3GPP TSG-T (Terminals) Meeting #10
Bangkok, Thailand, 6 - 8 December, 2000

Tdoc TP-000208

3GPP TS 11.13 V1.0.0 (2000-12)

Technical Specification

3rd Generation Partnership Project; Technical Specification Group Terminals; Test specification for SIM API for Java Card™ (Release 99)

Keywords
<keyword[, keyword]>

3GPP

Postal address

3GPP support office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet
http://www.3gpp.org

Contents

1 Scope 6

2 References 6

3 Definitions and abbreviations 7
3.1 Definitions 7
3.2 Abbreviations 7

4 Test Environment 8
4.1 Applicability 8
4.2 Test environment description 8
4.3 Tests format 8
4.3.1 Test Area Reference: 8
4.3.1.1 Conformance requirements 9
4.3.1.2 Test Suite files 9
4.3.1.3 Test Procedure 9
4.3.1.4 Test Coverage 10
4.4 AID Coding 10
4.5 Test Equipment 10
4.5.1 APDU tool 11
4.5.2 Util package 11
4.5.3 Install file format 11
4.5.4 Conversion parameters file 11
4.5.5 Toolkit Test Applet 11
4.6 Testing methodology 11
4.6.1 Test interfaces and facilities 11

5 Test plan 12

6 API Test Plan 12
6.1 Package sim.toolkit 12
6.1.1 Interface ToolkitConstants 12
6.1.1.1 Constants 12
6.1.2 Interface ToolkitInterface 15
6.1.3 Class MEProfile 15
6.1.3.1 Method check 15
6.1.3.2 Method check 16
6.1.4 Class ViewHandler 17
6.1.5 Class EditHandler 18
6.1.6 Class ProactiveHandler 18
6.1.6.1 Method getTheHandler 18
6.1.6.2 Method init 19
6.1.6.3 Method initDisplayText 20
6.1.6.4 Method initGetInkey 25
6.1.6.5 Method initGetInput 29
6.1.6.6 Method send 34
6.1.6.7 Method getLength 37
6.1.6.8 Method copy 38
6.1.6.9 Method findTLV 40
6.1.6.10 Method getValueLength 42
6.1.6.11 Method getValueByte 44
6.1.6.12 Method copyValue 45
6.1.6.13 Method compareValue 48
6.1.6.14 Method findAndCopyValue 52
6.1.6.15 Method findAndCopyValue 55
6.1.6.16 Method findAndCompareValue 59
6.1.6.17 Method findAndCompareValue 63
6.1.6.18 Method appendArray 68
6.1.6.19 Method appendTLV 69

6.1.6.20 Method appendTLV 70
6.1.6.21 Method appendTLV 70
6.1.6.22 Method appendTLV 71
6.1.6.23 Method clear 72
6.1.7 Class EnvelopeHandler 73
6.1.7.1 Method getEnvelopeTag 73
6.1.7.2 Method getItemIdentifier 74
6.1.7.3 Method getSecuredDataLength 75
6.1.7.4 Method getSecureDataOffset 77
6.1.7.5 Method getTheHandler 78
6.1.7.6 Method getTPUDLOffset 79
6.1.7.7 Method getLength 81
6.1.7.8 Method copy 82
6.1.7.9 Method findTLV 84
6.1.7.10 Method getValueLength 86
6.1.7.11 Method getValueByte 88
6.1.7.12 Method copyValue 89
6.1.7.13 Method compareValue 92
6.1.7.14 Method findAndCopyValue 96
6.1.7.15 Method findAndCopyValue 99
6.1.7.16 Method findAndCompareValue 103
6.1.7.17 Method findAndCompareValue 107
6.1.8 Class ProactiveResponseHandler 113
6.1.8.1 Method copyAdditionalInformation 113
6.1.8.2 Method copyTextString 117
6.1.8.3 Method getAdditionalInformationLength 122
6.1.8.4 Method getGeneralResult 125
6.1.8.5 Method getItemIdentifier 127
6.1.8.6 Method getTextStringCodingScheme 129
6.1.8.7 Method GetTextStringLength 132
6.1.8.8 Method getTheHandler 135
6.1.8.9 Method getLength 136
6.1.8.10 Method copy 137
6.1.8.11 Method findTLV 140
6.1.8.12 Method getValueLength 142
6.1.8.13 Method getValueByte 144
6.1.8.14 Method copyValue 145
6.1.8.15 Method compareValue 148
6.1.8.16 Method findAndCopyValue 152
6.1.8.17 Method findAndCopyValue 155
6.1.8.18 Method findAndCompareValue 159
6.1.8.19 Method findAndCompareValue 163
6.1.9 Class EnvelopeResponseHandler 169
6.1.9.1 Method getTheHandler 169
6.1.9.2 Method post 170
6.1.9.3 Method postAsBERTLV 171
6.1.9.4 Method appendArray 172
6.1.9.5 Method appendTLV 173
6.1.9.6 Method appendTLV 174
6.1.9.7 Method appendTLV 175
6.1.9.8 Method appendTLV 176
6.1.9.9 Method clear 177
6.1.9.10 Method getLength 178
6.1.9.11 Method copy 179
6.1.9.12 Method findTLV 181
6.1.9.13 Method getValueLength 183
6.1.9.14 Method getValueByte 184
6.1.9.15 Method copyValue 186
6.1.9.16 Method compareValue 189
6.1.9.17 Method findAndCopyValue 193
6.1.9.18 Method findAndCopyValue 196
6.1.9.19 Method findAndCompareValue 200

6.1.9.20 Method findAndCompareValue 203
6.1.10 Class ToolkitRegistry 209
6.1.10.1 Method allocateTimer 209
6.1.10.2 Method changeMenuEntry 211
6.1.10.3 Method clearEvent 217
6.1.10.4 Method disableMenuEntry 219
6.1.10.5 Method enableMenuEntry 220
6.1.10.6 Method getEntry 222
6.1.10.7 Method getPollInterval 223
6.1.10.8 Method initMenuEntry 224
6.1.10.9 Method isEventSet 228
6.1.10.10 Method releaseTimer 229
6.1.10.11 Method requestPollInterval 231
6.1.10.12 Method setEvent 233
6.1.10.13 Method setEventList 235
6.1.11 Class ToolkitException 240
6.1.11.1 Exception Constants 240
6.1.11.2 Constructor ToolkitException 241
6.1.11.3 Method throwIt 241
6.2 Package sim.access 243
6.2.1 Interface SIMView 243
6.2.1.1 Constants 243
6.2.1.2 Method select 245
6.2.1.3 Method select 249
6.2.1.4 Method status 252
6.2.1.5 Method readBinary 254
6.2.1.6 Method updateBinary 256
6.2.1.7 Method readRecord 258
6.2.1.8 Method updateRecord 260
6.2.1.9 Method seek 263
6.2.1.10 Method increase 265
6.2.1.11 Method invalidate 267
6.2.1.12 Method rehabilitate 268
6.2.1.13 Test of File System 269
6.2.2 Class SIMSystem 269
6.2.2.1 Method getTheSIMView 269
6.2.3 Class SIMViewException 271
6.2.3.1 Method throwIt 271
6.2.3.2 Constructor 271
6.2.3.3 Reason Codes 272
6.2.4 Class SIMViewException 273
6.3 SIM Toolkit Framework: 273

7 Annex A: Acronyms 274
7.1 Annex A.1: Classes Acronyms 274
7.2 Annex A.2: Methods Acronyms 275
7.2.1 ProactiveHandler methods 275
7.2.2 ProactiveResponseHandler methods 275
7.2.3 ToolkitRegistry methods 275
7.2.4 EditHandler methods 276
7.2.5 ViewHandler methods 276
7.2.6 sim.access.SIMView Methods / Fields 277
7.2.7 EnvelopeHandler methods 278
7.2.8 EnvelopeResponseHandler methods 278

8 Annex B: Script file syntax description 278

Foreword

1 Scope

The present document describes the technical characteristics and methods of test for testing the SIM API for Java Card (TM) [x] implemented in the subscriber identity modules (SIMs) for the Pan European digital cellular communications system and Personal Communication Systems (PCS) operating in the 900 MHz and 1 800 MHz band (GSM 900 and DCS 1 800), standardized by 3GPP (to be confirmed)

The present document covers the minimum characteristics considered necessary in order to provide sufficient performance for SIMs to provide compliance to the core specifications and to ensure interoperability.

It does not necessarily include all the characteristics which may be required by a user or subscriber, nor does it necessarily represent the optimum performance achievable.

(The present document is part of the GSM-series of technical specifications) To be confirmed. The present document neither replaces any of the other GSM technical specifications or GSM related ETSs or ENs, nor is it created to provide full understanding of (or parts of) the GSM 900 and DCS 1 800 systems. The present document lists the requirements, and provides the methods of test for testing the SIM API for Java Card (TM) [x] implemented in a SIMfor conformance to the GSM standard.

For a full description of the system, reference should be made to all the GSM technical specifications or GSM related ETSs or ENs. Clause 2 provides a complete list of the GSM technical specifications, GSM related ETSs, ENs, and ETRs, on which this conformance test specifications is based.

If there is a difference between this present test specification, and any other GSM technical specification or GSM related ETS or EN, then the other GSM technical specification or GSM related ETS or EN shall prevail.

2 References

[1] GSM 01.04 version 5.0.1: "Digital cellular telecommunications system (Phase 2+); Abbreviations and acronyms".

[2] GSM 03.38 version 5.6.1: "Digital cellular telecommunications system (Phase 2+); Alphabets and language-specific information".

[3] GSM 11.11 Version 8.2.0: "Digital cellular telecommunication system (Phase 2+); Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface".

[4] GSM 11.14 version 8.2.0: "Digital cellular telecommunications system (Phase 2+); Specification of the SIM application toolkit for the Subscriber Identity Module – Mobile Equipment (SIM – ME) interface".

[5] GSM 11.17 version 7.0.2: “Subscriber Identity Module” (SIM) conformance test specification”.

[6] GSM 03.40 version 5.7.0: "Digital cellular telecommunications system (Phase 2+); Technical realisation of the Short Message Service (SMS); Point-to-Point (PP)".

[7] GSM 03.19: "Digital cellular telecommunications system (Phase 2+); Subscriber Identity Module Application Programming Interface (SIM API); SIM API for Java Card™; Stage 2".

[8] GSM 03.48: "Digital cellular telecommunications system (Phase 2+); Security Mechanisms for the SIM application toolkit; Stage 2"

[9] ISO/IEC 7816-3 (1997) " Identification cards - Integrated circuit(s) cards with contacts, Part 3: Electronic signals and transmission protocols"

[10] GSM 02.19 "Digital cellular telecommunications system (Phase 2+, Release 98); Subscriber Identity Module Application Programming Interface (SIM API); Service description; Stage 1"

[11] SUN Java Card Specification "Java Card 2.1 API Specification "

[12] SUN Java Card Specification "Java Card 2.1 Runtime Environment Specification"

[13] SUN Java Card Specification "Java Card 2.1 VM Architecture Specification"

[14] SUN Java Card Specification “Java Card 2.1 Development Kit”

[15] 3G TS 31.110 v1.0.0 “Numbering System for Telecommunication IC card applications”

3 Definitions and abbreviations

3.1 Definitions

The definitions specified in GSM 11.10-1 [12] clause 3.3 shall apply, unless otherwise specified in the present clause.

Applet installation parameters: Default values for applet installation parameters.

Test Applet: Applet designed to test an specific functionality of the SIM API 03.19 specification.

Security parameters: Minimum security requirements defined for the applet installation process.

Applet loading script: File containing the APDU commands that will load and install the test applet in the card.

Pre Test Script file: File containing the APDU commands that will verify the correct state of the card and perform the required initialisation before test execution.

Test Script file: File containing the APDU commands that will execute and verify the test results.

Post Test Script file: File containing the APDU commands that will re-initialise the card in order to restore the initial state after test execution.

Test Output file: TBD.

3.2 Abbreviations

For the purpose of the present document, the following abbreviations apply, in addition to those listed in GSM 01.04 [2]:

AC Application Code

AID Application Identifier

APDU Application Protocol Data Unit

API Application Programming Interface

CAD Card Acceptance Device

FFS For Further Study

IFD Interface Device

JCRE Java Card™ Run Time Environment

JVM Java Virtual Machine

SIM Subscriber Identity Module

SE Sending Entity

4 Test Environment

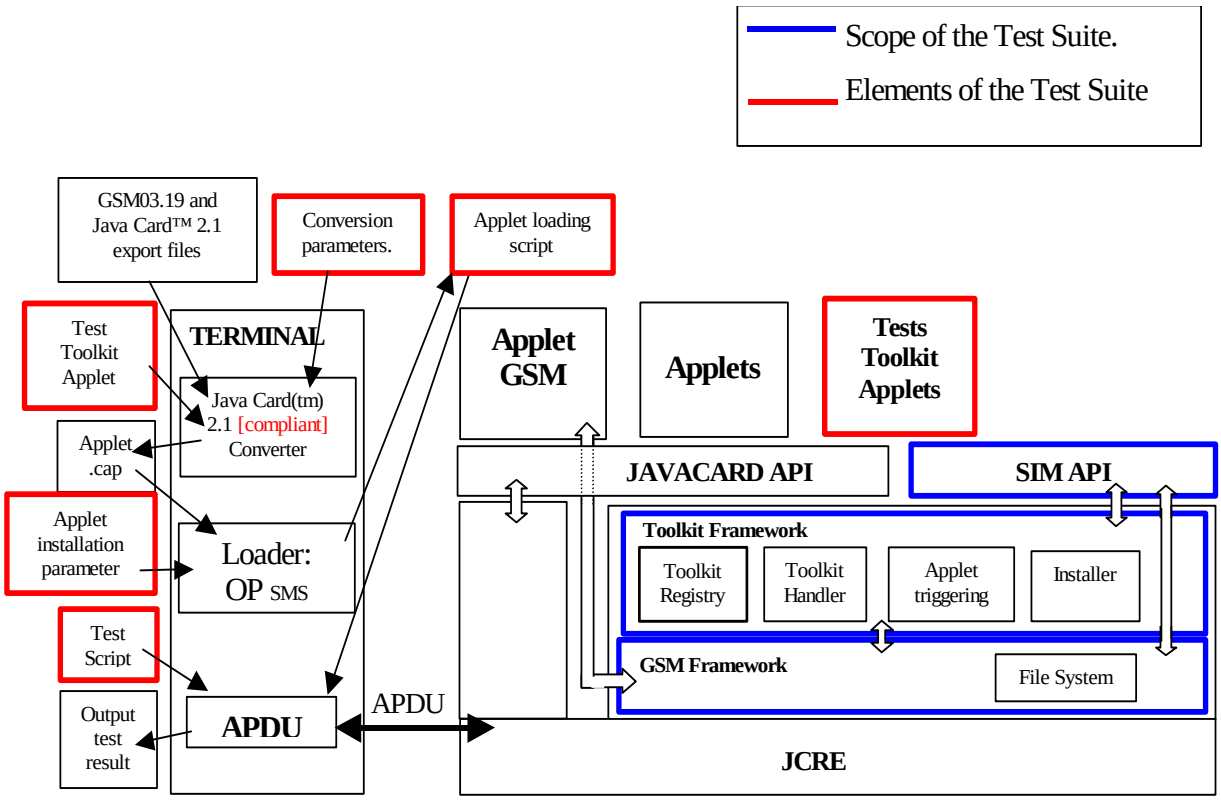
This clause specifies requirements that shall be met and the testing rules that shall be followed during the test procedure.

4.1 Applicability

The tests defined in this specification shall be performed taking into account the services supported by the card as specified in the EF SST file.

4.2 Test environment description

The general architecture for the test environment is:



4.3 Tests format

4.3.1 Test Area Reference:

Each test area is referenced as follows:

API Testing:: 'API_[package name]_[classname]_[methodname]' where

package name :

sim.access package: '1'

sim.toolkit package: '2'

class name:

yyy : 3 letters for each class.

See Annex A for full classes acronyms list.

method name:

zzzz[input parameters]:

See Annex A for full methods name acronyms list.

FWK : framework testing

[TBD]

LDR : loader testing

[TBD]

4.3.1.1 Conformance requirements

The conformance requirements are expressed in the following way:

- Method prototype as listed in GSM03.19 specification.
- Normal execution:
  - Contains normal execution and correct parameters limit values, each referenced as a Conformance Requirement Reference Normal (CRRN)
- Parameters error:
  - Contains parameters errors and incorrect parameters limit values, each referenced as a Conformance Requirement Reference Parameter Error (CRRP)
- Context error:
  - Contains errors due to the context the method is used in, each referenced as a Conformance Requirement Reference Context Error (CCRC)

4.3.1.2 Test Suite files

The files included in the Test Area use the following naming convention:

- Test Script : [Test Area Reference]_[Test script number].scr
- Test Applet: [Test Area Reference]_ [Test applet number].java
- Installation parameter: [Test Area Reference] _[Installation parameter script number].install
- Load Script : [Test Area Reference] _[Load Script number].ldr
- Conversion parameter: [Test Area Reference] _[Conversion parameters script number].cnv

The test script, applet, installation parameters, load script and conversion parameters numbers start from '1'

4.3.1.3 Test Procedure

Each test procedure contains a table to indicate the expected responses form the API and/or the APDU level as follows:

Test Case			
Id	**Description**	**API Expectation**	**APDU Expectation**
	Test Case detailed description	*API expected behavior.*	*Expected response at APDU level.*

4.3.1.4 Test Coverage

The table at the end of each test procedure indicates the correspondence between the Conformance Requirements Reference (CRR) and the different test cases.

4.4 AID Coding

The AID coding for the API Test Packages, Applet classes and Applet shall be as specified in 3G TS 31.110. In addition, the following TAR values are defined for use within the present document:

TAR (3 bytes/ 24 bits):

test package (3 bits):

000 reserved (as TAR= '00.00.00' is reserved for Card Manager)

001 API

010 Framework

011 Loader

111 org.t3.gsm0319.tests.util

other values are RFU

Specific applet test name(21 bits):

for org.t3.gsm0319.tests.util test package: 000...00

for API test package(3 bits)

001 sim.access

010 sim.toolkit

other are RFU

Class (5 bits): need to be assigned specification order see Annex [XX] for the full list

Method (6 bits): need to be assigned specification order see Annex [XX] for the full list

Applet Class number (5 bits): linked to Test Area

Applet Instance number (2 bits) defined in the test procedure, 00 for package and class

Application Provider specific field (1 byte):

'00' for Package

'01' for Applet class

'02' for Applet Instance

For example, the AID of Package org.t3.gsm0319.tests.util is 'A0 00 00 00 09 00 02 FF FF FF FF 89 E0 00 00 00'

4.5 Test Equipment

This sub clauses recommends a minimum specification for each of the items of test equipment referenced in the tests.

4.5.1 APDU tool

This test tool shall meet the following requirements:

- be able to send command to the card TPDU;
- be able to check none, only a part, or all of the data returned;
- be able to check none, only part, or all of the status returned;
- be able to accept all valid status codes returned;
- be able to support Reader commands;
- be able to generate a log file for each test execution.
- if more data is returned than defined in the test specification, the tool shall continue;
- if less data is returned than defined in the test specification, the tool shall aborts and return an error;
- if there is an error in data or status returned, the tool shall abort and return an error.

The log file produced by the test tool shall include the following information:

- all commands issued;
- all data returned;
- all status returned;
- all errors codes;
- expected data and status in case of error;
- comments from the scripts;
- a log message to report success or failure of the test.

