCK facility
mode	CLCK mode
passwd	password
class_type	class type

Return:

AT_FAIL	Execution of command failed
AT_CMPL	Execution of command completed
AT_EXCT	Execution of command is in progress
AT_BUSY	Execution of command is rejected due to a busy command handler

Description:

This is the implementation of the AT command AT+CLCK, which enables or disables the SIM lock.

Changes:

The changes in this function for the SMS are described in section 3.3.4 . The changes for the GPRS part is to set fdn_mode according to the input parameters;

4.3.2 sAT_PlusCGATT()

Prototype:

GLOBAL T_ACI_RETURN sAT_PlusCGATT (T_ACI_CMD_SRC srcId, T_CGATT_STATE state)

Parameters:

srcID	source ID
state	attach state

Return:

AT_FAIL	Execution of command failed
AT_CMPL	Execution of command completed
AT_EXCT	Execution of command is in progress
AT_BUSY	Execution of command is rejected due to a busy command handler

Description:

This is the functional counterpart to the +CGATT, which attaches or detaches the cell phone the GPRS network.

Changes:

When an attachment to the GPRS network is required, fdn_mode will be checked. If FDN feature is on and no entry as "*99#" is found in the FD phonebook then the attachment will not be performed. AT_FAIL will be returned and an error message of "+CME ERROR: operation not allowed" will be returned.

4.3.3 sAT_PlusCGDATA()

Prototype:
GLOBAL T_ACI_RETURN sAT_PlusCGDATA (T_ACI_CMD_SRC srcId, char *L2P, SHORT *cids)

Parameters:

srcId	source ID
L2P	
cids	context ID

Return:

AT_FAIL	Execution of command failed
AT_CMPL	Execution of command completed

AT_EXCT	Execution of command is in progress
AT_BUSY	Execution of command is rejected due to a busy command handler

Description:

Changes:

Check fdn_mode, if FDN feature is on and no entry as "*99#" is found in the FD phonebook then the attachment to the GPRS network will not be performed. AT_FAIL will be returned and "+CME ERROR: operation not allowed" will be returned.

4.3.4 SAT_PlusCGClass()

Prototype:

GLOBAL T_ACI_RETURN sAT_PlusCGCLASS (T_ACI_CMD_SRC srcId, T_CGCLASS_CLASS m_class)

Parameters:

srcID	source ID
m_class	mobile class

Return:

AT_FAIL	Execution of command failed
AT_CMPL	Execution of command completed
AT_EXCT	Execution of command is in progress
AT_BUSY	Execution of command is rejected due to a busy command handler

Description:

This is the functional counterpart to the +CGCLASS used to set the mobile class. Before the changing of the class, the mobile attempts to attach the GPRS network if necessary.

Changes:

When the mobile class is changed to a class that needs a GPRS attachment (eg. From CC to B), fdn_mode will be checked. If FDN feature is on and no entry as "*99#" is found in the FD phonebook then the attachment will not be performed and an error of "+CME ERROR: operation not allowed" will be returned.

4.3.5 SAT_PercentCGClass()

Prototype:

GLOBAL T_ACI_RETURN sAT_PercentCGCLASS (T_ACI_CMD_SRC srcId, T_CGCLASS_CLASS m_class)

Parameters:

srcID	source ID
m_class	mobile class

Return:

AT_FAIL	Execution of command failed

AT_CMPL	Execution of command completed
AT_EXCT	Execution of command is in progress
AT_BUSY	Execution of command is rejected due to a busy command handler

Description:

This is the functional counterpart to the %CGCLASS= GPRS AT command which sets the specified GPRS mobile class.

Changes:

When the mobile class is changed to a class that needs a GPRS attachment (eg. From CC to B), fdn_mode will be checked. If FDN feature is on and no entry as “*99#” is found in the FD phonebook then the attachment will not be performed and an error of of “+CME ERROR: operation not allowed” will be returned.

4.3.6 atGD()

Prototype:
GLOBAL T_ATI_RSLT atGD (char *cl, UBYTE srcId, BOOL *gprs_command)

Parameters:

cl	Command line
srcID	Source ID
gprs_command	Indication if the command line is valid

Return:

AT_FAIL	Execution of command failed
AT_CMPL	Execution of command completed
AT_EXCT	Execution of command is in progress
AT_BUSY	Execution of command is rejected due to a busy command handler
ATI_FAIL_NO_OUTPUT	Execution of command failed but no output
ATI_CMPL_NO_OUTPUT	Execution of command completed but no output

Description:
This is the functional counterpart to the +ATD AT command.

Changes:
Check fdn_mode before processing WAP over GPRS. But the FDN check is done in the function sAT+CGDATA() called by ATD*99# to make an attachment to the GPRS network.

4.3.7 cmhSM_activate_context ()

Prototype:
GLOBAL void cmhSM_activate_context(void)

Parameters:
void

Return:
void

Changes:

Function prototype change to GLOBAL T_ACI_RETURN cmhSM_activate_context(void) which indicates the result of the context activation. Check fdn_mode before any further processes. If FDN is enabled, AT_FAIL will be returned.

4.3.8 sAT_PlusCGACT ()

Prototype:

GLOBAL T_ACI_RETURN sAT_PlusCGACT (T_ACI_CMD_SRC srcId, T_CGACT_STATE state, SHORT *cids)

Parameters:

srcid	Source ID
state	CGACT command state
cids	context ID

Return:

AT_FAIL	Execution of command failed
AT_CMPL	Execution of command completed
AT_EXCT	Execution of command is in progress
AT_BUSY	Execution of command is rejected due to a busy command handler

Changes:

Add a handling of the returned value from function cmhSM_activate_context(). But in fact, this function is only used to deactivate context now, so the implementation will not be tested but will be left here for the future.

4.3.9 cmhSM_next_work_cid()

Prototype:

GLOBAL BOOL cmhSM_next_work_cid (T_ACI_AT_CMD curCmd)

Parameters:

curCmd	AT command ID

Return:

An indicator showing whether the next context ID is valid.

Changes:

Add a handling of the returned value from function cmhSM_activate_context(). The SM state, gPPP state, pointer and work_cids should be reset if the running command is AT_CMD_CGACT and the return from cmhSM_activate_context() is AT_FAIL.

4.4 Functions Added

4.4.1 pb_get_fdn_mode()

Prototype:

static UBYTE pb_get_fdn_mode (void)

Parameters:

Void

Return:

NO_OPERATION	No operation for FDN;
FDN_ENABLE	FDN feature is activated;
FDN_DISABLE	FDN feature is deactivated.

Description:

This function returns the FDN mode.

4.4.2 pb_set_fdn_mode()

Prototype:

void pb_set_fdn_mode (UBYTE fdnmode)

Parameters:

fdnmode	CLCK mode

Return:

void

Description:

This function sets the FDN mode.

5 Test Plan

This section describes how the new features are to be tested. The test includes target test and windows simulation test. The target test is only focusing on the verification of the new feature. Windows simulation test includes the standard routine test (with the existing test cases) and the new feature test (with the new test cases).

The test procedure will be divided into different scenarios, namely test cases. Each test case will be divided into test steps. Test steps are the elements of test cases. In each test step several sub steps can be performed. In this section test steps will be described and based on the test steps, test cases will be described. Additionally an overview of the test result will be given.

5.1 New Defined Test Steps

5.1.1 Test Step 1: Access FD phb in text mode

The following tables shows the sub steps defined for the test step **acess_fd_phb_in_text_mode__aciphb071()** added in aciphb_test test and as test step **access_fd_phb__gaci901()** in gaci_test.

Sub Steps	Description	Corresponding commands/functions	Expected Results
a.	Initialize phonebooks	initialize_phonebook_aciphb051()	NA
b.	Set error level to 2	AT+CMEE=2	OK
c.	Set to text mode	AT+CMGF=1	OK
d.	Access FD phonebook	AT+CPBS=”FD”	OK

MSC:

Pre-run: initialize_phonebook_aciphb051()

```
UART                              ACI                            PS
   |                                |                              |
   |           AT+CMEE=2            |                              |
   *----------------------------->  *                              |
   |                                |                              |
   |              OK                |                              |
   *< ------------------------------*                              |
   |                                |                              |
   |           AT+CMGF=1            |                              |
   *----------------------------->  *                              |
   |                                |                              |
   |              OK                |                              |
   *< ------------------------------*                              |
   |                                |                              |
   |        AT+CPBS= “FD”           |                              |
   *----------------------------->  *                              |
   |                                |                              |
   |             OK                 |                              |
   *<---------------------------- *                           |
   |                              |                           |
```

5.1.2 Test Step 2: Activate FDN in text mode

The following tables shows the sub steps defined for the test step **activate_fdn_in_text_mode__aciphb072()** added in aciphb_test test and **activate_fdn__gaci902**() in gaci_test.

Sub Steps	Description	Corresponding commands/functions	Expected Results
a.	Set to text mode	AT+CMGF=1	OK
b.	Activate FDN feature	AT+CLCK=”FD”,1,”1234”,15	OK
c.	Check FDN State	AT+CLCK= “FD”, 2	+CLCK: 1 OK

MSC:

```
UART                             ACI                           PS
   |                              |                           |
   |          AT+CMGF=1           |                           |
   *----------------------------> *                           |
   |                              |                           |
   |            OK                |                           |
   *< ----------------------------*                           |
   |                              |                           |
   |    AT+CLCK="FD",1, "1234", 15|                           |
   *----------------------------> *                           |
   |                              |                           |
   |                              |      sim_activate_req     |
   |                              *-------------------------->*
   |                              |                           |
   |                              |      sim_activate_cnf     |
   |                              *<------------------------- *
   |                              |                           |
   |                              |     sim_read_record_req   |
   |                              |-------------------------->*
   |                              |                           |
   |            OK                |                           |
   *< ----------------------------*                           |
   |                              |                           |
   |                              |     sim_read_record_cnf   |
   |                              *<--------------------------*
   |                              |                           |
   |                              |     sim_read_record_req   |
   |                              |-------------------------->*
   |                              |                           |
   |                              |     sim_read_record_cnf   |
   |                              *<--------------------------*
   |                              |                           |
   |                              |     sim_read_record_req   |
   |                              |-------------------------->*
   |                               |                           |
   |                               |     sim_read_record_cnf   |
   |                               *<--------------------------*
   |                               |                           |
   |                               |     sim_read_record_req   |
   |                               |-------------------------->*
   |                               |                           |
   |                               |     sim_read_record_cnf   |
   |                               *<--------------------------*
   |                               |                           |
   |        AT+CLCK= "FD",2        |                           |
   *----------------------------> *                            |
   |                               |                           |
   |            +CLCK: 1           |                           |
   *<---------------------------- *                            |
   |                               |                           |
   |             OK                |                           |
   *< ----------------------------*                            |
```

5.1.3 Test Step 3: Empty FD phb

The following tables shows the sub steps defined for the test step **empty_entries_in_fd_phb__aciphb072()** added in aciphb_test test and **empty_entries_in_fd_phb__gaci903 ()** in gaci_test.