Refer to annex B for the script file syntax definition.

4.5.2 Util package

[To include zip file with package *org.t3.gsm0319.tests.util*]

4.5.3 Install file format

[To be defined]

4.5.4 Conversion parameters file

The conversion parameters format follow the one specified in Sun's Java Card 2.1 Development Kit [14].

4.5.5 Toolkit Test Applet

4.6 Testing methodology

4.6.1 Test interfaces and facilities

The SIM-ME interface provides the main transport interface for the purpose of performing conformance tests.

The SIM API interface provides the main test interface for the purpose of performing conformance tests.

5 Test plan

The test plan is divided according to the SIM API specification, that way the tests will follow the class hierarchy for the sim.toolkit and sim.access package; for the SIM Toolkit framework this test plan describes the different points that will be tested with the present test specification.

6 API Test Plan

6.1 Package sim.toolkit

6.1.1 Interface ToolkitConstants

6.1.1.1 Constants

6.1.1.1.1 Test Area Reference: API_2_TKC_CONS

6.1.1.1.2 Conformance Requirement:

There is no API, only constants. This constants shall be compare to its definition in the API.

Normal Execution

CRRN1 : The Toolkit Constants shall all have the same name and value defined in the GSM03.19 normalization.

Parameters error

None

Context error

None

6.1.1.1.3 Test suite files:

No additional requirements for the GSM personalisation.

- Test Script: API_2_TKC_CONS.scr
- Test case trigger: 1- first applet check the first 80 constants

  2- second applet checked the 66 others.
- Test Applet: API_2_TKC_CONS_1.java

  API_2_TKC_CONS_2.java
- Installation parameter: Same as default applet
- Load Script: API_2_TKC_CONS.ldr

  At the end of the script the applet is loaded but not instantiated.
- Conversion parameter: API_2_TKC_CONS.cnv

6.1.1.1.4 Test Procedure

First applet triggered:

Test Case	
Id	**Test purpose**
01	Check constant EVENT_PROFILE_DOWNLOAD=1
02	Check constant EVENT_FORMATTED_SMS_PP_ENV=2
03	Check constant EVENT_FORMATTED_SMS_PP_UPD=3
04	Check constant EVENT_UNFORMATTED_SMS_PP_ENV=4
05	Check constant EVENT_UNFORMATTED_SMS_PP_UPD=5
06	Check constant EVENT_UNFORMATTED_SMS_CB=6
07	Check constant EVENT_MENU_SELECTION=7
08	Check constant EVENT_MENU_SELECTION_HELP_REQUEST=8
09	Check constant EVENT_CALL_CONTROL_BY_SIM=9
10	Check constant EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM=10
11	Check constant EVENT_TIMER_EXPIRATION=11
12	Check constant EVENT_EVENT_DOWNLOAD_MT_CALL12
13	Check constant EVENT_EVENT_DOWNLOAD_CALL_CONNECTED=13
14	Check constant EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED=14
15	Check constant EVENT_EVENT_DOWNLOAD_LOCATION_STATUS=15
16	Check constant EVENT_EVENT_DOWNLOAD_USER_ACTIVITY=16
17	Check constant EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE=17
18	Check constant EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS=18
19	Check constant EVENT_STATUS_COMMAND=127
20	Check constant EVENT_UNRECOGNIZED_ENVELOPE=-1
21	Check constant BTAG_PROACTIVE_SIM_COMMAND=0xD0
22	Check constant BTAG_SMS_PP_DOWNWLOAD=0xD1
23	Check constant BTAG_CELL_BROADCAST_DOWNLOAD=0xD2
24	Check constant BTAG_MENU_SELECTION=0xD3
25	Check constant BTAG_CALL_CONTROL=0xD4
26	Check constant BTAG_MO_SHORT_MESSAGE_CONTROL=0xD5
27	Check constant BTAG_EVENT_DOWNLOAD=0xD6
28	Check constant BTAG_TIMER_EXPIRATION=0xD7
29	Check constant TAG_COMMAND_DETAILS=0x01
30	Check constant TAG_DEVICE_IDENTITIES=0x02
31	Check constant TAG_RESULT=0x03
32	Check constant TAG_DURATION=0x04
33	Check constant TAG_ALPHA_IDENTIFIER=0x05
34	Check constant TAG_ADDRESS=0x06
35	Check constant TAG_CAPABILITY_CONFIGURATION_PARAMETERS=0x07
36	Check constant TAG_CALLED_PARTY_SUBADDRESS=0x08
37	Check constant TAG_SS_STRING=0x09
38	Check constant TAG_USSD_STRING=0x0A
39	Check constant TAG_SMS_TPDU=0x0B
40	Check constant TAG_CELL_BROADCAST_PAGE=0x0C
41	Check constant TAG_TEXT_STRING=0x0D
42	Check constant TAG_TONE=0x0E
43	Check constant TAG_ITEM=0x0F
44	Check constant TAG_ITEM_IDENTIFIER=0x10
45	Check constant TAG_RESPONSE_LENGTH=0x11
46	Check constant TAG_FILE_LIST=0x12
47	Check constant TAG_LOCATION_INFORMATION=0x13
48	Check constant TAG_IMEI=0x14
49	Check constant TAG_HELP_REQUEST=0x15
50	Check constant TAG_NETWORK_MEASUREMENT_RESULTS=0x16
51	Check constant TAG_DEFAULT_TEXT=0x17
52	Check constant TAG_ITEMS_NEXT_ACTION_INDICATOR=0x18
53	Check constant TAG_EVENT_LIST=0x19
54	Check constant TAG_CAUSE=0x1A
55	Check constant TAG_LOCATION_STATUS=0x1B
56	Check constant TAG_TRANSACTION_IDENTIFIER=0x1C
57	Check constant TAG_BCCH_CHANNEL_LIST=0x1D
58	Check constant TAG_ICON_IDENTIFIER=0x1E
59	Check constant TAG_ITEM_ICON_IDENTIFIER_LIST=0x1F
60	Check constant TAG_CARD_READER_STATUS=0x20
61	Check constant TAG_CARD_ATR=0x21
62	Check constant TAG_C_APDU=0x22
63	Check constant TAG_R_APDU=0x23
64	Check constant TAG_TIMER_IDENTIFIER=0x24
65	Check constant TAG_TIMER_VALUE=0x25
66	Check constant TAG_DATE_TIME_AND_TIME_ZONE=0x26
67	Check constant TAG_CALL_CONTROL_REQUESTED_ACTION=0x27
68	Check constant TAG_AT_COMMAND=0x28
69	Check constant TAG_AT_RESPONSE=0x29
70	Check constant TAG_BC_REPEAT_INDICATOR=0x2A

71	Check constant TAG_IMMEDIATE_RESPONSE=0x2B
72	Check constant TAG_DTMF_STRING=0x2C
73	Check constant TAG_SET_CR=0x80
74	Check constant TAG_SET_NO_CR=0x7F
75	Check constant TLV_LENGTH_CODED_2BYTES=0X81
76	Check constant TLV_NOT_FOUND=0X00
77	Check constant TLV_FOUND_CR_SET=0x01
78	Check constant TLV_FOUND_CR_NOT_SET=0x02
79	Check constant PRO_CMD_REFRESH=0x01
80	Check constant PRO_CMD_MORE_TIME=0x02

Second applet triggered:

Test Case	
Id	**Test purpose**
01	Check constant PRO_CMD_SET_UP_CALL=0x10
02	Check constant PRO_CMD_SEND_SS=0x11
03	Check constant PRO_CMD_SEND_USSD=0x12
04	Check constant PRO_CMD_SEND_SHORT_MESSAGE=0x13
05	Check constant PRO_CMD_SEND_DTMF=0x14
06	Check constant PRO_CMD_PLAY_TONE=0x20
07	Check constant PRO_CMD_DISPLAY_TEXT=0x21
08	Check constant PRO_CMD_GET_INKEY=0x22
09	Check constant PRO_CMD_GET_INPUT=0x23
10	Check constant PRO_CMD_SELECT_ITEM=0x24
11	Check constant PRO_CMD_PROVIDE_LOCAL_INFORMATION=0x26
12	Check constant PRO_CMD_TIMER_MANAGEMENT=0x27
13	Check constant PRO_CMD_SET_UP_IDLE_MODE_TEXT=0x28
14	Check constant PRO_CMD_PERFORM_CARD_APDU=0x30
15	Check constant PRO_CMD_POWER_ON_CARD=0x31
16	Check constant PRO_CMD_POWER_OFF_CARD=0x32
17	Check constant PRO_CMD_GET_READER_STATUS=0x33
18	Check constant PRO_CMD_RUN_AT_COMMAND=0x34
19	Check constant DEV_ID_KEYPAD=0x01
20	Check constant DEV_ID_DISPLAY=0x02
21	Check constant DEV_ID_EARPIECE=0x03
22	Check constant DEV_ID_ADDITIONAL_CARD_READER_0=0x10
23	Check constant DEV_ID_ADDITIONAL_CARD_READER_1=0x11
24	Check constant DEV_ID_ADDITIONAL_CARD_READER_2=0x12
25	Check constant DEV_ID_ADDITIONAL_CARD_READER_3=0x13
26	Check constant DEV_ID_ADDITIONAL_CARD_READER_4=0x14
27	Check constant DEV_ID_ADDITIONAL_CARD_READER_5=0x15
28	Check constant DEV_ID_ADDITIONAL_CARD_READER_6=0x16
29	Check constant DEV_ID_ADDITIONAL_CARD_READER_7=0x17
30	Check constant DEV_ID_SIM=0x81
31	Check constant DEV_ID_ME=0x82
32	Check constant DEV_ID_NETWORK=0x83
33	Check constant DCS_DEFAULT_ALPHABET=0x00
34	Check constant DCS_8_BIT_DATA=0x04
35	Check constant DCS_UCS2=0x08
36	Check constant SW1_RP_ERROR=0x9E
37	Check constant SW1_RP_ACK=0x9F
38	Check constant POLL_NO_DURATION=0
39	Check constant POLL_SYSTEM_DURATION=(-1)
40	Check constant RES_CMD_PERF=0x00
41	Check constant RES_CMD_PERF_PARTIAL_COMPR=0x01
42	Check constant RES_CMD_PERF_MISSING_INFO=0x02
43	Check constant RES_CMD_PERF_REFRESH_ADD_EF_READ=0x03
44	Check constant RES_CMD_PERF_REQ_ICON_NOT_DISP=0x04
45	Check constant RES_CMD_PERF_MODIF_CC_SIM=0x05
46	Check constant RES_CMD_PERF_SESSION_TERM_USER=0x10
47	Check constant RES_CMD_PERF_BACKWARD_MOVE_REQ=0x11
48	Check constant RES_CMD_PERF_NO_RESP_FROM_USER=0x12
49	Check constant RES_CMD_PERF_HELP_INFO_REQ=0x13
50	Check constant RES_CMD_PERF_USSD_TRANSAC_TERM=0x14
51	Check constant RES_TEMP_PB_ME_UNABLE_PROC=0x20
52	Check constant RES_TEMP_PB_SESSION_TERM_USER=0x21
53	Check constant RES_TEMP_PB_USER_REJECT_CALL_REQ=0x22
54	Check constant RES_TEMP_PB_USER_CLEAR_CALL=0x23
55	Check constant RES_TEMP_PB_IN_CONTR_TIMER_STATE=0x24
56	Check constant RES_TEMP_PB_INTERACT_CC_BY_SIM=0x25

57	Check constant RES_ERROR_CMD_BEYOND_ME_CAPAB=0x30
58	Check constant RES_ERROR_CMD_TYP_NOT_UNDERSTOOD=0x31
59	Check constant RES_ERROR_CMD_DATA_NOT_UNDERSTOOD=0x32
60	Check constant RES_ERROR_CMD_NUMBER_NOT_KNOWN=0x33
61	Check constant RES_ERROR_SS_RETURN_ERROR=0x34
62	Check constant RES_ERROR_SMS_RP_ERROR=0x35
63	Check constant RES_ERROR_REQ_VALUES_MISS=0x36
64	Check constant RES_ERROR_USSD_RETURN_ERROR=0x37
65	Check constant RES_ERROR_MULTIPLE_CARD_ERROR=0x38
66	Check constant RES_ERROR_INTERACT_CC_SMSMO_BY_SIM=0x39

We expect for each case the comparison is true

6.1.1.1.5 Test Coverage

CR number	*Test case number*
1	***each case of the two applets***

6.1.2 Interface ToolkitInterface

6.1.3 Class MEProfile

6.1.3.1 Method check

6.1.3.1.1 Test Area Reference: API_2_MEP_CHECB

6.1.3.1.2 Conformance Requirement:

The method with following header shall compliant to its definition in the API.

```
public static boolean check(byte index)
                throws  ToolkitException
```

Normal Execution

CRRN1: The method checks a facility in the handset profile.

Parameters error

CRRP1: The method shall throw ME_PROFILE_NOT_AVAILABLE ToolkitException if Terminal Profile data are not available

Context error

None

6.1.3.1.3 Test suite files:

No Additional requirements for the GSM personalisation:

- Test Script: API_2_MEP_CHECB_1.scr
- Test Applet: API_2_MEP_CHECB_1.java
- Installation parameter: API_2_MEP_CHECB.install (Same as default).
- Load Script: API_2_MEP_CHECB.ldr

  At the end of the script the applet is loaded but not instantiated.
- Conversion parameter: API_2_MEP_CHECB.cnv

6.1.3.1.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**No Terminal Profile is registered** Triggered by unformated SMS Index = 1	ME_PROFILE_NOT_AVAILABLE ToolkitException is thrown	
2	**Terminal Profile, Facility is supported** index = 0	true is returned by the method	
3	**Terminal Profile, Facility is not supported** Index = 15	false is returned by the method	

6.1.3.1.5 Test Coverage

CRR number	*Test case number*
N1	***2,3***
P1	***1***

6.1.3.2 Method check

6.1.3.2.1 Test Area Reference: API_1_MEP_CHEC_BSS

6.1.3.2.2 Conformance Requirement:

The method with following header shall compliant to its definition in the API.

```
public static boolean check(byte[] mask,
                            short offset,
                            short length)
              throws  java.lang.NullPointerException,
                      java.lang.ArrayIndexOutOfBoundsException,
                      ToolkitException
```

Normal Execution

CRRN1: The method checks all the facilities corresponding to bits set to 1 in the mask buffer.

Parameters error

CRRP1: The method shall throw java.lang.NullPointerException if mask is null.

CRRP2: The method shall throw java.lang.ArrayIndexOutOfBoundsException if offset or length or both would cause access outside array bounds.

CRRP3: The method shall throw ME_PROFILE_NOT_AVAILABLE ToolkitException if Terminal Profile data are not available.

Context error

None

6.1.3.2.3 Test suite files:

No Additional requirements for the GSM personalisation:

- Test Script: API_1_MEP_CHEC_BSS_1.scr
- Test Applet: API_1_MEP_CHEC_BSS_1.java
- Installation parameter: API_1_ MEP_CHEC_BSS.install

  Same as default applet but specific Terminal Profile :

- Display of extension Text not supported (facility index = 15)

  Terminal Profile = 0xFF7FFFFFFFFFFFFFFFFFFFFFFFFFFF

- Load Script: API_1_MEP_CHEC_BSS.ldr

  At the end of the script the applet is loaded but not instantiated.

- Conversion parameter: API_1_MEP_CHEC_BSS.cnv

6.1.3.2.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**No Terminal Profile is registered** Triggered by unformated SMS Mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F Offset = 0 Length = 16	ME_PROFILE_NOT_AVAILABLE ToolkitException is thrown	
2	**NULL as parameter to check** mask= NULL	NullPointerException is thrown	
3	**Offset > mask.length** mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F Offset = 17	ArrayIndexOutOfBounds Exception is thrown	
4	**Offset < 0** mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F Offset = -1	ArrayIndexOutOfBounds Exception is thrown	
5	**Length > mask.length** mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F Offset = 0 Length = 18	ArrayIndexOutOfBounds Exception is thrown	
6	**Offset + length > mask.length** mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F Offset = 9 Length = 9	ArrayIndexOutOfBounds Exception is thrown	
7	**Check all the Terminal Profile** mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F Offset = 0 Length = 16	false is returned by the method because facility 15 is not supported	
8	**Check a part of the Terminal Profile** mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F Offset = 15 Length = 2	true is returned by the method : the 14 first facilities have been checked successfully	

6.1.3.2.5 Test Coverage

CRR number	*Test case number*
N1	***7,8***
P1	***2***
P2	***3,4,5,6***
P3	***1***

6.1.4 Class ViewHandler

It is not possible to test the methods provided by this class as it is declared ‘abstract’; it will be done in the class inheriting it: EditHandler, EnvelopeHandler, ProactiveResponseHandler, ProactiveHandler.

6.1.5 Class EditHandler

It is not possible to test the methods provided by this class as it is declared ‘abstract’; it will be done in the class inheriting it: EnvelopeResponseHandler, ProactiveHandler.

6.1.6 Class ProactiveHandler

6.1.6.1 Method getTheHandler

6.1.6.1.1 Test Area Reference : API_2_PAH_GTHD

6.1.6.1.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public static ProactiveHandler getTheHandler()
                                    throws ToolkitException
```

Normal Execution

CRRN1: The method shall return the single system instance of the ProactiveHandler class.

Parameter Error

No requirements

Context Error

CRRC1: The method shall throw ToolkitException.HANDLER_NOT_AVAILABLE if the handler is busy.

6.1.6.1.3 Test Suite files

- Test Script: API_2_PAH_GTHD_1.scr
- Test Applet: API_2_PAH_GTHD _1.java
- Installation parameter: API_2_PAH_GTHD.install (Same as default applet)
- Load Script: API_2_PAH_GTHD.ldr
- Conversion parameter: API_2_PAH_GTHD.cnv

6.1.6.1.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**Trigger the applet with a PROFILE_DOWNLOAD**		
1	**getTheHandler() twice**	**The returned objects shall be the same**	
2	getTheHandler()	The reference shall be a ProactiveHandler	
3	getTheHandler()	The reference shall not be null	
4	Build and send a Proactive command **Do not send the Terminal Response**		Proactive command
	Trigger the applet one more time (CALL CONTROL)	HANDLER_NOT_AVAILABLE ToolkitException is	

	call the getTheHandler() method	thrown by getTheHandler()	

6.1.6.1.5 Test Coverage

CRR number	Test case number
N1	1, 2, 3
C1	4 (in framework test)

6.1.6.2 Method init

6.1.6.2.1 Test Area Reference: API_2_PAH_INITBBB

6.1.6.2.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public void init(byte type,
                 byte qualifier,
                 byte dstDevice)
```

Normal Execution

CRRN1 : The `init()` method initialises the next Proactive command in the ProactiveHandler, with Command details and Device Identities TLV. The source device is always the SIM Card (81h). The Comprehension Required flags are set.

CRRRN2 : The Command number may take any value between 01h and FEh.

CRRN3 : The `init()` method clears the ProactiveHandler before initialising it.

CRRN4 : No TLV is selected after a call to the method.

CRRN5 : The handler is not sent to the mobile by the `init()` method.

Parameter Error

No requirements

Context Error

No requirements

6.1.6.2.3 Test Suite files

- Test Script: API_2_PAH_INITBBB _1.scr
- Test Applet: API_2_PAH_INITBBB _1.java
- Installation parameter: API_2_PAH_INITBBB.install (Same as default applet)
- Load Script: API_2_PAH_INITBBB.ldr
- Conversion parameter: API_2_PAH_INITBBB.cnv

6.1.6.2.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the `init()` method** **`type = 01h`**		

	qualifier = 02h **dstDevice = 03h**		
	Verify each simple TLV of the handler by using `ViewHandler` methods		
2	**Verify the command number value**	01h-FEh	
3	**Call the `init()` method** **type = FFh** **qualifier = FEh** **destination = FDh**		
	Verify each simple TLV of the handler by using `ViewHandler` methods		
4	**Select the 1st TLV in the handler** **Call the `init()` method with any value**		
	Call the `getValueLength()` method	**`UNAVAILABLE_ELEMENT`** **`ToolkitException` is thrown by `getValueLength()`**	

6.1.6.2.5 Test Coverage

CRR number	Test case number
N1	1, 3
N2	2
N3	3
N4	4
N5	1, 3

6.1.6.3 Method initDisplayText

6.1.6.3.1 Test Area Reference : API_2_PAH_INDTBB_BSS

6.1.6.3.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public void initDisplayText(byte qualifier,
                            byte dcs,
                            byte[] buffer,
                            short offset,
                            short length)
                     throws java.lang.NullPointerException,
                            java.lang.ArrayIndexOutOfBoundsException,
                            ToolkitException
```

Normal Execution

CRRN1: The method shall build a DISPLAY TEXT proactive command in the ProactiveHandler, using `qualifier`, `dcs` and `buffer` parameters. Comprehension required flags are set.

CRRN2: A call to this method clears the handler then initialises it.

CRRN3: No TLV is selected after a call to the method.

CRRN4: The DISPLAY TEXT command is not sent by the method.

CRRN5: The Command Number may take any value between 01h and FEh.

Parameter Error

CRRP1: The method shall throw `NullPointerException` if `buffer` is `null`.

CRRP2: If `offset` or `length` or both would cause access outside array bounds, an `ArrayIndexOutOfBoundsException` shall be thrown.

Context Error

CRRC1: A `ToolkitException.HANDLER_OVERFLOW` shall be thrown if the ProactiveHandler is too small to put the requested data.

6.1.6.3.3 Test Suite files

- Test Script: API_2_PAH_INDTBB_BSS_1.scr
- Test Applet: API_2_PAH_INDTBB_BSS_1.java
- Installation parameter: API_2_PAH_INDTBB_BSS.install (Same as default applet)
- Load Script: API_2_PAH_INDTBB_BSS.ldr
- Conversion parameter: API_2_PAH_INDTBB_BSS.cnv

6.1.6.3.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**NULL as parameter to buffer** **`buffer = NULL`**	`NullPointerException` is thrown	

2	**`offset ≥ buffer.length`**
	`buffer = "Text"`
	`offset = 4`
		`ArrayIndexOutOfBoundsException` is thrown	

3	**`offset < 0`**
	`buffer = "Text"`
	`offset = -1`
		`ArrayIndexOutOfBoundsException` is thrown	

4	**`length > buffer.length`**
	`buffer = "Text"`
	`offset = 0`
	`length = 5`
		`ArrayIndexOutOfBoundsException` is thrown	

5	**`offset + length > buffer.length`**
	`buffer = "Text"`
	`offset = 3`
	`length = 2`
		`ArrayIndexOutOfBoundsException` is thrown	

6	**`length < 0`** `buffer = "Text"`	`ArrayIndexOutOfBoundsException` is thrown	

	offset = 3 length = -1		
7	**Successful call, buffer is the whole buffer** qualifier = 0 dcs = 4 buffer = “TextA” offset = 0 **length = 5**		
	Verify the command number value		
8	**Send the command**		DISPLAY TEXT Pro command **qualifier = 00h** **dcs = 4** Text = “TextA”
9	**Succesfull call, buffer is part of a buffer** **Send the command** qualifier = 0 dcs = 4 buffer = “12TextB” offset = 2 length = 5		**DISPLAY TEXT Pro command** **qualifier = 00h** **dcs = 4** **Text = “TextB”**
10	**Succesfull call, buffer is part of a buffer** **Send the command** qualifier = 0 dcs = 4 buffer = “TextC12” offset = 0 length = 5		**DISPLAY TEXT Pro command** **qualifier = 00h** **dcs = 4** **Text = “TextC”**
11	**Succesfull call, buffer is part of a buffer** **Send the command** qualifier = 0 dcs = 4 buffer = “12TextD34” offset = 2 length = 5		**DISPLAY TEXT Pro command** **qualifier = 00h** **dcs = 4** **Text = “TextD”**
12	**Succesfull call, qualifier = 81h**		**DISPLAY TEXT Pro command**

	Send the command qualifier = 81h dcs = 4 buffer = “TextE” offset = 0 length = 5		**command** **qualifier = 81h** **dcs = 4** **Text = “TextE”**
13	**Succesfull call, DCS=0 (7 bits)** **Send the command** qualifier = 0 dcs = 0 buffer = “TextF” offset = 0 length = 5		**DISPLAY TEXT Pro command** **qualifier = 00h** **dcs = 0** **Text = “TextF”**
14	**Succesfull call, DCS=8 (UCS2)** **Send the command** qualifier = 0 dcs = 8 buffer = “TextG” offset = 0 length = 5		**DISPLAY TEXT Pro command** **qualifier = 00h** **dcs = 8** **Text = “TextG”**
15	**Call the `init()` method with any value** **Then build and send a DISPLAY TEXT command** qualifier = 0 dcs = 4 buffer = “TextHTextH” offset = 0 length = 10		**DISPLAY TEXT Pro command** **qualifier = 00h** **dcs = 4** **Text = “TextHTextH”**
16	**Successful call, text length is null** **Send the command** qualifier = 0 dcs = 4 buffer = “” (not null buffer) offset = 0 length = 0		**DISPLAY TEXT Pro command** **qualifier = 00h** **Text String TLV =** 8D 00
17	**Select a TLV in the ProactiveHandler** **Call the initDisplayText() method**	`UNAVAILABLE_ELEMENT ToolkitException` is thrown by `getValueLength()`	

	Call the getValueLength() method		
18	**Successful call, buffer length = 7Eh** qualifier = 0 dcs = 4 buffer = “UUU…” offset = 0 length = 7Eh		DISPLAY TEXT Pro command **Text String TLV =** 8D 7F 04 55 55…
19	**Successful call, buffer length = 7Fh** qualifier = 0 dcs = 4 buffer = “UUU…” offset = 0 length = 7Fh		DISPLAY TEXT Pro command **Text String TLV =** 8D 81 80 04 55 55…
20	**Successful call, buffer length = 240** Qualifier = 0 dcs = 4 buffer = “UUU…” offset = 0 length = 240		DISPLAY TEXT Pro command **Text String TLV =** 8D 81 F1 04 55 55…
21	**Call the initDisplayText() method with a too long buffer** qualifier = 0 dcs = 4 buffer = “XXXX…” **offset = 0** **length = 241**	HANDLER_OVERFLOW ToolkitException is thrown	
22	**Call the initDisplayText() without sending the command**		

6.1.6.3.5 Test Coverage

CRR number	Test case number
N1	**8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20**
N2	**15**
N3	**17**

N4	**22**
N5	**7**
P1	**1**
P2	**2, 3, 4, 5, 6**
C1	**21**

6.1.6.4 Method initGetInkey

6.1.6.4.1 Test Area Reference: API_2_PAH_INGKBB_BSS

6.1.6.4.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public void initGetInkey(byte qualifier,
                         byte dcs,
                         byte[] buffer,
                         short offset,
                         short length)
                  throws java.lang.NullPointerException,
                         java.lang.ArrayIndexOutOfBoundsException,
                         ToolkitException
```

Normal Execution

CRRN1: The method shall build a GET INKEY proactive command in the ProactiveHandler, using `qualifier`, `dcs` and `buffer` parameters. Comprehension Required flags are set.

CRRN2: A call to this method clears the handler then initialises it.

CRRN3: No TLV is selected after a call to the method.

CRRN4: The GET INKEY command is not sent by the method.

CRRN5: The Command Number may take any value between 01h and FEh.

Parameter Error

CRRP1: The method shall throw `NullPointerException` if `buffer` is `null`.

CRRP2: If `offset` or `length` or both would cause access outside array bounds, a `ArrayIndexOutOfBoundsException` shall be thrown.

Context Error

CRRC1: A `ToolkitException.HANDLER_OVERFLOW` shall be thrown if the ProactiveHandler is to small to put the requested data.

6.1.6.4.3 Test Suite files

- Test Script: API_2_PAH_INGKBB_BSS_1.scr
- Test Applet: API_2_PAH_INGKBB_BSS_1.java
- Installation parameter: API_2_PAH_INGKBB_BSS.install
- Load Script: API_2_PAH_INGKBB_BSS.ldr
- Conversion parameter: API_2_PAH_INGKBB_BSS.cnv

6.1.6.4.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**NULL as parameter to buffer** **buffer = NULL**	NullPointerException is thrown	
2	**offset ≥ buffer.length** buffer = “Text” offset = 4	ArrayIndexOutOfBoundsException is thrown	
3	**offset < 0** buffer = “Text” offset = -1	ArrayIndexOutOfBoundsException is thrown	
4	**length > buffer.length** buffer = “Text” offset = 0 length = 5	ArrayIndexOutOfBoundsException is thrown	
5	**offset + length > buffer.length** buffer = “Text” offset = 3 length = 2	ArrayIndexOutOfBoundsException is thrown	
6	**length < 0** buffer = “Text” offset = 3 length = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Successful call, buffer is the whole buffer** qualifier = 0 dcs = 4 buffer = “TextA” offset = 0 length = 5		
	Verify the command number value		
8	**Send the command**		GET INKEY Pro command **qualifier = 00h** **dcs = 4** Text = “TextA”
9	**Succesfull call, buffer is part of a buffer** **Send the command** qualifier = 0 dcs = 4		**GET INKEY Pro command** **qualifier = 00h** **dcs = 4** **Text = “TextB”**

	buffer = “12TextB” offset = 2 length = 5		
10	**Succesfull call, buffer is part of a buffer** **Send the command** qualifier = 0 dcs = 4 buffer = “TextC12” offset = 0 length = 5		**GET INKEY Pro command** **qualifier = 00h** **dcs = 4** **Text = “TextC”**
11	**Succesfull call, buffer is part of a buffer** **Send the command** qualifier = 0 dcs = 4 buffer = “12TextD34” offset = 2 length = 5		**GET INKEY Pro command** **qualifier = 00h** **dcs = 4** **Text = “TextD”**
12	**Succesfull call, qualifier = 81h** qualifier = 81h dcs = 4 buffer = “TextE” offset = 0 length = 5		**GET INKEY Pro command** **qualifier = 81h** **dcs = 4** **Text = “TextE”**
13	**Succesfull call, DCS=0 (7 bits)** qualifier = 0 dcs = 0 buffer = “TextF” offset = 0 length = 5		**GET INKEY Pro command** **qualifier = 00h** **dcs = 0** **Text = “TextF”**
14	**Succesfull call, DCS=8 (UCS2)** qualifier = 0 dcs = 8 buffer = “TextG” offset = 0 length = 5		**GET INKEY Pro command** **qualifier = 00h** **dcs = 8** **Text = “TextG”**
15	**Call the `init()` method with any value** **Then build and send a GET INKEY command**		**GET INKEY Pro command**

	qualifier = 0 dcs = 4 buffer = “TextHTextH” offset = 0 length = 10		**qualifier = 00h** **dcs = 4** **Text =** **“TextHTextH”**
16	**Successful call, text length is null** **Send the command** qualifier = 0 dcs = 4 buffer = “” offset = 0 length = 0		**GET INKEY Pro command** **qualifier = 00h** **Text String TLV =** 8D 00
17	**Select a TLV in the ProactiveHandler** **Call the initGetInkey() method** **Call the getValueLength() method**	UNAVAILABLE_ELEMENT ToolkitException is thrown by getValueLength()	
18	**Successful call, buffer length = 7Eh** qualifier = 0 dcs = 4 buffer = “UUU…” offset = 0 length = 7Eh		GET INKEY Pro command **Text String TLV =** 8D 7F 04 55 55…
19	**Successful call, buffer length = 7Fh** qualifier = 0 dcs = 4 buffer = “UUU…” offset = 0 length = 7Fh		GET INKEY Pro command **Text String TLV =** 8D 81 80 04 55 55…
20	**Successful call, buffer length = 240** Qualifier = 0 dcs = 4 buffer = “UUU…” offset = 0		GET INKEY Pro command **Text String TLV =** 8D 81 F1 04 55 55…

	length = 240		
21	**Call the initGetInkey() method with a too long buffer** qualifier = 0 dcs = 4 buffer = "XXXX…" **offset = 0** **length = 241**	HANDLER_OVERFLOW ToolkitException is thrown	
22	**Call the initGetInkey() without sending the command**		

6.1.6.4.5 Test Coverage

CRR number	Test case number
N1	**8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20**
N2	**15**
N3	**17**
N4	**22**
N5	**7**
P1	**1**
P2	**2, 3, 4, 5, 6**
C1	**21**

6.1.6.5 Method initGetInput

6.1.6.5.1 Test Area Reference : API_2_PAH_INGPBB_BSSSS

6.1.6.5.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public void initGetInput(byte qualifier,
                         byte dcs,
                         byte[] buffer,
                         short offset,
                         short length,
                         short minRespLength,
                         short maxRespLength)
                  throws java.lang.NullPointerException,
                         java.lang.ArrayIndexOutOfBoundsException,
                         ToolkitException
```

Normal Execution

CRRN1: The method shall build a GET INPUT proactive command in the ProactiveHandler, using `qualifier`, `dcs`, `buffer, minRespLength` and `maxRespLength` parameters. Comprehension Required flags are set.

CRRN2: A call to this method clears the handler then initialises it.

CRRN3: No TLV is selected after a call to the method.

CRRN4: The GET INPUT command is not sent by the method.

CRRN5: The Command Number may take any value between 01h and FEh.

Parameter Error

CRRP1: The method shall throw `NullPointerException` if `buffer` is `null`.

CRRP2: If `offset` or `length` or both would cause access outside array bounds, a `ArrayIndexOutOfBoundsException` shall be thrown.

Context Error

CRRC1: A `ToolkitException.HANDLER_OVERFLOW` shall be thrown if the ProactiveHandler is to small to put the requested data.

6.1.6.5.3 Test Suite files

- Test Script: API_2_PAH_INGPBB_BSSSS_1.scr
- Test Applet: API_2_PAH_INGPBB_BSSSS_1.java
- Installation parameter: API_2_PAH_INGPBB_BSSSS.install
- Load Script: API_2_PAH_INGPBB_BSSSS.ldr
- Conversion parameter: API_2_PAH_INGPBB_BSSSS.cnv

6.1.6.5.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**NULL as parameter to buffer** **`buffer = NULL`**	`NullPointerException` is thrown	

2	**`offset ≥ buffer.length`**
	`buffer = "Text"`
	`offset = 4`
		`ArrayIndexOutOfBoundsException` is thrown	

3	**`offset < 0`**
	`buffer = "Text"`
	`offset = -1`
		`ArrayIndexOutOfBoundsException` is thrown	

4	**`length > buffer.length`**
	`buffer = "Text"`
	`offset = 0`
	`length = 5`
		`ArrayIndexOutOfBoundsException` is thrown	

5	**`offset + length > buffer.length`**
	`buffer = "Text"`
	`offset = 3`
	`length = 2`
		`ArrayIndexOutOfBoundsException` is thrown	

6	**`length < 0`**
	`buffer = "Text"`
	`offset = 3`
		`ArrayIndexOutOfBoundsException` is thrown	

	length = -1		
7	**Successful call, buffer is the whole buffer** qualifier = 0 dcs = 4 buffer = “TextA” offset = 0 length = 5 minRespLength = 00h maxRespLength = FFh		
	Verify the command number value		
8	**Send the command**		GET INPUT Pro command **qualifier = 00h** **dcs = 4** **Text = “TextA”** **Min Length = 00h** Max Length = FFh
9	**Succesfull call, buffer is part of a buffer** **Send the command** qualifier = 0 dcs = 4 buffer = “12TextB” offset = 2 length = 5 minRespLength = 10h maxRespLength = FFh		**GET INPUT Pro command** **qualifier = 00h** **dcs = 4** **Text = “TextB”** **Min Length = 10h** **Max Length = FFh**
10	**Succesfull call, buffer is part of a buffer** **Send the command** qualifier = 0 dcs = 4 buffer = “TextC12” offset = 0 length = 5 minRespLength = FFh maxRespLength = FFh		**GET INPUT Pro command** **qualifier = 00h** **dcs = 4** **Text = “TextC”** **Min Length = FFh** **Max Length = FFh**
11	**Succesfull call, buffer is part of a buffer**		**GET INPUT Pro command**

	Send the command qualifier = 0 dcs = 4 buffer = “12TextD34” offset = 2 length = 5 minRespLength = 00h maxRespLength = 00h		**qualifier = 00h** **dcs = 4** **Text = “TextD”** **Min Length = 00h** **Max Length = 00h**
12	**Succesfull call, qualifier = 81h** qualifier = 81h dcs = 4 buffer = “TextE” offset = 0 length = 5 minRespLength = 00h maxRespLength = 10h		**GET INPUT Pro command** **qualifier = 81h** **dcs = 4** **Text = “TextE”** **Min Length = 00h** **Max Length = 10h**
13	**Succesfull call, DCS=0 (7 bits)** qualifier = 0 dcs = 0 buffer = “TextF” offset = 0 length = 5 minRespLength = 10h maxRespLength = 10h		**GET INPUT Pro command** **qualifier = 00h** **dcs = 0** **Text = “TextF”** **Min Length = 10h** **Max Length = 10h**
14	**Succesfull call, DCS=8 (UCS2)** qualifier = 0 dcs = 8 buffer = “TextG” offset = 0 length = 5 minRespLength = 00h maxRespLength = FFh		**GET INPUT Pro command** **qualifier = 00h** **dcs = 8** **Text = “TextG”** **Min Length = 00h** **Max Length = FFh**
15	**Call the `init()` method with any value** **Then build and send a GET INPUT command** qualifier = 0 dcs = 4 buffer = “TextHTextH” offset = 0		**GET INPUT Pro command** **qualifier = 00h** **dcs = 4** **Text = “TextHTextH”** **Min Length = 00h** **Max Length = 10h**

	length = 10 minRespLength = 00h maxRespLength = 10h		
16	**Successful call, text length is null** **Send the command** **qualifier = 0** dcs = 4 buffer = “” offset = 0 length = 0 minRespLength = 00h maxRespLength = 10h		**GET INPUT Pro command** **qualifier = 00h** **Text String TLV =** 8D 00 **Min Length = 00h** **Max Length = 10h**
17	**Select a TLV in the ProactiveHandler** **Call the initGetInput() method** **Call the getValueLength() method**	UNAVAILABLE_ELEMENT ToolkitException is thrown by getValueLength()	
18	**Successful call, buffer length = 7Eh** qualifier = 0 dcs = 4 buffer = “UUU…” offset = 0 length = 7Eh minRespLength = 00h maxRespLength = 10h		GET INPUT Pro command **Text String TLV =** 8D 7F 04 55 55… **Min Length = 00h** **Max Length = 10h**
19	**Successful call, buffer length = 7Fh** qualifier = 0 dcs = 4 buffer = “UUU…” offset = 0 length = 7Fh minRespLength = 00h maxRespLength = 10h		GET INPUT Pro command **Text String TLV =** 8D 81 80 04 55 55… **Min Length = 00h** **Max Length = 10h**
20	**Successful call, buffer length = 236**		GET INPUT Pro command **Text String TLV =** 8D 81 ED 04 55 55…

	Qualifier = 0 dcs = 4 buffer = “UUU…” offset = 0 length = 236 minRespLength = 00h maxRespLength = 10h		
21	**Call the initGetInput() method with a too long buffer** qualifier = 0 dcs = 4 buffer = “XXXX…” **offset = 0** length = 237 minRespLength = 00h **maxRespLength = 10h**	HANDLER_OVERFLOW ToolkitException is thrown	
22	**Call the initGetInput() without sending the command**		

6.1.6.5.5 Test Coverage

CRR number	Test case number
N1	8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 20
N2	15
N3	17
N4	22
N5	7
P1	1
P2	2, 3, 4, 5, 6
C1	21

6.1.6.6 Method send

6.1.6.6.1 Test Area Reference: API_2_PAH_SEND

6.1.6.6.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte send()
```

Normal Execution

CRRN1: The `send()` method send the current proactive command to the mobile.