Sub Steps	Description	Corresponding commands/functions	Expected Results
a.	Check used and max records used in FD phb	AT+CPBS?	+CPBS: "FD",2,4 OK
c.	Attempt to overwrite the entry 1 with empty content	AT+CPBW=1	+CME ERROR: SIM PIN2 required
d.	Give in PIN2	AT+CPIN= "1234"	OK
e.	Overwrite entry 1	AT+CPBW=1	OK
f.	Overwrite entry 2	AT+CPBW=2	OK
h.	Check if the phone book is cleaned up	AT+CPBS?	+CPBS: "FD",0,4 OK

MSC:

Pre-run: Test Step1& Step 2

```
UART                             ACI                            PS
   |                               |                           |
   |          AT+CPBS?             |                           |
   *------------------------------>*                           |
   |                               |                           |
   |        +CPBS: "FD",2,4        |                           |
   *< ----------------------------*                            |
   |                              |                           |
   |            OK                |                           |
   *< ----------------------------*                           |
   |                              |                           |
   |         AT+CPBS?             |                           |
   *----------------------------> *                           |
   |                              |                           |
   |     +CPBS: “FD”,2,4          |                           |
   *< ----------------------------*                           |
   |                              |                           |
   |       AT+CPBW=1              |                           |
   *----------------------------> *                           |
   |                              |                           |
   | +CME ERROR: SIM PIN2 required|                           |
   *< ----------------------------*                           |
   |                              |                           |
   |         AT+CPIN= “1234”      |                           |
   *----------------------------> *                           |
   |                              |                           |
   |                              |     sim_verify_pin_req    |
   |                              *-------------------------->*
   |                              |                           |
   |                              |     sim_verify_pin_cnf    |
   |                              *<------------------------- *
   |                              |                           |
   |              OK              |                           |
   *<---------------------------- *                           |
   |                              |                           |
   |       AT+CPBW=1              |                           |
   *----------------------------> *                           |
   |                              |                           |
   |                              |    sim_update_record_req  |
   |                              |-------------------------->*
   |                              |                           |
   |                              |    sim_update_record_cnf  |
   |                              *<--------------------------*
   |                              |                           |
   |       AT+CPBW=2              |                           |
   *----------------------------> *                           |
   |                              |                           |
   |                              |    sim_update_record_req  |
   |                              |-------------------------->*
   |                              |                           |
   |                              |    sim_update_record_cnf  |
   |                              *<--------------------------*
   |                              |                           |
   |          AT+CPBS?            |                           |
   *----------------------------->*                           |
   |                              |                           |
   |        +CPBS: “FD”,0,4       |                           |
   *< ----------------------------*                           |
   |                              |                           |
   |            OK                |                           |
   *< ----------------------------*                           |
```

5.1.4 Test Step 4: Add entry (+491792546349) into empty FD phonebook

The following tables shows the sub steps defined for the test step **add_one_entry_into_empty_fd_phb__aciphb074()** added in aciphb_test test.

Sub Steps	**Description**	**Corresponding commands/functions**	**Expected Results**
a.	Step 3	Empty the FD phonebook	NA
b.	Add entry into FDN	AT+CPBW=, "+491792546349", ,	OK
c.	Check phb being used	AT+CPBS?	+CPBS: "FD",1,4
d.	Check phb entry	AT+CPBR=1,4	+CPBR: 1, "491792546349",145, ""

MSC:

Pre-run: Test Step3

```
UART                              ACI                          PS
   |                               |                            |
   |     AT+CPBW=, "+491792546349", , |                         |
   *------------------------------>*                            |
   |                               |                            |
   |                               |    sim_update_record_req   |
   |                               |--------------------------->*
   |                               |                            |
   |                               |    sim_update_record_cnf   |
   |                               *<---------------------------*
   |                               |                            |
   |             OK                |                            |
   *< -----------------------------*                            |
   |                               |                            |
   |           AT+CPBS?            |                            |
   *------------------------------>*                            |
   |                               |                            |
   |         +CPBS: "FD",1,4       |                            |
   *< -----------------------------*                            |
   |                               |                            |
   |             OK                |                            |
   *< -----------------------------*                            |
   |                               |                            |
   |           AT+CPBR=1,4         |                            |
   *------------------------------>*                            |
   |                               |                            |
   |+CPBR: 1, "491792546349",145, ""                            |
   *< -----------------------------*                            |
   |                               |                            |
   |             OK                |                            |
   *< -----------------------------*                            |
   |                               |                            |
```

5.1.5 Test Step 5: Sent SMS in text mode rejected

The following tables shows the sub steps defined for the test step **sending_of_new_sms_in_text_mode_rejected__aciphb075()** added in aciphb_test test.

Sub Steps	Description	Corresponding commands/functions	Expected Results
a.	Send a new SMS to <da>	AT+CMGS= “+491792546349”	>
b.	Write “test” as the message content and ctrl+z to send out the message.	test<CTRL+Z>	+CMS ERROR: operation not allowed

MSC:

Pre Configuration: Operation in text mode

```
UART                             ACI                             PS
   |                              |                               |
   |     AT+CMGS= “+491792546349”  |                               |
   *----------------------------->*                               |
   |                              |                               |
   |             >                |                               |
   *< ----------------------------*                               |
   |                              |                               |
   |        test<CTRL+Z>          |                               |
   *----------------------------->*                               |
   |                              |                               |
   |+CMS ERROR: operation not allowed |                           |
   *< ----------------------------*                               |
```

5.1.6 Test Step 6: Sent SMS in text mode allowed

The following tables shows the sub steps defined for the test step **sending_of_new_sms_in_text_mode_allowed__aciphb076()** added in aciphb_test test.

Sub Steps	Description	Corresponding commands/functions	Expected Results
a.	Send a new SMS to <da>	AT+CMGS= “+491792546349”	>
b.	Write “test” as the message content and ctrl+z to send out the message.	test<CTRL+Z>	+CMGS: n OK

MSC:

Pre Configuration: Operation in text mode

```
UART                                ACI                            PS
   |                                 |                              |
   |    AT+CMGS= "+491792546349"     |                              |
   *----------------------------->*                              |
   |                                 |                              |
   |            >                    |                              |
   *< ----------------------------*                              |
   |                                 |                              |
   |       test<CTRL+Z>              |                              |
   *----------------------------->*                              |
   |                                 |                              |
   |                                 |    mnsms_submit_req          |
   |                                 |-------------------------->*
   |                                 |                              |
   |                                 |    mnsms_submit_cnf          |
   |                                 *<--------------------------*
   |                                 |                              |
   |           +CMGS: n              |                              |
   *< ----------------------------*                              |
   |                                 |                              |
   |              OK                 |                              |
   |<-----------------------------*                              |
```

5.1.7 Test Step 7: Send SMS in PDU mode allowed

The following tables shows the sub steps defined for the test step **sending_of_new_sms_in_text_mode_allowed__aciphb077()** added in aciphb_test test.