CRRN2: The returned byte is equal to general result of the command (first byte of Result TLV in Terminal Response).

CRRN3: The handler remains unchanged after a call to send() method until the use of initXX() or appendTLV().

CRRN4: There is no invocation of select() or deselect() method.

CRRN5: A pending toolkit applet transaction at the method invocation is aborted.

Parameter Error

No requirements

Context Error

No requirements

6.1.6.6.3 Test Suite files

- Test Script: API_2_PAH_SEND_1.scr
- Test Applet: API_2_PAH_SEND_1.java
- Installation parameter: API_2_PAH_SEND.install
- Load Script: API_2_PAH_SEND.ldr
- Conversion parameter: API_2_PAH_SEND.cnv

6.1.6.6.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	**Build and send a DISPLAY TEXT command**
	`qualifier = 00h`
	`dcs = 04h`
	`buffer = 'Text'`
			DISPLAY TEXT Pro command

2	**Terminal Response with General Result = 00** **`Result TLV = 03 01 00` (command performed successfully)**	Result of `send()` is **00h**	

3	**Build and send a DISPLAY TEXT command**
	`qualifier = 00h`
	`dcs = 04h`
	`buffer = 'Text'`
			DISPLAY TEXT Pro command

4	**Terminal Response with General Result = 01, without Additional information on result** **`Result TLV = 03 01 01` (command performed with partial comprehension)**	Result of `send()` is **01h**	

5	**Build and send a DISPLAY TEXT command**
	`qualifier = 00h`
	`dcs = 04h`
	`buffer = 'Text'`
			DISPLAY TEXT Pro command

6	**Terminal Response with General Result = 01, with Additional information on result** **`Result TLV = 03 02 01 55` (command performed with partial comprehension)**	Result of `send()` is **01h**	
7	**Build and send a DISPLAY TEXT command** `qualifier = 00h`		**DISPLAY TEXT Pro command**

	dcs = 04h **buffer = 'Text'**		
8	**Terminal Response with General Result = 02** **Result TLV = 03 04** 02 65 43 21 **(Missing information)**	Result of send() is **02h**	
9	**Build and send a 7Fh byte command (DISPLAY TEXT)** qualifier = 00h dcs = 04h **buffer = "UUUUU…"** **length =** 73h		**DISPLAY TEXT Pro command** **BER-TLV = D0 7F** Text String TLV = 8D 74 04 55 55 55…
10	**Build and send a 80h byte command (DISPLAY TEXT)** qualifier = 00h dcs = 04h **buffer = "UUUUU…"** **length =** 74h		**DISPLAY TEXT Pro command** **BER-TLV = D0 81 80** Text String TLV = 8D 75 04 55 55 55…
11	**Build and send a maximum length command (length of the handler should be 253)** **DISPLAY TEXT :** Qualifier = 0 dcs = 4 buffer = "UUU…" offset = 0 **length = 240**		DISPLAY TEXT Pro command **BER-TLV = D0 81 FD** **Text String TLV =** **8D 81 F1 04 55 55…**
12	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Pro command
	Verify ProactiveHandler was not modified		
13	**Build and send a DISPLAY TEXT command** Verify there is no invocation of select() or deselect() method.		DISPLAY TEXT Pro command
14	**Build and send a DISPLAY TEXT command**		**DISPLAY TEXT Pro command**
	Terminal Response with 2 Result TLV **1st Result TLV = 03 02** 02 **12** **2nd Result TLV = 03 03** 03 **34 56**	Result of send() is **02h**	
15	**Send an empty buffer (What shall be the behaviour ?)**		
16	**Begin a transaction** JCSystem. beginTransaction()		
	Send a Proactive Command		**Proactive Command**
	Terminal Response		

	Verify there is no pending transaction **`JCSystem.commitTransaction()`**	TransactionException.NOT_IN_PROGRESS is thrown by commitTransaction()	

6.1.6.6.5 Test Coverage

CRR number	Test case number
N1	1, 3, 5, 7, 9, 10, 11, 12, 13, 14
N2	2, 4, 6, 8, 14
N3	12
N4	13
N5	16

6.1.6.7 Method getLength

6.1.6.7.1 Test Area Reference API_2_PAH_GLEN

6.1.6.7.2 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
public short getLength()
                    throws ToolkitException
```

Normal Execution

CRRN1: returns the length in bytes of the TLV list.

Parameter Error

No requirements

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.HANDLER_NOT_AVAILABLE.

6.1.6.7.3 Test Suite files

- Test Script: API_2_PAH_GLEN_1.scr
- Test Applet: API_2_PAH_GLEN_1.java
- Installation parameter: API_2_PAH_GLEN.install
- Load Script: API_2_PAH_GLEN.ldr
- Conversion parameter: API_2_PAH_GLEN.cnv

6.1.6.7.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Clear the handler** **`getLength()`**	Result of `getLength()` is 0	
2	**Call the `init()` method** `getLength()`	Result of `getLength()` is 9	

3	**Call the initDisplayText() method, with buffer length = 240** **getLength()**	Result of `getLength()` is 253	
4	**Build a 7Fh Proactive Handler** **getLength()**	Result of `getLength()` is 7Fh	
5	**Build a 80h Proactive Handler** **getLength()**	Result of `getLength()` is 80h	

6.1.6.7.5 Test Coverage

CRR number	Test case number
N1	1, 2, 3
C1	Does not apply for Proactive Handler

6.1.6.8 Method copy

6.1.6.8.1 Test Area Reference API_2_PAH_COPY_BSS

6.1.6.8.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short copy(byte[] dstBuffer,
                  short dstOffset,
                  short dstLength)
                   throws java.lang.NullPointerException,
                        java.lang.ArrayIndexOutOfBoundsException,
                        ToolkitException
```

Normal Execution

CRRN1: copies the simple TLV list contained in the handler to the destination byte array.

CRRN2: returns dstOffset + dstLength.

Parameter Error

CRRP1: if dstBuffer is null a NullPointerException is thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative, an ArrayIndexOutOfBoundsException is thrown.

CRRP3: if dstLength is grater than the length of the simple TLV List, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException. OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.6.8.3 Test Suite files

- Test Script: API_2_PAH_COPY_BSS_1.scr
- Test Applet: API_2_PAH_COPY_BSS_1.java
- Installation parameter: API_2_PAH_COPY_BSS.install
- Load Script: API_2_PAH_COPY_BSS.ldr
- Conversion parameter: API_2_PAH_COPY_BSS.cnv

6.1.6.8.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**NULL as parameter to dstBuffer**	`NullPointerException` is thrown	
2	Call the `init()` method		
	dstOffset ≥ dstBuffer.length **dstBuffer.length = 5** dstOffset = 5 dstLength = 1	`ArrayIndexOutOfBoundsException` is thrown	
3	**dstOffset < 0** **dstBuffer.length = 5** dstOffset = -1 dstLength = 1	`ArrayIndexOutOfBoundsException` is thrown	
4	**dstLength > dstBuffer.length** **dstBuffer.length = 5** dstOffset = 0 dstLength = 6	`ArrayIndexOutOfBoundsException` is thrown	
5	**dstOffset + dstLength > dstBuffer.length** **dstBuffer.length = 5** dstOffset = 3 dstLength = 3	`ArrayIndexOutOfBoundsException` is thrown	
6	**dstLength < 0** **dstBuffer.length = 5** dstOffset = 0 dstLength = -1	`ArrayIndexOutOfBoundsException` is thrown	
7	**dstLength > length of the simple TLV list** **dstBuffer.length = 10** dstOffset = 0 dstLength = 10	`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown	
8	**Successful call, dstBuffer is the whole buffer** **dstBuffer.length = 9** dstOffset = 0	Result of `copy()` is 9	

	dstLength = 9		
9	**Compare the buffer**	Result of `arrayCompare()` is 0	
10	**Successful call, dstBuffer is part of a buffer** **dstBuffer.length = 15** dstOffset = 3 dstLength = 9	Result of `copy()` is 12	
11	**Compare the whole buffer**	Result of `arrayCompare()` is 0	
12	**Successful call, dstBuffer is part of a buffer** **dstBuffer.length = 15** dstOffset = 3 dstLength = 6	Result of `copy()` is 9	
13	**Compare the whole buffer**	Result of `arrayCompare()` is 0	

6.1.6.8.5 Test Coverage

CRR number	Test case number
N1	**9, 11, 13**
N2	**8, 10, 12**
P1	**1**
P2	**2, 3, 4, 5, 6**
P3	**7**
C1	**Does not apply for Proactive Handler**

6.1.6.9 Method findTLV

6.1.6.9.1 Test Area Reference API_2_PAH_FINDBB

6.1.6.9.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte findTLV(byte tag, byte occurrence)
              throws ToolkitException
```

Normal Execution

Looks for the indicated occurrence of a TLV element from the beginning of the TLV list (handler buffer):

CRRN1: the method is successful if the required occurrence exists then the corresponding TLV becomes current.

CRRN2: if the method is successful then it returns TLV_FOUND_CR_SET when Comprehension Required flag is set.

CRRN3: if the method is successful then it returns TLV_FOUND_CR_NOT_SET when Comprehension Required flag is not set.

CRRN4: if the required occurrence of the TLV element does not exist, no TLV is selected and TLV_NOT_FOUND is returned.

CRRN5: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if an input parameter is not valid (e.g. occurrence = 0) an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException BAD_INPUT_PARAMETER. The current TLV is no longer defined.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.6.9.3 Test Suite files

- Test Script: API_2_PAH_FINDBB_1.scr
- Test Applet: API_2_PAH_FINDBB_1.java
- Installation parameter: API_2_PAH_FINDBB.install
- Load Script: API_2_PAH_FINDBB.ldr
- Conversion parameter: API_2_PAH_FINDBB.cnv

6.1.6.9.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialise the handler**		
	Invalid input parameter **`Occurrence = 0`**	ToolkitException.BAD_INPUT_PARAMETER is thrown	
2	**Call the `init()` method**		
	Search 1st TLV **`Tag = 01h`** `Occurrence = 1`	**Result is `TLV_FOUND_CR_SET`**	
3	**Call the `getValueLength()` method**	**Result is 03h**	
4	**Search 2nd TLV** **`Tag = 02h`** `Occurrence = 1`	**Result is `TLV_FOUND_CR_SET`**	
5	**Call the `getValueLength()` method**	**Result is 02h**	
6	**Search a wrong tag** **`Tag = 03h`** `Occurrence = 1`	**Result is `TLV_NOT_FOUND`**	
7	**Call the `getValueLength()` method**	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
8	**Search a tag with wrong occurrence** **`Tag = 01h`** `Occurrence = 2`	**Result is `TLV_NOT_FOUND`**	
9	**Call the `getValueLength()` method**	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
10	**Append a TLV with tag=02h**		
	Search the TLV	**Result is `TLV_FOUND_CR_NOT_SET`**	

	Tag = 02h Occurrence = 2		
11	**Append a TLV with tag=04h**		
	Search the TLV **Tag = 04h** Occurrence = 1	**Result is TLV_FOUND_CR_NOT_SET**	
12	**Search tag 81h** **Tag = 81h** Occurrence = 1	**Result is TLV_FOUND_CR_SET**	
13	**Search tag 84h** **Tag = 84h** Occurrence = 1	**Result is TLV_FOUND_CR_NOT_SET**	

6.1.6.9.5 Test Coverage

CRR number	Test case number
N1	3, 5
N2	2, 4
N3	10, 11
N4	6, 7,8, 9
N5	12, 13
P1	1
C1	Does not apply for Proactive Handler

6.1.6.10 Method getValueLength

6.1.6.10.1 Test Area Reference API_2_PAH_GVLEN

6.1.6.10.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short getValueLength()
                        throws ToolkitException
```

Normal Execution

CRRN1: gets and returns the binary length of the value field for the last TLV element which has been found in the handler.

Parameter Error

No requirements

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.6.10.3 Test Suite files

- Test Script: API_2_PAH_GVLEN_1.scr

- Test Applet: API_2_PAH_GVLEN_1.java
- Installation parameter: API_2_PAH_GVLEN.install
- Load Script: API_2_PAH_GVLEN.ldr
- Conversion parameter: API_2_PAH_GVLEN.cnv

6.1.6.10.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the `init()` method**		
	getValueLength()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	**Call the `initDisplayText()` method** **`length = 0`**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `00h`**	
3	**Call the `initDisplayText()` method** **`length = 1 (+ dcs byte)`**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `02h`**	
4	**Call the `initDisplayText()` method** **`length = 7Eh (+ dcs byte)`**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `7Fh`**	
5	**Call the `initDisplayText()` method** **`length = 7Fh (+ dcs byte)`**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `80h`**	
6	**Call the `initDisplayText()` method** **`length = F0h` (maximum text length)**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `F1h`**	

6.1.6.10.5 Test Coverage

CRR number	Test case number
N1	**2, 3, 4, 5, 6**
C1	**Does not apply for Proactive Handler**

C2	1

6.1.6.11 Method getValueByte

6.1.6.11.1 Test Area Reference API_2_PAH_GVBYTS

6.1.6.11.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte getValueByte(short valueOffset)
                    throws ToolkitException
```

Normal Execution

CRRN1: Gets a byte from the last TLV element which has been found in the handler and returns its value (1 byte).

Parameter Error

CRRP1: if valueOffset is out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.6.11.3 Test Suite files

- Test Script: API_2_PAH_GVBYTS_1.scr
- Test Applet: API_2_PAH_GVBYTS_1.java
- Installation parameter: API_2_PAH_GVBYTS.install
- Load Script: API_2_PAH_GVBYTS.ldr
- Conversion parameter: API_2_PAH_GVBYTS.cnv

6.1.6.11.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	**Call the `init()` method**
	`type = FFh`
	`qualifier = FEh`
	`destination = FDh`		

	getValueByte(0)	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	Search TLV `01h` (Command Details TLV)		
	getValueByte(3)	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
3	Search TLV `01h` (Command Details TLV)		
	getValueByte(2)	**Result is FEh (qualifier)**	

4	Search TLV 02h (Device Identities TLV)		
	getValueByte(0)	**Result is 81h (Source)**	
5	initDisplayText() **buffer = 00 01 … 7D** **length = 7Eh** Search TLV 0Dh (Text String TLV)		
	getValueByte(7E)	**Result is 7Dh**	
6	initDisplayText() **buffer = 00 01 … 7D 7E** **length = 7Fh** Search TLV 0Dh (Text String TLV)		
	getValueByte(7E)	**Result is 7Dh**	
7	getValueByte(7F)	**Result is 7Eh**	
8	initDisplayText() **buffer = 00 01 … EF** **length = F0h** Search TLV 0Dh (Text String TLV)		
	getValueByte(F0)	**Result is EFh**	

6.1.6.11.5 Test Coverage

CRR number	Test case number
N1	3, 4, 5, 6, 7, 8
P1	2
C1	Does not apply for Proactive Handler
C2	1

6.1.6.12 Method copyValue

6.1.6.12.1 Test Area Reference API_2_PAH_CPYVS_BSS

6.1.6.12.2 Conformance Requirement

The method with following prototype shall be compliant with its definition in the API.

```
public short copyValue(short valueOffset,
                       byte[] dstBuffer,
                       short dstOffset,
                       short dstLength)
                throws java.lang.NullPointerException,
                       java.lang.ArrayIndexOutOfBoundsException,
                       ToolkitException
```

Normal Execution

CRRN1: copies a part of the last TLV element which has been found, into a destination. buffer.

CRRN2: returns dstOffset + dstLength.

Parameter Error

CRRP1: if dstBuffer is null NullPointerException is thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative ArrayIndexOutOfBoundsException is thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.6.12.3 Test Suite files

- Test Script: API_2_PAH_CPYVS_BSS _1.scr
- Test Applet: API_2_PAH_GVBYTS_1.java
- Installation parameter: API_2_PAH_GVBYTS.install
- Load Script: API_2_PAH_GVBYTS.ldr
- Conversion parameter: API_2_PAH_GVBYTS.cnv

6.1.6.12.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler Select a TLV		
	copyValue() with a null dstBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15 Select Text String TLV		
	dstOffset ≥ dstBuffer.length **`dstBuffer.length = 5`** **`dstOffset = 5`** **`dstLength = 1`**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	dstOffset < 0 **`dstBuffer.length = 5`** **`dstOffset = -1`** **`dstLength = 1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	dstLength >dstBuffer.length **`dstBuffer.length = 5`** **`dstOffset = 0`**	`ArrayIndexOutOfBoundsException` is thrown	

	dstLength = 6		
5	dstOffset + dstLength >dstBuffer.length **dstBuffer.length = 5** **dstOffset = 3** **dstLength = 3**	`ArrayIndexOutOfBoundsException` is thrown	
6	dstLength < 0 **dstBuffer.length = 5** **dstOffset = 0** **dstLength = -1**	`ArrayIndexOutOfBoundsException` is thrown	
7	initDisplayText() with length = 5 Select Text String TLV		
	valueOffset ≥ Text String Length **valueOffset = 6** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	
8	valueOffset < 0 **valueOffset = -1** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	
9	dstLength > Text String length **valueOffset = 0** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 7**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	
10	valueOffset + dstLength > Text String length **valueOffset = 2** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 5**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	
11	Initialise the handler		
	copyValue()	`ToolkitException.UNAVAILABLE_ELEMENT` is thrown	
12	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F** Select Text String TLV		
	Successful call **valueOffset = 0** **dstBuffer.length = 17**	**Result of `copyValue()` is 17**	

	dstOffset = 0 **dstLength = 17**		
13	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
14	initialise dstBuffer **dstBuffer = 55 55 … 55**		
	Successful call **valueOffset = 2** **dstBuffer.length = 20** **dstOffset = 3** **dstLength = 12**	**Result of copyValue() is 15**	
15	Compare buffer **buffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**	Result is 00h	

6.1.6.12.5 Test Coverage

CRR number	Test case number
N1	13, 15
N2	12, 14
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for Proactive Handler
C2	11

6.1.6.13 Method compareValue

6.1.6.13.1 Test Area Reference API_2_PAH_CPRVS_BSS

6.1.6.13.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte compareValue(short valueOffset,
                         byte[] compareBuffer,
                         short compareOffset,
                         short compareLength)
                  throws java.lang.NullPointerException,
                         java.lang.ArrayIndexOutOfBoundsException,
                         ToolkitException
```

Normal Execution

Compares the last found TLV element with a buffer:

CRRN1: returns 0 if identical.