Sub Steps	Description	Corresponding commands/functions	Expected Results
a.	Access PDU mode	AT+CMGF= 0	OK
b.	Send the length of the message	AT+CMGS=18	>
c.	Send hex string	0001020C91947129453694000004F4F29C0E<CTRL+Z>	+CMGS: n OK

MSC:

Pre Configuration: NA

```
UART                                ACI                            PS
   |                                 |                              |
   |          AT+CMGF=0              |                              |
   *----------------------------->*                              |
   |                                 |                              |
   |            OK                   |                              |
   *< ----------------------------*                              |
   |                                 |                              |
   |     AT+CMGS= 18                |                                 |
   *------------------------------->*                                 |
   |                                |                                 |
   |        +CMGS: >                |                                 |
   *< ----------------------------*                                   |
   |0001020C91947129453694000004F |                                   |
   |4F29C0E<CTRL+Z>                 |                                 |
   *------------------------------->*                                 |
   |                                |                                 |
   |        +CMGS: n                |                                 |
   *< ----------------------------*                                   |
   |                                |                                 |
   |               OK               |                                 |
   |<-----------------------------*                                   |
```

5.1.8 Test Step 8: Sent SMS in PDU Mode Rejected

The following tables shows the sub steps defined for the test step **sending_of_new_sms_in_pdu_mode_rejected__aciphb078()** added in aciphb_test test.

Sub Steps	Description	Corresponding commands/functions	Expected Results
a.	Access PDU mode	AT+CMGF= 0	OK
b.	Send the length of the message	AT+CMGS=18	>
c.	Send hex string	0001020C91947129453694000004 F4F29C0E<CTRL+Z>	+CMS ERROR: operation not allowed

MSC:

Pre Configuration: NA

```
UART                                 ACI                              PS
    |                                |                                 |
    |          AT+CMGF=0             |                                 |
    *------------------------------->*                                 |
    |                                |                                 |
    |             >                  |                                 |
    *< ----------------------------*                                   |
    |                                |                                 |
    |                                |                                 |
    |0001020C91947129453694000004F |                                   |
    |4F29C0E<CTRL+Z>                 |                                 |
    *------------------------------->*                                 |
    |                                |                                 |
    |+CMS ERROR: operation not allowed |                               |
    *< ----------------------------*                                   |
    |                                |                                 |
```

5.1.9 Test Step 9: Store SMS in text mode successfully

The following tables shows the sub steps defined for the test step **store_message_into_memory_in_text_mode_successfully__aciphb079()** added in aciphb_test test.

Sub Steps	**Description**	**Corresponding commands/functions**	**Expected Results**
a.	Set to text mode	AT+CMGF=1	OK
b.	Write SMS	AT+CMGW=“+491792546349”	>
c.	Write “test” as the message content and ctrl+z to store the message.	test<CTRL+Z>	+CMGW: n OK

MSC:

Pre Configuration: Operation in text mode

```
UART                                ACI                             PS
   |                                 |                               |
   |          AT+CMGF=1              |                               |
   *----------------------------->  *                               |
   |                                 |                               |
   |            OK                   |                               |
   *< ----------------------------*                               |
   |                                 |                               |
   |    AT+CMGW= “12345”             |                               |
   *------------------------------>*                               |
   |                                 |                               |
   |            >                    |                               |
   *< ----------------------------*                               |
   |                                 |                               |
   |       test<CTRL+Z>              |                               |
   *------------------------------>*                               |
   |                                 |                               |
   |                                 |    mnsms_store_req            |
   |                                 |--------------------------->*
   |                                 |                               |
   |                                 |    mnsms_store_cnf            |
   |                                 *<---------------------------*
   |                                 |                               |
   |          +CMGW: n               |                               |
   *< ----------------------------*                               |
   |                                 |                               |
   |              OK                 |                               |
   |<------------------------------*                               |
```

5.1.10 Test Step 10: Send SMS from storage successfully

The following tables shows the sub steps defined for the test step **send_message_from_storage_successfully__aciphb080()** added in aciphb_test test.

Sub Steps	Description	Corresponding commands/functions	Expected Results
a.	Send an SMS from memory	AT+CMSS=1	+CMSS: n OK

MSC:

```
UART                             ACI                           PS
   |                              |                            |
   |    AT+CMSS= 1                |                            |
   *----------------------------->*                            |
   |                              |                            |
   |                              |    mnsms_submit_req        |
   |                              |--------------------------->*
   |                              |                            |
   |                              |    mnsms_submit_cnf        |
   |                              *<---------------------------*
   |                              |                            |
   |           +CMSS: n           |                            |
   *< ----------------------------*                            |
   |                              |                            |
   |              OK              |                            |
   |<-----------------------------*                            |
```

5.1.11 Test Step 11: Send SMS from storage to a specified number successfully

The following tables shows the sub steps defined for the test step **send_message_from_storage_to_specified_number_successfully__aciphb081()** added in aciphb_test test.

Sub Steps	Description	Corresponding commands/functions	Expected Results
a.	Send a SMS from memory to a number	AT+CMSS=1, “+491792546349”	+CMSS: n OK

MSC:

```
UART                             ACI                           PS
```

```
   |                              |                            |
   | AT+CMSS= 1, “+491792546349”  |                            |
   *----------------------------->*                            |
   |                              |                            |
   |                              |    mnsms_submit_req        |
   |                              |--------------------------->*
   |                              |                            |
   |                              |    mnsms_submit_cnf        |
   |                              *<---------------------------*
   |                              |                            |
   |          +CMSS: 2            |                            |
   *< ----------------------------*                            |
   |                              |                            |
   |             OK               |                            |
   |<-----------------------------*                            |
```

5.1.12 Test Step 12: Send SMS from storage rejected

The following tables shows the sub steps defined for the test step **send_message_from_storage_rejected__aciphb082()** added in aciphb_test test.