CRRN2: returns -1 if the first miscomparing byte in simple TLV List is less than that in compareBuffer.

CRRN3: returns 1 if the first miscomparing byte in simple TLV List is greater than that in compareBuffer.

Parameter Error

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset or compareLength or both would cause access outside array bounds, or if compareLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.6.13.3 Test Suite files

- Test Script: API_2_PAH_CPRVS_BSS_1.scr
- Test Applet: API_2_PAH_CPRVS_BSS_1.java
- Installation parameter: API_2_PAH_CPRVS_BSS.install
- Load Script: API_2_PAH_CPRVS_BSS.ldr
- Conversion parameter: API_2_PAH_CPRVS_BSS.cnv

6.1.6.13.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler Select a TLV		
	compareValue() with a null compareBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15 Select Text String TLV		
	compareOffset ≥ compareBuffer.length **`compareBuffer.length = 5`** **`compareOffset = 5`** **`compareLength = 1`**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	compareOffset < 0 **`compareBuffer.length = 5`** **`compareOffset = -1`** **`compareLength = 1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	compareLength >compareBuffer.length	`ArrayIndexOutOfBoundsException` is thrown	

	compareBuffer.length = 5 **compareOffset = 0** **compareLength = 6**	xception is thrown	
5	compareOffset + compareLength >compareBuffer.length **compareBuffer.length = 5** **compareOffset = 3** **compareLength = 3**	ArrayIndexOutOfBoundsException is thrown	
6	compareLength < 0 **compareBuffer.length = 5** **compareOffset = 0** **compareLength = -1**	ArrayIndexOutOfBoundsException is thrown	
7	initDisplayText() with length = 5 Select Text String TLV		
	valueOffset ≥ Text String Length **valueOffset = 6** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
8	valueOffset < 0 **valueOffset = -1** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
9	compareLength > Text String length **valueOffset = 0** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 7**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
10	valueOffset + compareLength > Text String length **valueOffset = 2** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 5**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
11	Initialise the handler		
	compareValue()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
12	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F** Select Text String TLV		
	Initialise compareBuffer		

	compareBuffer = **04 00 01 … 0F**		
	Compare buffers **valueOffset = 0** **compareOffset = 0** **compareLength = 17**	**Result is 00h**	
13	Initialise compareBuffer **compareBuffer =** **04 00 01 02 03** **04 05 06 07 08** **05 0A 0B 0C 0D** **0E** 10		
	Compare buffers with same parameters	**Result is -1**	
14	Initialise compareBuffer **compareBuffer =** 03 **00 01 … 0F**		
	Compare buffers with same parameters	**Result is +1**	
15	Initialise compareBuffer **compareBuffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**		
	Compare buffers **valueOffset = 2** **compareOffset = 3** **compareLength = 12**	**Result is 00h**	
16	Initialise compareBuffer **compareBuffer =** **55 55 55** 02 01 **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is -1**	
17	Initialise compareBuffer **compareBuffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A** 0A 0D		

	55 55 55 55 55		
	Compare buffers with same parameters	**Result is +1**	

6.1.6.13.5 Test Coverage

CRR number	Test case number
N1	**12, 15**
N2	**13, 16**
N3	**14, 17**
P1	**1**
P2	**2, 3, 4, 5, 6**
P3	**7, 8, 9, 10**
C1	**Does not apply for Proactive Handler**
C2	**11**

6.1.6.14 Method findAndCopyValue

6.1.6.14.1 Test Area Reference API_2_PAH_FACYB_BS

6.1.6.14.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short findAndCopyValue(byte tag,
                              byte[] dstBuffer,
                            short dstOffset)
                throws     java.lang.NullPointerException,
                           java.lang.ArrayIndexOutOfBoundsException,
                              ToolkitException
```

Normal Execution

CRRN1: looks for the first occurrence of a TLV element from the beginning of a TLV list and copy its value into a destination buffer.

CRRN2: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN3: if the method is successful then the corresponding TLV becomes current and dstOffset + length of the copied value is returned.

CRRN4: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if dstBuffer is null NullPointerException shall be thrown.

CRRP2: if dstOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.6.14.3 Test Suite files

- Test Script: API_2_PAH_FACYB_BS_1.scr
- Test Applet: API_2_PAH_FACYB_BS_1.java
- Installation parameter: API_2_PAH_FACYB_BS.install
- Load Script: API_2_PAH_FACYB_BS.ldr
- Conversion parameter: API_2_PAH_FACYB_BS.cnv

6.1.6.14.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler		
	FindAndCopyValue() with a null dstBuffer	`NullPointerException` is thrown	
2	InitDisplayText() with length = 15		
	dstOffset ≥ dstBuffer.length **`tag = 0Dh`** **`dstBuffer.length = 20`** **`dstOffset = 20`**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	dstOffset < 0 **`dstBuffer.length = 20`** **`dstOffset = -1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	length > dstBuffer.length **`dstBuffer.length = 15`** **`dstOffset = 0`**	`ArrayIndexOutOfBoundsException` is thrown	
5	DstOffset + length >dstBuffer.length **`DstBuffer.length = 20`** **`DstOffset = 5`**	`ArrayIndexOutOfBoundsException` is thrown	
6	initDisplayText()		
	Select a TLV (tag 02h)		
	findAndCopyValue() **`tag = 03h`**	`ToolkitException.UNAVAILABLE_ELEMENT` is thrown	
	Call the `getValueLength()` method	**`ToolkitException.UNAVAILABLE_ELEMENT` is thrown.**	
7	initDisplayText() **`dcs = 4`** **`buffer = 00 01 … 0F`**		
	Successful call **`Tag = 0Dh`** **`DstBuffer.length = 17`** **`DstOffset = 0`**	**Result of `copyValue()` is 17**	
8	Compare buffer **`buffer = 04 00 01 … 0F`**	Result is 00h	

9	initialise dstBuffer **dstBuffer = 55 55 … 55**		
	Successful call **dstBuffer.length = 20** **dstOffset = 2**	**Result of copyValue() is 19**	
10	Compare buffer **buffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**	Result is 00h	
11	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	append a 2nd Text String TLV		
	Successful call **tag = 0Dh** **dstBuffer.length = 17** **dstOffset = 0**	**Result of copyValue() is 17**	
12	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
13	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Successful call (with tag 8Dh) **tag = 8Dh** **dstBuffer.length = 17** **dstOffset = 0**	**Result of copyValue() is 17**	
14	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
15	Append tag 0Fh **buffer = 00 01 … 0F**		
	Successful call (with tag 8Fh) **tag = 8Fh** **dstBuffer.length = 16** **dstOffset = 0**	**Result of copyValue() is 16**	
16	Compare buffer **buffer = 00 01 … 0F**	Result is 00h	

6.1.6.14.5 Test Coverage

CRR number	Test case number
N1	8, 10, 12
N2	6
N3	7, 9, 11
N4	13, 14, 15, 16
P1	1
P2	2, 3, 4, 5
C1	Does not apply for Proactive Handler

6.1.6.15 Method findAndCopyValue

6.1.6.15.1 Test Area Reference API_2_PAH_FACYBS_BSS

6.1.6.15.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short findAndCopyValue(byte tag,
                              byte occurence,
                             short valueOffset,
                             byte[] dstBuffer,
                             short dstOffset,
                             short dstLength)
                      throws java.lang.NullPointerException,
                            java.lang.ArrayIndexOutOfBoundsException,
                             ToolkitException
```

Normal Execution

CRRN1: looks for the indicated occurrence of a TLV element from the beginning of a TLV list and copy its value into a destination buffer.

CRRN2: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN3: if the method is successful then the corresponding TLV becomes current and dstOffset + dstLength is returned.

CRRN4: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if dstBuffer is null NullPointerException shall be thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.6.15.3 Test Suite files

- Test Script: API_2_PAH_FACYBS_BSS_1.scr
- Test Applet: API_2_PAH_FACYBS_BSS_1.java
- Installation parameter: API_2_PAH_FACYBS_BSS.install
- Load Script: API_2_PAH_FACYBS_BSS.ldr
- Conversion parameter: API_2_PAH_FACYBS_BSS.cnv

6.1.6.15.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler		
	findAndCopyValue() with a null dstBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15		
	dstOffset ≥ dstBuffer.length **`tag = 0Dh, occurrence = 1`** **`valueOffset = 0`** **`dstBuffer.length = 5`** **`dstOffset = 5`** **`dstLength = 1`**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	dstOffset < 0 **`dstBuffer.length = 5`** **`dstOffset = -1`** **`dstLength = 1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	dstLength >dstBuffer.length **`dstBuffer.length = 5`** **`dstOffset = 0`** **`dstLength = 6`**	`ArrayIndexOutOfBoundsException` is thrown	
5	dstOffset + dstLength >dstBuffer.length **`dstBuffer.length = 5`** **`dstOffset = 3`** **`dstLength = 3`**	`ArrayIndexOutOfBoundsException` is thrown	
6	dstLength < 0 **`dstBuffer.length = 5`** **`dstOffset = 0`** **`dstLength = -1`**	`ArrayIndexOutOfBoundsException` is thrown	
7	initDisplayText() with length = 5		
	valueOffset ≥ Text String Length **`tag = 0Dh, occurrence = 1`** **`valueOffset = 6`** **`dstBuffer.length = 15`** **`dstOffset = 0`**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	

	dstLength = 1		
8	valueOffset < 0 **valueOffset = -1** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
9	dstLength > Text String length **valueOffset = 0** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 7**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
10	valueOffset + dstLength > Text String length **valueOffset = 2** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 5**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
11	InitDisplayText()		
	Select a TLV (tag 02h)		
	findAndCopyValue() **tag = 0Dh** **occurrence =** 2	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the `getValueLength()` method	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
12	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Successful call **tag = 0Dh, occurrence = 1** **valueOffset = 0** **dstBuffer.length = 17** **dstOffset = 0** **dstLength = 17**	**Result of `findAndCopyValue()` is 17**	
13	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
14	initialise dstBuffer **dstBuffer = 55 55 … 55**		
	Successful call **tag = 0Dh, occurrence = 1** **valueOffset = 2** **dstBuffer.length = 20** **dstOffset = 3** **dstLength = 12**	**Result of `copyValue()` is 15**	

15	Compare buffer **buffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**	Result is 00h	
16	Append a Text String TLV **tag = 0D** **buffer = 00 11 22 33 44 55 (no specific DCS byte)**		
	Successful call **tag = 0Dh, occurrence = 1** **valueOffset = 0** **dstBuffer.length = 17** **dstOffset = 0** **dstLength = 17**	**Result of `findAndCopyValue()` is 17**	
17	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
18	Successful call **tag = 0Dh, occurrence = 2** **valueOffset = 0** **dstBuffer.length = 6** **dstOffset = 0** **dstLength = 6**	**Result of `findAndCopyValue()` is 6**	
19	Compare buffer **buffer = 00 11 22 33 44 55**	Result is 00h	
20	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Successful call (with tag 8Dh) **tag = 8Dh** **dstBuffer.length = 17** **dstOffset = 0**	**Result of `copyValue()` is 17**	
21	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
22	Append tag 0Fh **buffer = 00 01 … 0F**		
	Successful call (with tag 8Fh) **tag = 8Fh** **dstBuffer.length = 16** **dstOffset = 0**	**Result of `copyValue()` is 16**	

23	Compare buffer **buffer = 00 01 … 0F**	Result is 00h	

6.1.6.15.5 Test Coverage

CRR number	Test case number
N1	**13, 15, 17, 19**
N2	**11**
N3	**12, 14, 16, 18**
N4	**20, 21, 22, 23**
P1	**1**
P2	**2, 3, 4, 5, 6**
P3	**7, 8, 9, 10**
C1	**Does not apply for Proactive Handler**

6.1.6.16 Method findAndCompareValue

6.1.6.16.1 Test Area Reference API_2_PAH_FACRB_BS

6.1.6.16.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte findAndCompareValue(byte tag,
                                byte[] compareBuffer,
                                short compareOffset)
                throws     java.lang.NullPointerException,
                           java.lang.ArrayIndexOutOfBoundsException,
                      ToolkitException
```

Normal Execution

Looks for the first occurrence of a TLV element from beginning of a TLV list and compare its value with a buffer :

CRRN1: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN2: if the method is successful then the corresponding TLV becomes current.

CRRN3: if identical returns 0.

CRRN4: if the first miscomparing byte in simple TLV is less than that in compareBuffer returns -1.

CRRN5: if the first miscomparing byte in simple TLV is greater than that in compareBuffer returns 1.

CRRN6: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.6.16.3 Test Suite files

- Test Script: API_2_PAH_FACRB_BS_1.scr
- Test Applet: API_2_PAH_FACRB_BS_1.java
- Installation parameter: API_2_PAH_FACRB_BS.install
- Load Script: API_2_PAH_FACRB_BS.ldr
- Conversion parameter: API_2_PAH_FACRB_BS.cnv

6.1.6.16.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler		
	findAndCompareValue() with a null dstBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15		
	compareOffset ≥ compareBuffer.length **`tag = 0Dh`** **`compareBuffer.length = 20`** **`compareOffset = 20`**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	compareOffset < 0 **`compareBuffer.length = 20`** **`compareOffset = -1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	length > compareBuffer.length **`compareBuffer.length = 15`** **`compareOffset = 0`**	`ArrayIndexOutOfBoundsException` is thrown	
5	compareOffset + length > compareBuffer.length **`compareBuffer.length = 20`** **`compareOffset = 5`**	`ArrayIndexOutOfBoundsException` is thrown	
6	InitDisplayText()		
	Select a TLV (tag 02h)		
	findAndCompareValue() **`tag = 03h`**	`ToolkitException.UNAVAILABLE_ELEMENT` is thrown	
	Call the `getValueLength()` method	**`ToolkitException.UNAVAILABLE_ELEMENT` is thrown.**	
7	initDisplayText() **`dcs = 4`** **`buffer = 00 01 … 0F`**		
	Initialise compareBuffer **`compareBuffer =`**		

	04 00 01 … 0F		
	Compare buffers **tag = 0Dh** **compareOffset = 0**	**Result is 00h**	
8	Verify current TLV **getValueLength()**	**Result is 17**	
9	Initialise compareBuffer **compareBuffer =** **04 00 01 …** 10		
	Compare buffers with same parameters	**Result is -1**	
10	Initialise compareBuffer **compareBuffer =** 03 **00 01 … 0F**		
	Compare buffers with same parameters	**Result is +1**	
11	Initialise compareBuffer **compareBuffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**		
	Compare buffers **compareOffset = 2**	**Result is 00h**	
12	append a Text String TLV **tag = 0Dh** **buffer = 00 11 22 33 44 55**		
	Initialise compareBuffer **compareBuffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**		
	Compare buffers **compareOffset = 2**	**Result is 00h**	
13	Initialise compareBuffer **compareBuffer =** **55 55 04** 01 **01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**		

	Compare buffers **compareOffset = 2**	**Result is -1**	
14	Initialise compareBuffer **compareBuffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D** 0D 10 **55**		
	Compare buffers **compareOffset = 2**	**Result is +1**	
15	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Initialise compareBuffer **CompareBuffer = 04 00 01 … 0F**		
	Successful call (with tag 8Dh) **tag = 8Dh** **compareBuffer.length = 17** **compareOffset = 0**	**Result is 00h**	
16	Append tag 0Fh **buffer = 00 01 … 0F**		
	Initialise compareBuffer **compareBuffer = 00 01 … 0F**		
	Successful call (with tag 8Fh) **tag = 8Fh** **compareBuffer.length = 16** **compareOffset = 0**	**Result is 00h**	

6.1.6.16.5 Test Coverage

CRR number	Test case number
N1	6
N2	8
N3	7, 11, 12
N4	9, 13
N5	10, 14
N6	15, 16
P1	1
P2	2, 3, 4, 5
C1	Does not apply for Proactive Handler

6.1.6.17 Method findAndCompareValue

6.1.6.17.1 Test Area Reference API_2_PAH_FACRBBS_BSS

6.1.6.17.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte findAndCompareValue(byte tag,
                                byte occurence,
                                short valueOffset,
                                byte[] compareBuffer,
                                short compareOffset,
                                short compareLength)
                         throws java.lang.NullPointerException,
                                java.lang.ArrayIndexOutOfBoundsException,
                                ToolkitException
```

Normal Execution

Looks for the indicated occurrence of a TLV element from the beginning of a TLV list and compare its value with a buffer:

CRRN1: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN2: if the method is successful then the corresponding TLV becomes current.

CRRN3: if identical 0 is returned.

CRRN4: if the first miscomparing byte in simple TLV is less than that in compareBuffer -1 is returned.

CRRN5: if the first miscomparing byte in simple TLV is greater than that in compareBuffer 1 is returned

CRRN6: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset or compareLength or both would cause access outside array bounds, or if compareLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, compareLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

CRRP4: if an input parameter is not valid (e.g. occurence = 0) an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException BAD_INPUT_PARAMETER.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.6.17.3 Test Suite files

- Test Script: API_2_PAH_FACRBBS_BSS_1.scr
- Test Applet: API_2_PAH_FACRBBS_BSS_1.java
- Installation parameter: API_2_PAH_FACRBBS_BSS.install

- Load Script: API_2_PAH_FACRBBS_BSS.ldr
- Conversion parameter: API_2_PAH_FACRBBS_BSS.cnv

6.1.6.17.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler		
	findAndCompareValue() with a null compareBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15		
	compareOffset ≥ compareBuffer.length **tag = 0Dh, occurrence = 1** **valueOffset = 0** **compareBuffer.length = 5** **compareOffset = 5** **compareLength = 1**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	compareOffset < 0 **compareBuffer.length = 5** **compareOffset = -1** **compareLength = 1**	`ArrayIndexOutOfBoundsException` is thrown	
4	compareLength >compareBuffer.length **compareBuffer.length = 5** **compareOffset = 0** **compareLength = 6**	`ArrayIndexOutOfBoundsException` is thrown	
5	compareOffset + compareLength >compareBuffer.length **compareBuffer.length = 5** **compareOffset = 3** **compareLength = 3**	`ArrayIndexOutOfBoundsException` is thrown	
6	compareLength < 0 **compareBuffer.length = 5** **compareOffset = 0** **compareLength = -1**	`ArrayIndexOutOfBoundsException` is thrown	
7	initDisplayText() with length = 5		
	valueOffset ≥ Text String Length **tag = 0Dh, occurrence = 1** **valueOffset = 6** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	
8	valueOffset < 0 **valueOffset = -1**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	

	compareBuffer.length = 15 **compareOffset = 0** **compareLength = 1**		
9	compareLength > Text String length **valueOffset = 0** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 7**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
10	valueOffset + compareLength > Text String length **valueOffset = 2** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 5**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
11	Invalid parameter **occurrence = 0**	**ToolkitException.BAD_INPUT_PARAMETER is thrown**	
12	InitDisplayText()		
	Select a TLV (tag 02h)		
	findAndCompareValue() **tag = 0Dh** **occurrence =** 2	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the `getValueLength()` method	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
13	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Initialise compareBuffer **compareBuffer =** **04 00 01 … 0F**		
	findAndCompareValue() **tag = 0Dh, occurrence = 1** **valueOffset = 0** **compareOffset = 0** **compareLength = 17**	**Result is 00h**	
14	Verify current TLV **getValueLength()**	**Result is 17**	
15	Initialise compareBuffer **compareBuffer =** **04 00 01 …** 10		
	Compare buffers with same parameters	**Result is -1**	

16	Initialise compareBuffer **compareBuffer =** 03 **00 01 … 0F**		
	Compare buffers with same parameters	**Result is +1**	
17	Initialise compareBuffer **compareBuffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**		
	Compare buffers **valueOffset = 2** **compareOffset = 3** **compareLength = 12**	**Result is 00h**	
18	Initialise compareBuffer **compareBuffer =** **55 55 55** 02 01 **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is -1**	
19	Initialise compareBuffer **compareBuffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A** 0A 0D **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is +1**	
20	append a Text String TLV **tag = 0Dh** **buffer = 00 11 22 33 44 55**		
	Initialise compareBuffer **compareBuffer =** **04 00 01 … 0F**		
	findAndCompareValue() **tag = 0Dh, occurrence = 1** **valueOffset = 0** **compareOffset = 0** **compareLength = 17**	**Result is 00h**	

21	Initialise compareBuffer **compareBuffer =** **00 11 22 33 44 55**		
	findAndCompareValue() **tag = 0Dh, occurrence = 2** **valueOffset = 0** **compareOffset = 0** **compareLength = 6**	**Result is 00h**	
22	Initialise compareBuffer **compareBuffer =** **00 11 22 33 44** 66		
	findAndCompareValue() **tag = 0Dh, occurrence = 2** **valueOffset = 0** **compareOffset = 0** **compareLength = 6**	**Result is -1**	
23	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Initialise compareBuffer **CompareBuffer = 04 00 01 … 0F**		
	Successful call (with tag 8Dh) **tag = 8Dh** **compareBuffer.length = 17** **compareOffset = 0**	**Result is 00h**	
24	Append tag 0Fh **buffer = 00 01 … 0F**		
	Initialise compareBuffer **compareBuffer = 00 01 … 0F**		
	Successful call (with tag 8Fh) **tag = 8Fh** **compareBuffer.length = 16** **compareOffset = 0**	**Result is 00h**	

6.1.6.17.5 Test Coverage

CRR number	Test case number
N1	12
N2	14
N3	13, 17, 20, 21
N4	15, 18, 22
N5	16, 19
N6	23, 24

P1	**1**
P2	**2, 3, 4, 5, 6**
P3	**7, 8, 9, 10**
P4	**11**
C1	**Does not apply for Proactive Handler**

6.1.6.18 Method appendArray

6.1.6.18.1 Test Area Reference: API_2_EDH_APDA

6.1.6.18.2 Conformance Requirement:

The method with the following method header shall comply to its definition in the API.

```
void appendArray(byte[] buffer, short offset, short length)
            throws java.lang.NullPointerException,
                   java.lang.ArrayIndexOutOfBoundsException,
                   ToolkitException
```

Normal Execution

CRRN1: appends a buffer into the Edithandler buffer

CRRN2: a successful append does not modify the TLV selected

Parameters error

CRRP1: if buffer is null, a java.lang.NullPointerException is thrown

CRRP2: if offset or length or both would cause access outside the array bounds, or if length is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.

Context error

CRRC1: if the EditHandler buffer is too small to append the requested data, a ToolkitException is thrown with reason code HANDLER_OVERFLOW

CRRC2: if the EditHandler buffer is busy, a ToolkitException is thrown with reason code HANDLER_NOT_AVAILABLE

6.1.6.18.3 Test suite files:

- Test Script : API_2_EDH_APDA_1.scr
- Test Applet: API_2_EDH_APDA_1.java
- Installation parameter: API_2_EDH_APDA.install (Same as default applet).
- Load Script: API_2_EDH_APDA.ldr
- Conversion parameter: API_2_EDH_APDA.cnv

6.1.6.18.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			

6.1.6.18.5 Test Coverage

CRR number	*Test case number*

6.1.6.19 Method appendTLV

6.1.6.19.1 Test Area Reference: API_2_EDH_APTLVBB

6.1.6.19.2 Conformance Requirement:

The method with the following method header shall comply to its definition in the API.

```
void appendTLV (byte tag, byte value)
               throws ToolkitException
```

Normal Execution

CRRN1 : Appends a TLV element to the current TLV list (1-byte element).

CRRN2 : A successful append does not modify the TLV selected.

Parameters error

None

Context error

CRRC1: if the EditHandler buffer is too small to append the requested data, a ToolkitException is thrown with reason code HANDLER_OVERFLOW

CRRC2: if the EditHandler buffer is busy, a ToolkitException is thrown with reason code HANDLER_NOT_AVAILABLE

6.1.6.19.3 Test suite files:

- Test Script: API_2_EDH_APTLVBB_1.scr
- Test Applet: API_2_EDH_APTLVBB_1.java
- Installation parameter: API_2_EDH_APTLVBB.install (Same as default applet).
- Load Script: API_2_EDH_APTLVBB.ldr
- Conversion parameter: API_2_EDH_APTLVBB.cnv

6.1.6.19.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			

6.1.6.19.5 Test Coverage

CRR number	*Test case number*

6.1.6.20 Method appendTLV

6.1.6.20.1 Test Area Reference: API_2_EDH_APTLVBBB

6.1.6.20.2 Conformance Requirements:

The method with the following method header shall comply to its definition in the API.

```
void appendTLV(byte tag, byte value1, byte value2)
               throws ToolkitException
```

Normal Execution

CRRN1: Appends a TLV element to the current TLV list (2-byte element).

CRRN2: A successful append does not modify the TLV selected.

Parameters error

None

Context error

CRRC1: if the EditHandler buffer is too small to append the requested data, a ToolkitException is thrown with reason code HANDLER_OVERFLOW

CRRC2: if the EditHandler buffer is busy, a ToolkitException is thrown with reason code HANDLER_NOT_AVAILABLE

6.1.6.20.3 Test suite files:

- Test Script: API_2_EDH_APTLVBBB_1.scr
- Test Applet: API_2_EDH_APTLVBBB_1.java
- Installation parameter: API_2_EDH_APTLVBBB.install
- Load Script: API_2_EDH_APTLVBBB.ldr
- Conversion parameter: API_2_EDH_APTLVBBB.cnv

6.1.6.20.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			

6.1.6.20.5 Test Coverage

CRR number	*Test case number*

6.1.6.21 Method appendTLV

6.1.6.21.1 Test Area Reference: API_2_EDH_APTLVB_BSS

6.1.6.21.2 Conformance Requirement:

The method with the following method header shall comply to its definition in the API.

```
void appendTLV(byte tag,
               byte[] value,
               short valueoffset,
               short valuelength)
```

```
                    throws java.lang.NullPointerException,
                           java.lang.ArrayIndexOutOfBoundsException,
                           ToolkitException
```

Normal Execution

CRRN1: Appends a TLV element to the current TLV list (byte-array element).

CRRN2: A successful append does not modify the TLV selected.

Parameters error

CRRP1: if value is null, a java.lang.NullPointerException is thrown

CRRP2: if valueoffset or valuelength or both would cause access outside the array bounds, or if length is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.

Context error

CRRC1: if the EditHandler buffer is too small to append the requested data, a ToolkitException is thrown with reason code HANDLER_OVERFLOW

CRRC2: if the EditHandler buffer is busy, a ToolkitException is thrown with reason code HANDLER_NOT_AVAILABLE

CRRC3: if valuelength is greater than 255, a ToolkitException is thrown with reason code BAD_INPUT_PARAMETER

6.1.6.21.3 Test suite files:

- Test Script: API_2_EDH_APTLVB_BSS_1.scr
- Test Applet: API_2_EDH_APTLVB_BSS_1.java
- Installation parameter: API_2_EDH_APTLVB_BSS.install (Same as default applet).
- Load Script: API_2_EDH_APTLVB_BSS.ldr
- Conversion parameter: API_2_EDH_APTLVB_BSS.cnv

6.1.6.21.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			

6.1.6.21.5 Test Coverage

CRR number	*Test case number*

6.1.6.22 Method appendTLV

6.1.6.22.1 Test Area Reference: API_2_EDH_APTLVBB_BSS

6.1.6.22.2 Conformance Requirement:

The method with the following method header shall comply to its definition in the API.

```
          void appendTLV (byte tag,
                          byte value1
                          byte[] value2,
                          short value2offset,
                          short value2length)
                         throws java.lang.NullPointerException,
```

```
                    java.lang.ArrayIndexOutOfBoundsException,
                    ToolkitException
```

Normal Execution

CRRN1 : Appends a TLV element to the current TLV list (1 byte and a byte-array element).

CRRN2 : A successful append does not modify the TLV selected.

Parameters error

CRRP1: if value2 is null, a java.lang.NullPointerException is thrown

CRRP2: if value2offset or value2length or both would cause access outside the array bounds, or if length is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.

Context error

CRRC1: if the EditHandler buffer is too small to append the requested data, a ToolkitException is thrown with reason code HANDLER_OVERFLOW

CRRC2: if the EditHandler buffer is busy, a ToolkitException is thrown with reason code HANDLER_NOT_AVAILABLE

CRRC3: if valuelength is greater than 255, a ToolkitException is thrown with reason code BAD_INPUT_PARAMETER

6.1.6.22.3 Test suite files:

- Test Script: API_2_EDH_APTLVBB_BSS_1.scr
- Test Applet: API_2_EDH_APTLVBB_BSS_1.java
- Installation parameter: API_2_EDH_APTLVBB_BSS.install (Same as default applet)
- Load Script: API_2_EDH_APTLVBB_BSS.ldr
- Conversion parameter: API_2_EDH_APTLVBB_BSS.cnv

6.1.6.22.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			

6.1.6.22.5 Test Coverage

CRR number	*Test case number*

6.1.6.23 Method clear

6.1.6.23.1 Test Area Reference: API_2_EDH_CLR

6.1.6.23.2 Conformance Requirement:

The method with the following method header shall comply to its definition in the API.

```
      void clear()
                throws ToolkitException
```

Normal Execution

CRRN1 : Clears the TLV list of an EditHandler and resets the current TLV selected.

Parameters error

No requirements

Context error

CRRC1 : if the EditHandler buffer is busy, a ToolkitException is thrown with reason code HANDLER_NOT_AVAILABLE

6.1.6.23.3 Test suite files:

- Test Script: API_2_EDH_CLR_1.scr
- Test Applet: API_2_EDH_CLR _1.java
- Installation parameter: API_2_EDH_CLR.install (Same as default applet)
- Load Script: API_2_EDH_CLR.ldr
- Conversion parameter: API_2_EDH_CLR.cnv

6.1.6.23.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			

6.1.6.23.5 Test Coverage

CRR number	*Test case number*

6.1.7 Class EnvelopeHandler

6.1.7.1 Method getEnvelopeTag

6.1.7.1.1 Test Area Reference: API_2_ENH_GENT

6.1.7.1.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public byte getEnvelopeTag()
```

Normal Execution

CRRN1: The method shall return the Envelope BER-TLV tag.

CRRN2: The Envelope BER TAG is available for all triggered toolkit applets from the invocation to the termination of their processToolkit method if the EnvelopeHandler is alvailable.

Parameters error

No requirements

Context error

No requirements

6.1.7.1.1 Test suite files:

- Test Script: API_2_ENH_GENT_1.scr
- Test Applet: API_2_ENH_GENT_1.java

- Installation parameter: API_2_ENH_GENT.install (Same as default applet)
- Load Script: API_2_ENH_GENT.ldr
- Conversion parameter: API_2_ENH_GENT.cnv

6.1.7.1.2 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	getEnvelopeTag called just after triggering of the application.	Returns 0xD1	
2	getEnvelopeTag called after a proactive command.	Returns 0xD1	
3	getEnvelopeTag called after a second proactive command.	Returns 0xD1	

6.1.7.1.3 Test Coverage

CRR number	*Test case number*
N1	***1, 2, 3***
N2	***1, 2, 3***

6.1.7.2 Method getItemIdentifier

6.1.7.2.1 Test Area Reference: API_2_ENH_GIID

6.1.7.2.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public byte getItemIdentifier()
        throws ToolkitException
```

Normal Execution

CRRN1: The method shall return the item identifier byte value.

CRRN2: The item identifier byte value returned shall be from the first Item Identifier TLV element.

CRRN3: If the element is available it becomes the TLV selected.

CRRN4: The item identifier is available for all triggered toolkit applets from the invocation to the termination of their processToolkit method if the EnvelopeHandler is available.

Parameters error

No requirements

Context error

CRRC1: The method shall throw ToolkitException (UNAVAILABLE_ELEMENT) if the item identifier TLV is not present.

6.1.7.2.3 Test suite files:

- Test Script: API_2_ENH_GIID_1.scr
- Test Applet: API_2_ENH_GIID_1.java
- Installation parameter: API_2_ENH_GIID.install (same as default applet)
- Load Script: API_2_ENH_GIID.ldr
- Conversion parameter: API_2_ENH_GIID.cnv

6.1.7.2.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Send envelope with item identifier TLV and identifier value of 03	Returns 03	
2	Send envelope with two item identifier TLV with first value FF and second 44	Returns FF	
3	Send envelope with two item identifier TLV with first value 81 and second 44, call twice the method getItemIdentifier	Returns 81 Returns 81	
4	Send envelope with item identifier TLV and value of 66. findTLV with TAG 02. getItemIdentifier and then getValueByte with offset 0	getItemIdentifier=getValueByte	
5	Send envelope without item identifier TLV and getItemIdentifier	ToolkitException (UNAVAILABLE_ELEMENT)	
6	Send Envelope with item identifier TLV (66), send proactive command. Then getItemIdentifier	Returns 66	

6.1.7.2.5 Test Coverage

CRR number	*Test case number*
N1	***1, 2, 3***
N2	***2, 3***
N3	***4***
N4	***6***
C1	***5***

6.1.7.3 Method getSecuredDataLength

6.1.7.3.1 Test Area Reference: API_2_ENH_GSDL

6.1.7.3.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public short getSecuredDataLength()
                    throws ToolkitException
```

Normal Execution

CRRN1: The method shall return the length of the secured data contained in a SMS TPDU TLV.

CRRN2: The length is from the first SMS TPDU TLV.

CRRN3: The length should not include padding bytes.

CRRN4: The method can be used if the event is EVENT_FORMATTED_SMS_PP_ENV and if the SMS TP-UD is formatted according to GSM03.48.

CRRN5: The method can be used if the event is EVENT_FORMATTED_SMS_PP_UPD and if the SMS TP-UD is formatted according to GSM03.48.

CRRN6: If the method is successful, the selected TLV should be the SMS TPDU TLV.

Parameters error

No requirements

Context error

CRRC1 : The method shall thrown ToolkitException (UNAVAILABLE_ELEMENT) in case of unavailable SMS TPDU TLV element.

CRRC2 : The method shall thrown ToolkitException (UNAVAILABLE_ELEMENT) in case of missing Secured Data.

6.1.7.3.3 Test suite files:

Specific triggering :

SMS CB

FORMATTED SMS PP UPD

- Test Script: API_2_ENH_GSDL_1.scr
- Test Applet: API_2_ENH_GSDL_1.java
- Installation parameter: API_2_ENH_GSDL.install (same as default applet)
- Load Script: API_2_ENH_GSDL.ldr
- Conversion parameter: API_2_ENH_GSDL.cnv

6.1.7.3.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Test with various length of TP-OA : 2, 6, 12		
2	Test with various length of RC/CC/DS : 0, 8		
3	Test with PCNTR = 0		
4	Test with PCNTR = 7		
5	Test with SecuredDataLength = 00	Returns 0x00	
6	Test with UserDataLength = 33		
7	Test with UserDataLength = 7F		
8	Test with UserDataLength = 80		
9	Test with UserDataLength = maximum length		
10	Send a SMS PP with 2 TPDU TLV and inside two different secured data lengths : 5 and 10	Returns 0x05	
11	Same test as 1 but with SMSPP_UPD		
12	Same test as 2 but with SMSPP_UPD		
13	Same test as 3 but with SMSPP_UPD		
14	Same test as 4 but with SMSPP_UPD		
15	Same test as 5 but with SMSPP_UPD		
16	Same test as 6 but with SMSPP_UPD		
17	Same test as 7 but with SMSPP_UPD		
18	Same test as 8 but with SMSPP_UPD		
19	Same test as 9 but with SMSPP_UPD		
20	Same test as 10 but with SMSPP_UPD		
21	findTLV device identities, getSecuredDataLength and then getValueByte to verify that the current TLV is the TPDU TLV		
22	Send an envelope SMS CB, getSecuredDataLength	ToolkitException UNAVAILABLE_ELEMENT	
23	Send an envelope SMS PP with UDHL = 0	ToolkitException UNAVAILABLE_ELEMENT	
24	Send an envelope SMS PP unformatted	ToolkitException UNAVAILABLE_ELEMENT	
25	Send an envelope 03.40 formatted, getSecuredDataLength	ToolkitException UNAVAILABLE_ELEMENT	

6.1.7.3.5 Test Coverage

CRR number	*Test case number*
N1	*1, 2, 3, 4, 5, 6, 7, 8, 9, 10*
N2	*10*
N3	*3, 4*
N4	*1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 25*
N5	*11, 12, 13, 14, 15,*

	16, 17, 18, 19, 20
N6	***21***
C1	***22***
C2	***23, 24, 25***

6.1.7.4 Method getSecureDataOffset

6.1.7.4.1 Test Area Reference: API_2_ENH_GSDO

6.1.7.4.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public short getSecuredDataOffset()
                    throws ToolkitException
```

Normal Execution

CRRN1: The method shall return the offset of the secured data first byte contained in a SMS TPDU TLV.

CRRN2: The offset is from the first SMS TPDU TLV.

CRRN3: The method can be used if the event is EVENT_FORMATTED_SMS_PP_ENV and if the SMS TP-UD is formatted according to GSM03.48.

CRRN4: The method can be used if the event is EVENT_FORMATTED_SMS_PP_UPD and if the SMS TP-UD is formatted according to GSM03.48.

CRRN5: If the method is successful, the selected TLV should be the SMS TPDU TLV.

Parameters error

No requirements

Context error

CRRC1: The method shall thrown ToolkitException (UNAVAILABLE_ELEMENT) in case of unavailable SMS TPDU TLV element.

CRRC2: The method shall thrown ToolkitException (UNAVAILABLE_ELEMENT) in case of missing Secured Data.