Sub Steps	Description	Corresponding commands/functions	Expected Results
a.	Send a SMS from memory to a number	AT+CMSS=1	"+CMS ERROR: operation not allowed"

MSC:

```
UART                             ACI                           PS
   |                              |                            |
   |         AT+CMSS= 1           |                            |
   *----------------------------->*                            |
   |                              |                            |
   |+CMS ERROR: operation not allowed |                        |
   |<-----------------------------*                            |
```

5.1.13 Test Step 13: Send SMS from storage to a specified number rejected

The following tables shows the sub steps defined for the test step **send_message_from_storage_to_specified_number_rejected__aciphb083()** added in aciphb_test test.

Sub Steps	Description	Corresponding commands/functions	Expected Results

a.	Send a SMS from memory to a number	AT+CMSS=1, “+491792546349”	"+CMS ERROR: operation not allowed"

MSC:

```
UART                                ACI                            PS
   |                                 |                              |
   | AT+CMSS= 1,“+491792546349”      |                              |
   *------------------------------>*                              |
   |                                 |                              |
   |+CMS ERROR: operation not allowed |                              |
   |<------------------------------*                              |
```

5.1.14 Test Step 14: Add SS-code *99# into Empty FD phb

The following tables shows the sub steps defined for the test step add_ss_code_into_empty_fd_phb__gaci904() added in aciphb_test test.

Sub Steps	Description	Corresponding commands/functions	Expected Results
a.	Step 3	Empty the FD phonebook	NA
b.	Add entry into FDN	AT+CPBW=1, “*99#”, ,	OK
c.	Check phb being used	AT+CPBS?	+CPBS: “FD”,1,4
d.	Check phb entry	AT+CPBR=1,4	+CPBR: 1, “*99#”,145, “”

MSC:

Pre-run: Test Step3

```
UART                                ACI                            PS
   |                                 |                              |
   |     AT+CPBW=1, “*99#”, ,        |                              |
   *------------------------------>*                              |
   |                                 |                              |
   |                                 |     sim_update_record_req    |
   |                                 |---------------------------->*
   |                                 |                              |
   |                                 |     sim_update_record_cnf    |
   |                                 *<----------------------------*
   |                                 |                              |
   |              OK                 |                              |
   *< ----------------------------*                              |
   |                                 |                              |
   |           AT+CPBS?              |                              |
   *------------------------------>*                              |
   |                                 |                              |
   |         +CPBS: “FD”,1,4         |                              |
   *< ----------------------------*                             |
   |                              |                             |
   |             OK               |                             |
   *< ----------------------------*                             |
   |                              |                             |
   |           AT+CPBR=1,4        |                             |
   *----------------------------->*                             |
   |                              |                             |
   |+CPBR: 1, "*99#",145, ""      |                             |
   *< ----------------------------*                             |
   |                              |                             |
   |             OK               |                             |
   *< ----------------------------*                             |
   |                              |                             |
```

5.2 Windows Simulation Test

5.2.1 New TCs for SMS

5.2.1.1 Test Case 1: Send SMS in Text Mode Rejected

Corresponding TCs:

ACIPHB075()

Description:

If FDN feature is enabled, sending of a new SMS in text mode is rejected if <da> is not in FDN list.

Reason for Test:

To verify that the SMS will not be sent in the above-mentioned situation.

Initial Configuration:

- Access FD phb in text mode (Test Step 1)
- Activate FDN feature (Test Step 2)
- Empty FD phb (Test Step 3)

Test Sequence:

1. Test Step 3;
2. Test Step 5.

5.2.1.2 Test Case 2: Send SMS in Text Mode Allowed

Corresponding TCs:

ACIPHB076A()

Description:

If FDN feature is enabled, sending of a new SMS in text mode is allowed if <da> is in FDN list.

Reason for Test:

To verify that the SMS will be sent in the above-mentioned situation.

Initial Configuration:

- Access FD phb in text mode (Test Step 1)
- Activate FDN feature (Test Step 2)
- Empty FD phb (Test Step 3)
- Add the destination number is in the empty FD phb (Test Step 4)

Test Sequence:

1. Step 4 (includes step1, 2 and 3);
2. Step 6.

5.2.1.3 Test Case 3: Send SMS in Text Mode Allowed with FDN deactivated

Corresponding TCs:

ACIPHB076B()

Description:

If FDN feature is disabled, sending of a new SMS in text mode is allowed even if <da> is not in FDN list.

Reason for Test:

To verify that the performance of sending an SMS is ok when FDN is activated and then deactivated.

Initial Configuration:

- Access FD phb in text mode (Test Step 1)
- Activate FDN feature (Test Step 2)
- Empty FD phb (Test Step 3)
- Deactivate FDN.

Test Sequence:

1. Step 3 (include Step 1 and 2);
2. Disable FDN;
3. Step 6.

5.2.1.4 Test Case 4: Send SMS in PDU Mode Allowed

Corresponding TCs:

ACIPHB078A()

Description:

If FDN feature is enabled, sending of a new SMS in PDU mode is allowed if <da> is in FDN list.

Reason for Test:

To verify that the SMS will be sent in the above-mentioned situation.

Initial Configuration:

- Access FD phb in text mode (Test Step 1)
- Activate FDN feature (Test Step 2)
- Empty FD phb (Test Step 3)
- Add the destination number is in the empty FD phb (Test Step 4)
- Access PDU mode

Test Sequence:

1. Step 4 (includes step1, 2 and 3);
2. Access PDU mode ("AT+CMGF=0");
3. Step 7.

5.2.1.5 Test Case 5: Send SMS in PDU Mode Rejected

Corresponding TCs:

ACIPHB077()

Description:

If FDN feature is enabled, sending of a new SMS in PDU mode is not allowed if <da> is not in FDN list.

Reason for Test:

To verify that the SMS will not be sent in the above-mentioned situation.

Initial Configuration:

- Access FD phb in text mode (Test Step 1)
- Activate FDN feature (Test Step 2)
- Empty FD phb (Test Step 3)
- Access PDU mode

Test Sequence:

1. Step 3 (includes step 1 and 2);
2. Access PDU mode (“AT+CMGF=0”);
3. Step 8.

5.2.1.6 Test Case 6: Send SMS in PDU Mode Allowed with FDN Deactivated

Corresponding TCs:

ACIPHB078B()

Description:

If FDN feature is disabled, sending of a new SMS in PDU mode is allowed even if <da> is not in FDN list.

Reason for Test:

To verify that the SMS will be sent in the above-mentioned situation.

Initial Configuration:

- Access FD phb in text mode (Test Step 1);
- Activate FDN feature (Test Step 2);
- Empty FD phb (Test Step 3);
- Deactivate FDN;
- Access PDU mode (“AT+CMGF=0”).

Test Sequence:

1. Step 3 (includes step 1 and 2);
2. Deactivate FDN (“AT+CLCK= “FD”,0, “0000”,15 ”);
3. Access PDU mode (“AT+CMGF=0”);
4. Step 7.

5.2.1.7 Test Case 7: Send SMS from Storage Allowed with FDN Deactivated

Corresponding TCs:

ACIPHB080A()

Description:

If FDN is disabled, sending SMS from storage in both text mode and PDU mode is allowed even if <da> is not in FDN list.

Reason for Test:

To verify that the SMS will be sent in the above-mentioned situation.

Initial Configuration:

- Access FD phb in text mode (Test Step 1)
- Empty FD phb (Test Step 3)

Test Sequence:

1. Test step 2 (includes step 1);
2. Deactivate FDN (“AT+CLCK= “FD”,0, “0000”,15 ”);
3. Step 9;
4. Step 10;
5. Step 11;
6. Access PDU mode (“AT+CMGF=0”);
7. Step 10;
8. Step 11.

5.2.1.8 Test Case 8: Send SMS from Storage Allowed

Corresponding TCs:

ACIPHB080B()

Description:

If FDN feature is enabled, sending an SMS from memory is allowed both in text mode and PDU mode if <da> is in FDN list.

Reason for Test:

To verify that the SMS will be sent from storage in the above-mentioned situation.

Initial Configuration:

- Access text mode and activate the FDN feature (Test step 1,2)
- Add the destination number is in the FD phb (Test Step 4)

Test Sequence:

1. Step 4 (includes step1,2);
2. Step 9;
3. Step 10;

4. Step 11;
5. Access PDU mode (“AT+CMGF=0”);
6. Step 10;
7. Step 11.

5.2.1.9 Test Case 9: Send SMS from Storage Rejected

Corresponding TCs:

ACIPHB081()

Description:

If FDN feature is enabled, sending an SMS from memory is rejected both in text mode and PDU mode if <da> is in FDN list.

Reason for Test:

To verify that the SMS will not be sent from storage in the above-mentioned situation.

Initial Configuration:

- Access text mode and activate the FDN feature (Test step 1,2)
- Empty FD phb (Test Step 3)

Test Sequence:

1. Step 1;
2. Step 2;
3. Step 8;
4. Step 9;
5. Step 10;
6. Access PDU mode (“AT+CMGF=0”);
7. Step 9;
8. Step 10.

5.2.2 New TCs for GPRS

5.2.2.1 Test Case 1: Attach to GPRS Allowed

Corresponding TCs:

GACI901()

Description:

If FDN feature is disabled, attach to GPRS network is allowed for all the possible AT commands.

Reason for Test:

To verify that the GPRS network can be attached in the above-mentioned situation.

Initial Configuration:

- setup_the_routing_for_the_gaci_test_gaci000()

Test Sequence:

1. Attach via +CGATT;
2. Detach;
3. Attach via +CGDATA;
4. Detach;
5. Attach via +CGCLASS;
6. Detach;
7. Attach via %CGCLASS;
8. Detach;
9. Attach via +CGAUTO;
10. Detach;

MSC:

```
UART                                ACI                           PS
    |                                |                             |
    |          AT+CGATT=1            |                             |
    |       AT+CGDATA="PPP" etc      |                             |
    *------------------------------->*                             |
    |                                |                             |
    |                                |    gmmreg_attach_req        |
    |                                |---------------------------->*
    |                                |                             |
    |                                |    mute (2000)              |
    |                                |                             |
    |                                |    gmmreg_attach_cnf        |
    |                                *<----------------------------*
    |                                |                             |
    |              OK                |                             |
    |<-------------------------------*                             |
    |                                |                             |
    |       AT+CGATT = 0             |                             |
    *------------------------------->*                             |
    |                              |                          |
    |                              |    gmmreg_detach_req     |
    |                              |------------------------->*
    |                              |                          |
    |                              |    gmmreg_detach_cnf     |
    |                              *<-------------------------*
    |                              |                          |
    |              OK              |                          |
    |<-----------------------------*                          |
      …
```

5.2.2.2 Test Case 2: Attach to GPRS via +CGATT Rejected

Corresponding TCs:

GACI902()

Description:

If FDN feature is enabled and there is no ss code as "*99#" in the FD phonebook, attach to GPRS network is blocked.

Reason for Test:

To verify that the GPRS network will be blocked in the above-mentioned situation.

Initial Configuration:

- Activate the FDN feature (Test step 2)
- Empty FD phb (Test Step 3)

Test Sequence:

11. Step 3 (includes step 2);
12. Enable FDN;
13. Attach the GPRS network via +CGATT.

MSC:

```
UART                             ACI                        PS
    |                              |                          |
    |         AT+CGATT=1           |                          |
    *----------------------------->*                          |
    |                              |                          |
    |                              |    gmmreg_attach_req     |
    |                              |------------------------->*
    |                              |                          |
    |                              |    mute (2000)           |
    |                              |                          |
    |                              |    gmmreg_attach_rej     |
    |                              *<-------------------------*
    |                              |                          |
    |              Error           |                            |
    |<-----------------------------*                            |
    |                              |                            |
```

5.2.2.3 Test Case 3: Attach to GPRS via +CGATT Allowed

Corresponding TCs:

GACI902()

Description:

If FDN feature is enabled, attach to GPRS network is allowed if there is a ss code as "*99#" in the FD phb.