6.1.7.4.3 Test suite files:

Specific triggering :

SMS CB

FORMATTED SMS PP UPD

- Test Script: API_2_ENH_GSDO_1.scr
- Test Applet: API_2_ENH_GSDO_1.java
- Installation parameter: API_2_ENH_GSDO.install (same as default applet)
- Load Script: API_2_ENH_GSDO.ldr
- Conversion parameter: API_2_ENH_GSDO.cnv

6.1.7.4.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Test with TP-OA length of 2**		
2	**Test with TP-OA length of 6**		
3	**Test with TP-OA length of 12**		

4	**Test with RC/CC/DS length of 0**		
5	**Test with RC/CC/DS length of 8**	**Returns 0x00**	
6	**Send a SMS PP with 2 TPDU TLV and inside two different secured data offsets**	**Returns X**	
7	**Same test as 1 but with SMSPP_UPD**		
8	**Same test as 2 but with SMSPP_UPD**		
9	**Same test as 3 but with SMSPP_UPD**		
10	**Same test as 4 but with SMSPP_UPD**		
11	**Same test as 5 but with SMSPP_UPD**		
12	**Same test as 6 but with SMSPP_UPD**		
13	**findTLV device identities, getSecuredDataOffset and then getValueByte to verify that the current TLV is the TPDU TLV**		
14	**Send an envelope SMS CB, getSecuredDataOffset**	**ToolkitException UNAVAILABLE_ELEMENT**	
15	**Send an envelope SMS PP with UDHL = 0**	**ToolkitException UNAVAILABLE_ELEMENT**	
16	**Send an envelope SMS PP unformatted**	**ToolkitException UNAVAILABLE_ELEMENT**	
17	**Send an envelope 03.40 formatted, getSecuredDataOffset**	**ToolkitException UNAVAILABLE_ELEMENT**	

6.1.7.4.5 Test Coverage

CRR number	*Test case number*
N1	***1, 2, 3, 4, 5, 6***
N2	***6***
N3	***1, 2, 3, 4, 5, 6, 17***
N4	***7, 8, 9, 10, 11, 12***
N5	***13***
C1	***14***
C2	***15, 16, 17***

6.1.7.5 Method getTheHandler

6.1.7.5.1 Test Area Reference: API_2_ENH_GTHD

6.1.7.5.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public static getTheHandler()
    throws ToolkitException
```

Normal Execution

CRRN1: The method shall return the single system instance of the EnvelopeHandler class.

CRRN2: The EnvelopeHandler is a Temporary JCRE Entry Point Object (see ref X)

Parameters error

No requirements

Context error

CRRC1: The method shall thrown ToolkitException (HANDLER_NOT_AVAILABLE) if the handler is busy.

6.1.7.5.3 Test suite files:

- Test Script: API_2_ENH_GTHD_1.scr
- Test Applet: API_2_ENH_GTHD_1.java

- Installation parameter: API_2_ENH_GTHD.install (same as default applet)
- Load Script: API_2_ENH_GTHD.ldr
- Conversion parameter: API_2_ENH_GTHD.cnv

6.1.7.5.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	getTheHandler twice	The returned objects shall be the same	
2	getTheHandler	The reference returned shall be an EnvelopeHandler (checkcast)	
3	getTheHandler	The reference returned shall not be null.	
4	getTheHandler and store it in a static field of the toolkit applet	SecurityException is thrown	
5	getTheHandler and store it in a field of the toolkit applet	SecurityException is thrown	

6.1.7.5.5 Test Coverage

CRR number	*Test case number*
N1	***1, 2, 3***
N2	***4,5***
C1	***To be checked in Framework tests and insert here cross reference***

6.1.7.6 Method getTPUDLOffset

6.1.7.6.1 Test Area Reference: API_2_ENH_GTPO

6.1.7.6.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public short getTPUDLOffset()
             throws ToolkitException
```

Normal Execution

CRRN1: The method shall return the TPUDL offset in a SMS TPDU TLV.

CRRN2: The offset is from the first SMS TPDU TLV.

CRRN3: The method can be used if the event is EVENT_FORMATTED_SMS_PP_ENV.

CRRN4: The method can be used if the event is EVENT_FORMATTED_SMS_PP_UPD.

CRRN5: The method can be used if the event is EVENT_UNFORMATTED_SMS_PP_ENV.

CRRN6: The method can be used if the event is EVENT_UNFORMATTED_SMS_PP_UPD.

CRRN7: If the method is successful, the selected TLV should be the SMS TPDU TLV.

Parameters error

No requirements

Context error

CRRC1: The method shall thrown ToolkitException (UNAVAILABLE_ELEMENT) in case of unavailable SMS TPDU TLV element.

CRRC2: The method shall thrown ToolkitException (UNAVAILABLE_ELEMENT) if the TPUDL field does not exist.

6.1.7.6.3 Test suite files:

Specific triggering :

FORMATTED SMS PP UPD

UNFORMATTED SMS PP UPD

UNFORMATTED SMS PP ENV

SMS CB

- Test Script: API_2_ENH_GTPO_1.scr
- Test Applet: API_2_ENH_GTPO_1.java
- Installation parameter: API_2_ENH_GTPO.install (Same as default applet)
- Load Script: API_2_ENH_GTPO.ldr

  At the end of the script the applet is loaded but not instantiated.

- Conversion parameter: API_2_ENH_GTPO.cnv

6.1.7.6.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Test with TP-OA length of 2		
2	Test with TP-OA length of 6		
3	Test with TP-OA length of 12		
4	Send a SMS PP with 2 TPDU TLV and inside two different UDL offsets		
5	Same test as 1 but with SMSPP formatted_UPD		
6	Same test as 2 but with SMSPP formatted UPD		
7	Same test as 3 but with SMSPP formatted UPD		
8	Same test as 4 but with SMSPP formatted UPD		
9	Same test as 1 but with SMSPP unformatted UPD		
10	Same test as 2 but with SMSPP unformatted UPD		
11	Same test as 3 but with SMSPP unformatted UPD		
12	Same test as 4 but with SMSPP unformatted UPD		
13	Same test as 1 but with SMSPP unformatted ENV		
14	Same test as 2 but with SMSPP unformatted ENV		
15	Same test as 3 but with SMSPP unformatted ENV		
16	Same test as 4 but with SMSPP unformatted ENV		
17	findTLV device identities, getTPUDLOffset and then getValueByte to verify that the current TLV is the TPDU TLV		
18	Send an envelope SMS CB, getTPUDLOffset	ToolkitException UNAVAILABLE_ELEMENT	

19	**Send an envelope SMS PP unformatted without TPUDL field, getTPUDLOffset**	**ToolkitException UNAVAILABLE_ELEMENT**	

6.1.7.6.5 Test Coverage

CRR number	***Test case number***
N1	***1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17***
N2	***4***
N3	***1, 2, 3, 4, 17***
N4	***5, 6, 7, 8***
N5	***13, 14, 15, 16***
N6	***9, 10, 11, 12***
N7	***17***
C1	***18***
C2	***19***

6.1.7.7 Method getLength

6.1.7.7.1 Test Area Reference API_2_PAH_GLEN

6.1.7.7.2 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
public short getLength()
                    throws ToolkitException
```

Normal Execution

CRRN1: returns the length in bytes of the TLV list.

Parameter Error

No requirements

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.HANDLER_NOT_AVAILABLE.

6.1.7.7.3 Test Suite files

- Test Script: API_2_PAH_GLEN_1.scr
- Test Applet: API_2_PAH_GLEN_1.java
- Installation parameter: API_2_PAH_GLEN.install
- Load Script: API_2_PAH_GLEN.ldr
- Conversion parameter: API_2_PAH_GLEN.cnv

6.1.7.7.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Clear the handler**	Result of `getLength()` is 0	

	getLength()		
2	**Call the `init()` method** `getLength()`	Result of `getLength()` is 9	
3	**Call the initDisplayText() method, with buffer length = 240** **getLength()**	Result of `getLength()` is 253	
4	**Build a 7Fh Proactive Handler** **getLength()**	Result of `getLength()` is 7Fh	
5	**Build a 80h Proactive Handler** **getLength()**	Result of `getLength()` is 80h	

6.1.7.7.5 Test Coverage

CRR number	Test case number
N1	**1, 2, 3**
C1	**Does not apply for Proactive Handler**

6.1.7.8 Method copy

6.1.7.8.1 Test Area Reference API_2_PAH_COPY_BSS

6.1.7.8.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short copy(byte[] dstBuffer,
                  short dstOffset,
                  short dstLength)
                   throws java.lang.NullPointerException,
                        java.lang.ArrayIndexOutOfBoundsException,
                        ToolkitException
```

Normal Execution

CRRN1: copies the simple TLV list contained in the handler to the destination byte array.

CRRN2: returns dstOffset + dstLength.

Parameter Error

CRRP1: if dstBuffer is null a NullPointerException is thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative, an ArrayIndexOutOfBoundsException is thrown.

CRRP3: if dstLength is grater than the length of the simple TLV List, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException. OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.7.8.3 Test Suite files

- Test Script: API_2_PAH_COPY_BSS_1.scr
- Test Applet: API_2_PAH_COPY_BSS_1.java
- Installation parameter: API_2_PAH_COPY_BSS.install
- Load Script: API_2_PAH_COPY_BSS.ldr
- Conversion parameter: API_2_PAH_COPY_BSS.cnv

6.1.7.8.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**NULL as parameter to dstBuffer**	`NullPointerException` is thrown	
2	Call the `init()` method		
	dstOffset ≥ dstBuffer.length **dstBuffer.length = 5** dstOffset = 5 dstLength = 1	`ArrayIndexOutOfBoundsException` is thrown	
3	**dstOffset < 0** **dstBuffer.length = 5** dstOffset = -1 dstLength = 1	`ArrayIndexOutOfBoundsException` is thrown	
4	**dstLength > dstBuffer.length** **dstBuffer.length = 5** dstOffset = 0 dstLength = 6	`ArrayIndexOutOfBoundsException` is thrown	
5	**dstOffset + dstLength > dstBuffer.length** **dstBuffer.length = 5** dstOffset = 3 dstLength = 3	`ArrayIndexOutOfBoundsException` is thrown	
6	**dstLength < 0** **dstBuffer.length = 5** dstOffset = 0 dstLength = -1	`ArrayIndexOutOfBoundsException` is thrown	
7	**dstLength > length of the simple TLV list** **dstBuffer.length = 10** dstOffset = 0	`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown	

	dstLength = 10		
8	**Successful call, dstBuffer is the whole buffer** **dstBuffer.length = 9** dstOffset = 0 dstLength = 9	Result of copy() is 9	
9	**Compare the buffer**	Result of arrayCompare() is 0	
10	**Successful call, dstBuffer is part of a buffer** **dstBuffer.length = 15** dstOffset = 3 dstLength = 9	Result of copy() is 12	
11	**Compare the whole buffer**	Result of arrayCompare() is 0	
12	**Successful call, dstBuffer is part of a buffer** **dstBuffer.length = 15** dstOffset = 3 dstLength = 6	Result of copy() is 9	
13	**Compare the whole buffer**	Result of arrayCompare() is 0	

6.1.7.8.5 Test Coverage

CRR number	Test case number
N1	9, 11, 13
N2	8, 10, 12
P1	1
P2	2, 3, 4, 5, 6
P3	7
C1	Does not apply for Proactive Handler

6.1.7.9 Method findTLV

6.1.7.9.1 Test Area Reference API_2_PAH_FINDBB

6.1.7.9.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte findTLV(byte tag, byte occurrence)
              throws ToolkitException
```

Normal Execution

Looks for the indicated occurrence of a TLV element from the beginning of the TLV list (handler buffer):

CRRN1: the method is successful if the required occurrence exists then the corresponding TLV becomes current.

CRRN2: if the method is successful then it returns TLV_FOUND_CR_SET when Comprehension Required flag is set.

CRRN3: if the method is successful then it returns TLV_FOUND_CR_NOT_SET when Comprehension Required flag is not set.

CRRN4: if the required occurrence of the TLV element does not exist, no TLV is selected and TLV_NOT_FOUND is returned.

CRRN5: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if an input parameter is not valid (e.g. occurrence = 0) an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException BAD_INPUT_PARAMETER. The current TLV is no longer defined.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.7.9.3 Test Suite files

- Test Script: API_2_PAH_FINDBB_1.scr
- Test Applet: API_2_PAH_FINDBB_1.java
- Installation parameter: API_2_PAH_FINDBB.install
- Load Script: API_2_PAH_FINDBB.ldr
- Conversion parameter: API_2_PAH_FINDBB.cnv

6.1.7.9.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialise the handler**		
	Invalid input parameter **`Occurrence = 0`**	ToolkitException.BAD_INPUT_PARAMETER is thrown	
2	**Call the `init()` method**		
	Search 1st TLV **`Tag = 01h`** `Occurrence = 1`	**Result is `TLV_FOUND_CR_SET`**	
3	**Call the `getValueLength()` method**	**Result is 03h**	
4	**Search 2nd TLV** **`Tag = 02h`** `Occurrence = 1`	**Result is `TLV_FOUND_CR_SET`**	
5	**Call the `getValueLength()` method**	**Result is 02h**	
6	**Search a wrong tag** **`Tag = 03h`** `Occurrence = 1`	**Result is `TLV_NOT_FOUND`**	
7	**Call the `getValueLength()` method**	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	

8	**Search a tag with wrong occurrence** **Tag = 01h** Occurrence = 2	**Result is TLV_NOT_FOUND**	
9	**Call the `getValueLength()` method**	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
10	**Append a TLV with tag=02h**		
	Search the TLV **Tag = 02h** Occurrence = 2	**Result is TLV_FOUND_CR_NOT_SET**	
11	**Append a TLV with tag=04h**		
	Search the TLV **Tag = 04h** Occurrence = 1	**Result is TLV_FOUND_CR_NOT_SET**	
12	**Search tag 81h** **Tag = 81h** Occurrence = 1	**Result is TLV_FOUND_CR_SET**	
13	**Search tag 84h** **Tag = 84h** Occurrence = 1	**Result is TLV_FOUND_CR_NOT_SET**	

6.1.7.9.5 Test Coverage

CRR number	Test case number
N1	3, 5
N2	2, 4
N3	10, 11
N4	6, 7,8, 9
N5	12, 13
P1	1
C1	Does not apply for Proactive Handler

6.1.7.10 Method getValueLength

6.1.7.10.1 Test Area Reference API_2_PAH_GVLEN

6.1.7.10.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short getValueLength()
                        throws ToolkitException
```

Normal Execution

CRRN1: gets and returns the binary length of the value field for the last TLV element which has been found in the handler.

Parameter Error

No requirements

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.7.10.3 Test Suite files

- Test Script: API_2_PAH_GVLEN_1.scr
- Test Applet: API_2_PAH_GVLEN_1.java
- Installation parameter: API_2_PAH_GVLEN.install
- Load Script: API_2_PAH_GVLEN.ldr
- Conversion parameter: API_2_PAH_GVLEN.cnv

6.1.7.10.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the `init()` method**		
	getValueLength()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	**Call the `initDisplayText()` method** **`length = 0`**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `00h`**	
3	**Call the `initDisplayText()` method** **`length = 1 (+ dcs byte)`**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `02h`**	
4	**Call the `initDisplayText()` method** **`length = 7Eh (+ dcs byte)`**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `7Fh`**	
5	**Call the `initDisplayText()` method** **`length = 7Fh (+ dcs byte)`**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `80h`**	
6	**Call the `initDisplayText()` method** **`length = F0h` (maximum text length)**		
	Search TLV `0Dh` (Text String TLV)		

	getValueLength()	**Result is F1h**	

6.1.7.10.5 Test Coverage

CRR number	Test case number
N1	**2, 3, 4, 5, 6**
C1	**Does not apply for Proactive Handler**
C2	**1**

6.1.7.11 Method getValueByte

6.1.7.11.1 Test Area Reference API_2_PAH_GVBYTS

6.1.7.11.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte getValueByte(short valueOffset)
                    throws ToolkitException
```

Normal Execution

CRRN1: Gets a byte from the last TLV element which has been found in the handler and returns its value (1 byte).

Parameter Error

CRRP1: if valueOffset is out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.7.11.3 Test Suite files

- Test Script: API_2_PAH_GVBYTS_1.scr
- Test Applet: API_2_PAH_GVBYTS_1.java
- Installation parameter: API_2_PAH_GVBYTS.install
- Load Script: API_2_PAH_GVBYTS.ldr
- Conversion parameter: API_2_PAH_GVBYTS.cnv

6.1.7.11.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	**Call the `init()` method**
	
	`type = FFh`
	`qualifier = FEh`
	
	`destination = FDh`		

	getValueByte(0)	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	Search TLV 01h (Command Details TLV)		
	getValueByte(3)	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
3	Search TLV 01h (Command Details TLV)		
	getValueByte(2)	**Result is FEh (qualifier)**	
4	Search TLV 02h (Device Identities TLV)		
	getValueByte(0)	**Result is 81h (Source)**	
5	initDisplayText() **buffer = 00 01 … 7D** **length = 7Eh** Search TLV 0Dh (Text String TLV)		
	getValueByte(7E)	**Result is 7Dh**	
6	initDisplayText() **buffer = 00 01 … 7D 7E** **length = 7Fh** Search TLV 0Dh (Text String TLV)		
	getValueByte(7E)	**Result is 7Dh**	
7	getValueByte(7F)	**Result is 7Eh**	
8	initDisplayText() **buffer = 00 01 … EF** **length = F0h** Search TLV 0Dh (Text String TLV)		
	getValueByte(F0)	**Result is EFh**	

6.1.7.11.5 Test Coverage

CRR number	Test case number
N1	3, 4, 5, 6, 7, 8
P1	2
C1	Does not apply for Proactive Handler
C2	1

6.1.7.12 Method copyValue

6.1.7.12.1 Test Area Reference API_2_PAH_CPYVS_BSS

6.1.7.12.2 Conformance Requirement

The method with following prototype shall be compliant with its definition in the API.

```
public short copyValue(short valueOffset,
                       byte[] dstBuffer,
```

```
                    short dstOffset,
                    short dstLength)
            throws java.lang.NullPointerException,
                    java.lang.ArrayIndexOutOfBoundsException,
                    ToolkitException
```

Normal Execution

CRRN1: copies a part of the last TLV element which has been found, into a destination. buffer.

CRRN2: returns dstOffset + dstLength.

Parameter Error

CRRP1: if dstBuffer is null NullPointerException is thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative ArrayIndexOutOfBoundsException is thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.7.12.3 Test Suite files

- Test Script: API_2_PAH_CPYVS_BSS _1.scr
- Test Applet: API_2_PAH_GVBYTS_1.java
- Installation parameter: API_2_PAH_GVBYTS.install
- Load Script: API_2_PAH_GVBYTS.ldr
- Conversion parameter: API_2_PAH_GVBYTS.cnv

6.1.7.12.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler Select a TLV		
	copyValue() with a null dstBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15 Select Text String TLV		
	dstOffset ≥ dstBuffer.length **`dstBuffer.length = 5`** **`dstOffset = 5`** **`dstLength = 1`**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	dstOffset < 0	`ArrayIndexOutOfBoundsException` is thrown	

	dstBuffer.length = 5 **dstOffset = -1** **dstLength = 1**	xception is thrown	
4	dstLength >dstBuffer.length **dstBuffer.length = 5** **dstOffset = 0** **dstLength = 6**	ArrayIndexOutOfBoundsException is thrown	
5	dstOffset + dstLength >dstBuffer.length **dstBuffer.length = 5** **dstOffset = 3** **dstLength = 3**	ArrayIndexOutOfBoundsException is thrown	
6	dstLength < 0 **dstBuffer.length = 5** **dstOffset = 0** **dstLength = -1**	ArrayIndexOutOfBoundsException is thrown	
7	initDisplayText() with length = 5 Select Text String TLV		
	valueOffset ≥ Text String Length **valueOffset = 6** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
8	valueOffset < 0 **valueOffset = -1** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
9	dstLength > Text String length **valueOffset = 0** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 7**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
10	valueOffset + dstLength > Text String length **valueOffset = 2** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 5**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
11	Initialise the handler		
	copyValue()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
12	initDisplayText() **dcs = 4**		

	buffer = 00 01 … 0F Select Text String TLV		
	Successful call **valueOffset = 0** **dstBuffer.length = 17** **dstOffset = 0** **dstLength = 17**	**Result of copyValue() is 17**	
13	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
14	initialise dstBuffer **dstBuffer = 55 55 … 55**		
	Successful call **valueOffset = 2** **dstBuffer.length = 20** **dstOffset = 3** **dstLength = 12**	**Result of copyValue() is 15**	
15	Compare buffer **buffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**	Result is 00h	

6.1.7.12.5 Test Coverage

CRR number	Test case number
N1	13, 15
N2	12, 14
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for Proactive Handler
C2	11

6.1.7.13 Method compareValue

6.1.7.13.1 Test Area Reference API_2_PAH_CPRVS_BSS

6.1.7.13.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte compareValue(short valueOffset,
                         byte[] compareBuffer,
                         short compareOffset,
                         short compareLength)
```

```
                    throws java.lang.NullPointerException,
                           java.lang.ArrayIndexOutOfBoundsException,
                           ToolkitException
```

Normal Execution

Compares the last found TLV element with a buffer:

CRRN1: returns 0 if identical.