Reason for Test:

To verify that the GPRS network can be attached in the above-mentioned situation.

Initial Configuration:

- Activate FDN feature (Test step 2)
- Add SS-code into FD phb (test step 14)

Test Sequence:

1. Step 2;
2. Step 14;
3. Attach the GPRS network..

MSC:

```
UART                             ACI                           PS
    |                              |                            |
    |           AT+CGATT=1         |                            |
    *----------------------------->*                            |
    |                              |                            |
    |                              |    gmmreg_attach_req       |
    |                              |--------------------------->*
    |                              |                            |
    |                              |    mute (2000)             |
    |                              |                            |
    |                              |    gmmreg_attach_cnf       |
    |                              *<---------------------------*
    |              OK              |                            |
    |<-----------------------------|                            |
```

5.2.2.4 Test Case 4: Activate Context via +CGDATA Rejected

Corresponding TCs:

GACI903()

Description:

If FDN feature is enabled and there is no ss code as "*99#" in the FD phonebook, attach to GPRS network is blocked.

Reason for Test:

To verify that the GPRS network will be blocked in the above-mentioned situation.

Initial Configuration:

- Activate the FDN feature (Test step 2)
- Empty FD phb (Test Step 3)

Test Sequence:

14. Step 3 (includes step 2);
15. Enable FDN;
16. Attach the GPRS network via +CGDATA.

MSC:

```
UART                             ACI                          PS
   |                              |                            |
   |         AT+CGDATA="PPP"      |                            |
   *----------------------------->*                            |
   |                              |                            |
   |                              |     gmmreg_attach_req      |
   |                              |--------------------------->*
   |                              |                            |
   |                              |     mute (2000)            |
   |                              |                            |
   |                              |     gmmreg_attach_rej      |
   |                              *<---------------------------*
   |                              |                            |
   |               Error          |                            |
   |<-----------------------------*                            |
```

5.2.2.5 Test Case 5: Activate Context via +CGDATA Allowed

Corresponding TCs:

Description:

If FDN feature is enabled, attach to GPRS network is allowed if there is a ss code as "*99#" in the FD phb.

Reason for Test:

To verify that the GPRS network can be attached in the above-mentioned situation.

Initial Configuration:

- Activate FDN feature (Test step 2)
- Add SS-code into FD phb (test step 14)

Test Sequence:

4. Step 2;
5. Step 14;
6. Attach the GPRS network..

MSC:

```
UART                               ACI                         PS
   |                                |                           |
   |           AT+CGDATA="PPP"      |                           |
   *------------------------------->*                           |
   |                                |                           |
   |                                |    gmmreg_attach_req      |
   |                                |-------------------------->*
   |                                |                           |
   |                                |    mute (2000)            |
   |                                |                           |
   |                                |    gmmreg_attach_cnf      |
   |                                *<--------------------------*
   |              OK                |                           |
   |<-------------------------------|                           |
```

5.2.2.6 Test Case 6: Activate Context via +CGCLASS Rejected

Corresponding TCs:

GACI905()

Description:

If FDN feature is enabled and there is no ss code as "*99#" in the FD phonebook, attach to GPRS network is blocked.

Reason for Test:

To verify that the GPRS network will be blocked in the above-mentioned situation.

Initial Configuration:

- Activate the FDN feature (Test step 2)
- Empty FD phb (Test Step 3)

Test Sequence:

1. Step 3 (includes step 2);
2. Enable FDN;
3. Attach the GPRS network via +CGCLASS.

MSC:

```
UART                            ACI                          PS
   |                              |                           |
   |        AT+CGCLASS=”B”        |                           |
   *----------------------------->*                           |
   |                              |                           |
   |                              |    gmmreg_attach_req      |
   |                              |-------------------------->*
   |                              |                           |
   |                              |    mute (2000)            |
   |                              |                           |
   |                              |    gmmreg_attach_rej      |
   |                              *<--------------------------*
   |                              |                           |
   |              Error           |                           |
   |<-----------------------------*                           |
```

5.2.2.7 Test Case 7: Activate Context via +CGCLASS Allowed

Corresponding TCs:

GACI911()

Description:

If FDN feature is enabled, attach to GPRS network is allowed if there is a ss code as “*99#” in the FD phb.

Reason for Test:

To verify that the GPRS network can be attached in the above-mentioned situation.

Initial Configuration:

- Activate FDN feature (Test step 2)
- Add SS-code into FD phb (test step 14)

Test Sequence:

7. Step 2;

8. Step 14;
9. Attach the GPRS network..

MSC:

```
UART                            ACI                          PS
   |                              |                           |
   |           AT+CGCLASS=”B”     |                           |
   *-----------------------------*                           |
   |                              |                           |
   |                              |   gmmreg_attach_req       |
   |                              |-------------------------->*
   |                              |                           |
   |                              |   mute (2000)             |
   |                              |                           |
   |                              |   gmmreg_attach_cnf       |
   |                              *<--------------------------*
   |              OK              |                           |
   |<-----------------------------|                           |
   |                              |                           |
```

5.2.2.8 Test Case 8: Activate Context via %CGCLASS Rejected

Corresponding TCs:

GACI906()

Description:

If FDN feature is enabled and there is no ss code as “*99#” in the FD phonebook, attach to GPRS network is blocked.

Reason for Test:

To verify that the GPRS network will be blocked in the above-mentioned situation.

Initial Configuration:

- Activate the FDN feature (Test step 2)
- Empty FD phb (Test Step 3)

Test Sequence:

17. Step 3 (includes step 2);
18. Enable FDN;
19. Attach the GPRS network via %CGCLASS.

MSC:

```
UART                              ACI                         PS
   |                               |                           |
   |        AT%CGCLASS="B"         |                           |
   *------------------------------>*                           |
   |                               |                           |
   |                               |    gmmreg_attach_req      |
   |                               |-------------------------->*
   |                               |                           |
   |                               |    mute (2000)            |
   |                               |                           |
   |                               |    gmmreg_attach_rej      |
   |                               *<--------------------------*
   |                               |                           |
   |              Error            |                           |
   |<------------------------------*                           |
```

5.2.2.9 Test Case 9: Activate Context via %CGCLASS Allowed

Corresponding TCs:

GACI912()

Description:

If FDN feature is enabled, attach to GPRS network is allowed if there is a ss code as "*99#" in the FD phb.

Reason for Test:

To verify that the GPRS network can be attached in the above-mentioned situation.

Initial Configuration:

- Activate FDN feature (Test step 2)
- Add SS-code into FD phb (test step 14)

Test Sequence:

10. Step 2;
11. Step 14;
12. Attach the GPRS network..

MSC:

```
UART                             ACI                           PS
   |                              |                             |
   |          AT%CGCLASS="B"      |                             |
   *----------------------------->*                             |
   |                              |                             |
   |                              |    gmmreg_attach_req        |
   |                              |---------------------------->*
   |                              |                             |
   |                              |    mute (2000)              |
   |                              |                             |
   |                              |    gmmreg_attach_cnf        |
   |                              *<----------------------------*
   |              OK              |                             |
   |<-----------------------------|                             |
   |                              |                             |
```

5.2.2.10 Test Case 10: Attachment via +CGAUTO Rejected

Corresponding TCs:

GACI907()

Description:

If FDN feature is enabled and there is no ss code as “*99#” in the FD phonebook, attach to GPRS network is blocked.

Reason for Test:

To verify that the GPRS network will be blocked in the above-mentioned situation.

Initial Configuration:

- Activate the FDN feature (Test step 2)
- Empty FD phb (Test Step 3)

Test Sequence:

20. Step 3 (includes step 2);
21. Enable FDN;
22. Attach the GPRS network via +CGAUTO.

MSC:

```
UART                             ACI                           PS
   |                              |                             |
   |       AT+CGAUTO=1            |                             |
   *----------------------------->*                             |
   |                              |                           |
   |                              |    gmmreg_attach_req      |
   |                              |-------------------------->*
   |                              |                           |
   |                              |    mute (2000)            |
   |                              |                           |
   |                              |    gmmreg_attach_rej      |
   |                              *<--------------------------*
   |                              |                           |
   |              Error           |                           |
   |<-----------------------------*                           |
```

5.2.2.11 Test Case 11: Attachment via +CGAUTO Allowed

Corresponding TCs:

GACI913()

Description:

If FDN feature is enabled, attach to GPRS network is allowed if there is a ss code as "*99#" in the FD phb.

Reason for Test:

To verify that the GPRS network can be attached in the above-mentioned situation.

Initial Configuration:

- Activate FDN feature (Test step 2)
- Add SS-code into FD phb (test step 14)

Test Sequence:

13. Step 2;
14. Step 14;
15. Attach the GPRS network..

MSC:

```
UART                             ACI                            PS
   |                              |                           |
   |          AT+CGAUTO=1         |                           |
   *----------------------------->*                           |
   |                              |                           |
   |                                 |    gmmreg_attach_req         |
   |                                 |---------------------------->*
   |                                 |                              |
   |                                 |    mute (2000)               |
   |                                 |                              |
   |                                 |    gmmreg_attach_cnf         |
   |                                *<----------------------------*
   |                                 |                              |
   |               OK                |                              |
   |<--------------------------------|                              |
```

5.2.3 Existing Windows Simulation Test

As a routine test, the following test suites will be run and the overview of the test result will be given in the next section.

- aci_test
- gaci_test
- asc_test
- aciphb_test

5.3 Test Result Overview

Windows simulation test:

Test Suite	Baseline: S300	Implemented	Comparison
Aci_test	ACI052A ACI123-ACI136 ACI162 ACI220-ACI226 ACI228 ACI229 ACI255 ACI406A ACI406D ACI500B-ACI505B ACI506-ACI544 ACI555-ACI561D ACI565A-ACI565D ACI572	ACI052A ACI123-ACI136 ACI162 ACI220-ACI226 ACI228 ACI229 ACI255 ACI406A ACI406D ACI500B-ACI505B ACI506-ACI544 ACI555-ACI561D ACI565A-ACI565D ACI572	Identical

	ACI680B-ACI684 ACI686 ACI687 950C FAILED	ACI680B-ACI684 ACI686 ACI687 950C FAILED	
Aciphb_test	100% passed	100% passed	Identical
Gaci_test	GACI036A GACI501A-GACI501D GACI602C-GACI602E GACI604B GACI607 failed	GACI036A GACI501A-GACI501D GACI602C-GACI602E GACI604B GACI607 failed	Identical
Asc_test	asc500-524 failed	asc500-524 failed	Identical

New Test cases:

SMS:

ACIPHB075()	**Passed**
ACIPHB076A()	**Passed**
ACIPHB076B()	**Passed**
ACIPHB077()	**Passed**
ACIPHB078A()	**Passed**
ACIPHB078B()	**Passed**
ACIPHB079()	**Passed**
ACIPHB080A()	**Passed**
ACIPHB080B()	**Passed**
ACIPHB080C()	**Passed**
ACIPHB081()	**Passed**

GPRS:

GACI901()	**Passed**
GACI902()	**Passed**
GACI903()	**Passed**
GACI904()	**Passed**

GACI905()	**Passed**
GACI906()	**Passed**
GACI907()	**Passed**
GACI908()	**Passed**
GACI909()	**Passed**
GACI910()	**Passed**
GACI911()	**Passed**
GACI912()	**Passed**
GACI913()	**Passed**

Target Test:

The same scenarios as the new windows test.