CRRN2: returns -1 if the first miscomparing byte in simple TLV List is less than that in compareBuffer.

CRRN3: returns 1 if the first miscomparing byte in simple TLV List is greater than that in compareBuffer.

Parameter Error

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset or compareLength or both would cause access outside array bounds, or if compareLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.7.13.3 Test Suite files

- Test Script: API_2_PAH_CPRVS_BSS_1.scr
- Test Applet: API_2_PAH_CPRVS_BSS_1.java
- Installation parameter: API_2_PAH_CPRVS_BSS.install
- Load Script: API_2_PAH_CPRVS_BSS.ldr
- Conversion parameter: API_2_PAH_CPRVS_BSS.cnv

6.1.7.13.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler Select a TLV		
	compareValue() with a null compareBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15 Select Text String TLV		
	compareOffset ≥ compareBuffer.length **`compareBuffer.length = 5`** **`compareOffset = 5`** **`compareLength = 1`**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	compareOffset < 0	`ArrayIndexOutOfBoundsException` is thrown	

	compareBuffer.length = 5 **compareOffset = -1** **compareLength = 1**		
4	compareLength >compareBuffer.length **compareBuffer.length = 5** **compareOffset = 0** **compareLength = 6**	`ArrayIndexOutOfBoundsException` is thrown	
5	compareOffset + compareLength >compareBuffer.length **compareBuffer.length = 5** **compareOffset = 3** **compareLength = 3**	`ArrayIndexOutOfBoundsException` is thrown	
6	compareLength < 0 **compareBuffer.length = 5** **compareOffset = 0** **compareLength = -1**	`ArrayIndexOutOfBoundsException` is thrown	
7	initDisplayText() with length = 5 Select Text String TLV		
	valueOffset ≥ Text String Length **valueOffset = 6** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	
8	valueOffset < 0 **valueOffset = -1** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	
9	compareLength > Text String length **valueOffset = 0** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 7**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	
10	valueOffset + compareLength > Text String length **valueOffset = 2** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 5**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	
11	Initialise the handler		
	compareValue()	`ToolkitException.UNAVAILABLE_ELEMENT` is thrown	
12	initDisplayText()		

	dcs = 4 **buffer = 00 01 … 0F** Select Text String TLV		
	Initialise compareBuffer **compareBuffer =** **04 00 01 … 0F**		
	Compare buffers **valueOffset = 0** **compareOffset = 0** **compareLength = 17**	**Result is 00h**	
13	Initialise compareBuffer **compareBuffer =** **04 00 01 02 03** **04 05 06 07 08** **05 0A 0B 0C 0D** **0E** 10		
	Compare buffers with same parameters	**Result is -1**	
14	Initialise compareBuffer **compareBuffer =** 03 **00 01 … 0F**		
	Compare buffers with same parameters	**Result is +1**	
15	Initialise compareBuffer **compareBuffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**		
	Compare buffers **valueOffset = 2** **compareOffset = 3** **compareLength = 12**	**Result is 00h**	
16	Initialise compareBuffer **compareBuffer =** **55 55 55** 02 01 **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is -1**	
17	Initialise compareBuffer		

	compareBuffer = **55 55 55 01 02** **03 04 05 06 07** **08 09 0A** 0A 0D **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is +1**	

6.1.7.13.5 Test Coverage

CRR number	Test case number
N1	**12, 15**
N2	**13, 16**
N3	**14, 17**
P1	**1**
P2	**2, 3, 4, 5, 6**
P3	**7, 8, 9, 10**
C1	**Does not apply for Proactive Handler**
C2	**11**

6.1.7.14 Method findAndCopyValue

6.1.7.14.1 Test Area Reference API_2_PAH_FACYB_BS

6.1.7.14.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short findAndCopyValue(byte tag,
                              byte[] dstBuffer,
                              short dstOffset)
                       throws java.lang.NullPointerException,
                              java.lang.ArrayIndexOutOfBoundsException,
                              ToolkitException
```

Normal Execution

CRRN1: looks for the first occurrence of a TLV element from the beginning of a TLV list and copy its value into a destination buffer.

CRRN2: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN3: if the method is successful then the corresponding TLV becomes current and dstOffset + length of the copied value is returned.

CRRN4: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if dstBuffer is null NullPointerException shall be thrown.

CRRP2: if dstOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.7.14.3 Test Suite files

- Test Script: API_2_PAH_FACYB_BS_1.scr
- Test Applet: API_2_PAH_FACYB_BS_1.java
- Installation parameter: API_2_PAH_FACYB_BS.install
- Load Script: API_2_PAH_FACYB_BS.ldr
- Conversion parameter: API_2_PAH_FACYB_BS.cnv

6.1.7.14.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler		
	FindAndCopyValue() with a null dstBuffer	`NullPointerException` is thrown	
2	InitDisplayText() with length = 15		
	dstOffset ≥ dstBuffer.length **`tag = 0Dh`** **`dstBuffer.length = 20`** **`dstOffset = 20`**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	dstOffset < 0 **`dstBuffer.length = 20`** **`dstOffset = -1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	length > dstBuffer.length **`dstBuffer.length = 15`** **`dstOffset = 0`**	`ArrayIndexOutOfBoundsException` is thrown	
5	DstOffset + length >dstBuffer.length **`DstBuffer.length = 20`** **`DstOffset = 5`**	`ArrayIndexOutOfBoundsException` is thrown	
6	initDisplayText()		
	Select a TLV (tag 02h)		
	findAndCopyValue() **`tag = 03h`**	`ToolkitException.UNAVAILABLE_ELEMENT` is thrown	
	Call the `getValueLength()` method	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
7	initDisplayText() **`dcs = 4`** **`buffer = 00 01 … 0F`**		
	Successful call **`Tag = 0Dh`**	**Result of `copyValue()` is 17**	

	DstBuffer.length = 17 **DstOffset = 0**		
8	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
9	initialise dstBuffer **dstBuffer = 55 55 … 55**		
	Successful call **dstBuffer.length = 20** **dstOffset = 2**	**Result of copyValue() is 19**	
10	Compare buffer **buffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**	Result is 00h	
11	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	append a 2nd Text String TLV		
	Successful call **tag = 0Dh** **dstBuffer.length = 17** **dstOffset = 0**	**Result of copyValue() is 17**	
12	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
13	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Successful call (with tag 8Dh) **tag = 8Dh** **dstBuffer.length = 17** **dstOffset = 0**	**Result of copyValue() is 17**	
14	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
15	Append tag 0Fh **buffer = 00 01 … 0F**		
	Successful call (with tag 8Fh) **tag = 8Fh** **dstBuffer.length = 16** **dstOffset = 0**	**Result of copyValue() is 16**	

16	Compare buffer **buffer = 00 01 … 0F**	Result is 00h	

6.1.7.14.5 Test Coverage

CRR number	Test case number
N1	8, 10, 12
N2	6
N3	7, 9, 11
N4	13, 14, 15, 16
P1	1
P2	2, 3, 4, 5
C1	Does not apply for Proactive Handler

6.1.7.15 Method findAndCopyValue

6.1.7.15.1 Test Area Reference API_2_PAH_FACYBS_BSS

6.1.7.15.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short findAndCopyValue(byte tag,
                              byte occurence,
                             short valueOffset,
                             byte[] dstBuffer,
                             short dstOffset,
                             short dstLength)
                      throws java.lang.NullPointerException,
                             java.lang.ArrayIndexOutOfBoundsException,
                             ToolkitException
```

Normal Execution

CRRN1: looks for the indicated occurrence of a TLV element from the beginning of a TLV list and copy its value into a destination buffer.

CRRN2: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN3: if the method is successful then the corresponding TLV becomes current and dstOffset + dstLength is returned.

CRRN4: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if dstBuffer is null NullPointerException shall be thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.7.15.3 Test Suite files

- Test Script: API_2_PAH_FACYBS_BSS_1.scr
- Test Applet: API_2_PAH_FACYBS_BSS_1.java
- Installation parameter: API_2_PAH_FACYBS_BSS.install
- Load Script: API_2_PAH_FACYBS_BSS.ldr
- Conversion parameter: API_2_PAH_FACYBS_BSS.cnv

6.1.7.15.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler		
	findAndCopyValue() with a null dstBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15		
	dstOffset ≥ dstBuffer.length **`tag = 0Dh, occurrence = 1`** **`valueOffset = 0`** **`dstBuffer.length = 5`** **`dstOffset = 5`** **`dstLength = 1`**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	dstOffset < 0 **`dstBuffer.length = 5`** **`dstOffset = -1`** **`dstLength = 1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	dstLength >dstBuffer.length **`dstBuffer.length = 5`** **`dstOffset = 0`** **`dstLength = 6`**	`ArrayIndexOutOfBoundsException` is thrown	
5	dstOffset + dstLength >dstBuffer.length **`dstBuffer.length = 5`** **`dstOffset = 3`** **`dstLength = 3`**	`ArrayIndexOutOfBoundsException` is thrown	
6	dstLength < 0 **`dstBuffer.length = 5`** **`dstOffset = 0`** **`dstLength = -1`**	`ArrayIndexOutOfBoundsException` is thrown	
7	initDisplayText() with length = 5		
	valueOffset ≥ Text String Length	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	

	tag = 0Dh, occurrence = 1 **valueOffset = 6** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**		
8	valueOffset < 0 **valueOffset = -1** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
9	dstLength > Text String length **valueOffset = 0** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 7**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
10	valueOffset + dstLength > Text String length **valueOffset = 2** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 5**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
11	InitDisplayText()		
	Select a TLV (tag 02h)		
	findAndCopyValue() **tag = 0Dh** **occurrence =** 2	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the `getValueLength()` method	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
12	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Successful call **tag = 0Dh, occurrence = 1** **valueOffset = 0** **dstBuffer.length = 17** **dstOffset = 0** **dstLength = 17**	**Result of `findAndCopyValue()` is 17**	
13	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
14	initialise dstBuffer **dstBuffer = 55 55 … 55**		
	Successful call **tag = 0Dh, occurrence = 1**	**Result of `copyValue()` is 15**	

	valueOffset = 2 **dstBuffer.length = 20** **dstOffset = 3** **dstLength = 12**		
15	Compare buffer **buffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**	Result is 00h	
16	Append a Text String TLV **tag = 0D** **buffer = 00 11 22 33 44 55 (no specific DCS byte)**		
	Successful call **tag = 0Dh, occurrence = 1** **valueOffset = 0** **dstBuffer.length = 17** **dstOffset = 0** **dstLength = 17**	**Result of findAndCopyValue() is 17**	
17	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
18	Successful call **tag = 0Dh, occurrence = 2** **valueOffset = 0** **dstBuffer.length = 6** **dstOffset = 0** **dstLength = 6**	**Result of findAndCopyValue() is 6**	
19	Compare buffer **buffer = 00 11 22 33 44 55**	Result is 00h	
20	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Successful call (with tag 8Dh) **tag = 8Dh** **dstBuffer.length = 17** **dstOffset = 0**	**Result of copyValue() is 17**	
21	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
22	Append tag 0Fh **buffer = 00 01 … 0F**		

	Successful call (with tag 8Fh) **tag = 8Fh** **dstBuffer.length = 16** **dstOffset = 0**	**Result of copyValue() is 16**	
23	Compare buffer **buffer = 00 01 … 0F**	Result is 00h	

6.1.7.15.5 Test Coverage

CRR number	Test case number
N1	**13, 15, 17, 19**
N2	**11**
N3	**12, 14, 16, 18**
N4	**20, 21, 22, 23**
P1	**1**
P2	**2, 3, 4, 5, 6**
P3	**7, 8, 9, 10**
C1	**Does not apply for Proactive Handler**

6.1.7.16 Method findAndCompareValue

6.1.7.16.1 Test Area Reference API_2_PAH_FACRB_BS

6.1.7.16.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte findAndCompareValue(byte tag,
                                byte[] compareBuffer,
                                short compareOffset)
                throws     java.lang.NullPointerException,
                           java.lang.ArrayIndexOutOfBoundsException,
                      ToolkitException
```

Normal Execution

Looks for the first occurrence of a TLV element from beginning of a TLV list and compare its value with a buffer :

CRRN1: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN2: if the method is successful then the corresponding TLV becomes current.

CRRN3: if identical returns 0.

CRRN4: if the first miscomparing byte in simple TLV is less than that in compareBuffer returns -1.

CRRN5: if the first miscomparing byte in simple TLV is greater than that in compareBuffer returns 1.

CRRN6: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.7.16.3 Test Suite files

- Test Script: API_2_PAH_FACRB_BS_1.scr
- Test Applet: API_2_PAH_FACRB_BS_1.java
- Installation parameter: API_2_PAH_FACRB_BS.install
- Load Script: API_2_PAH_FACRB_BS.ldr
- Conversion parameter: API_2_PAH_FACRB_BS.cnv

6.1.7.16.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler		
	findAndCompareValue() with a null dstBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15		
	compareOffset ≥ compareBuffer.length **`tag = 0Dh`** **`compareBuffer.length = 20`** **`compareOffset = 20`**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	compareOffset < 0 **`compareBuffer.length = 20`** **`compareOffset = -1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	length > compareBuffer.length **`compareBuffer.length = 15`** **`compareOffset = 0`**	`ArrayIndexOutOfBoundsException` is thrown	
5	compareOffset + length > compareBuffer.length **`compareBuffer.length = 20`** **`compareOffset = 5`**	`ArrayIndexOutOfBoundsException` is thrown	
6	InitDisplayText()		
	Select a TLV (tag 02h)		
	findAndCompareValue() **`tag = 03h`**	`ToolkitException.UNAVAILABLE_ELEMENT` is thrown	
	Call the `getValueLength()` method	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
7	initDisplayText() **`dcs = 4`** **`buffer = 00 01 … 0F`**		

	Initialise compareBuffer **compareBuffer =** **04 00 01 … 0F**		
	Compare buffers **tag = 0Dh** **compareOffset = 0**	**Result is 00h**	
8	Verify current TLV **getValueLength()**	**Result is 17**	
9	Initialise compareBuffer **compareBuffer =** **04 00 01 …** 10		
	Compare buffers with same parameters	**Result is -1**	
10	Initialise compareBuffer **compareBuffer =** 03 **00 01 … 0F**		
	Compare buffers with same parameters	**Result is +1**	
11	Initialise compareBuffer **compareBuffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**		
	Compare buffers **compareOffset = 2**	**Result is 00h**	
12	append a Text String TLV **tag = 0Dh** **buffer = 00 11 22 33 44 55**		
	Initialise compareBuffer **compareBuffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**		
	Compare buffers **compareOffset = 2**	**Result is 00h**	
13	Initialise compareBuffer **compareBuffer =** **55 55 04** 01 **01** **02 03 04 05 06** **07 08 09 0A 0B**		

	0C 0D 0E 0F 55		
	Compare buffers **compareOffset = 2**	**Result is -1**	
14	Initialise compareBuffer **compareBuffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D** 0D 10 **55**		
	Compare buffers **compareOffset = 2**	**Result is +1**	
15	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Initialise compareBuffer **CompareBuffer = 04 00 01 … 0F**		
	Successful call (with tag 8Dh) **tag = 8Dh** **compareBuffer.length = 17** **compareOffset = 0**	**Result is 00h**	
16	Append tag 0Fh **buffer = 00 01 … 0F**		
	Initialise compareBuffer **compareBuffer = 00 01 … 0F**		
	Successful call (with tag 8Fh) **tag = 8Fh** **compareBuffer.length = 16** **compareOffset = 0**	**Result is 00h**	

6.1.7.16.5 Test Coverage

CRR number	Test case number
N1	6
N2	8
N3	7, 11, 12
N4	9, 13
N5	10, 14
N6	15, 16
P1	1
P2	2, 3, 4, 5
C1	Does not apply for Proactive Handler

6.1.7.17 Method findAndCompareValue

6.1.7.17.1 Test Area Reference API_2_PAH_FACRBBS_BSS

6.1.7.17.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte findAndCompareValue(byte tag,
                                byte occurence,
                                short valueOffset,
                                byte[] compareBuffer,
                                short compareOffset,
                                short compareLength)
                         throws java.lang.NullPointerException,
                                java.lang.ArrayIndexOutOfBoundsException,
                                ToolkitException
```

Normal Execution

Looks for the indicated occurrence of a TLV element from the beginning of a TLV list and compare its value with a buffer:

CRRN1: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN2: if the method is successful then the corresponding TLV becomes current.

CRRN3: if identical 0 is returned.

CRRN4: if the first miscomparing byte in simple TLV is less than that in compareBuffer -1 is returned.

CRRN5: if the first miscomparing byte in simple TLV is greater than that in compareBuffer 1 is returned

CRRN6: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset or compareLength or both would cause access outside array bounds, or if compareLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, compareLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

CRRP4: if an input parameter is not valid (e.g. occurence = 0) an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException BAD_INPUT_PARAMETER.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.7.17.3 Test Suite files

- Test Script: API_2_PAH_FACRBBS_BSS_1.scr
- Test Applet: API_2_PAH_FACRBBS_BSS_1.java
- Installation parameter: API_2_PAH_FACRBBS_BSS.install

- Load Script: API_2_PAH_FACRBBS_BSS.ldr
- Conversion parameter: API_2_PAH_FACRBBS_BSS.cnv

6.1.7.17.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler		
	findAndCompareValue() with a null compareBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15		
	compareOffset ≥ compareBuffer.length **tag = 0Dh, occurrence = 1** **valueOffset = 0** **compareBuffer.length = 5** **compareOffset = 5** **compareLength = 1**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	compareOffset < 0 **compareBuffer.length = 5** **compareOffset = -1** **compareLength = 1**	`ArrayIndexOutOfBoundsException` is thrown	
4	compareLength >compareBuffer.length **compareBuffer.length = 5** **compareOffset = 0** **compareLength = 6**	`ArrayIndexOutOfBoundsException` is thrown	
5	compareOffset + compareLength >compareBuffer.length **compareBuffer.length = 5** **compareOffset = 3** **compareLength = 3**	`ArrayIndexOutOfBoundsException` is thrown	
6	compareLength < 0 **compareBuffer.length = 5** **compareOffset = 0** **compareLength = -1**	`ArrayIndexOutOfBoundsException` is thrown	
7	initDisplayText() with length = 5		
	valueOffset ≥ Text String Length **tag = 0Dh, occurrence = 1** **valueOffset = 6** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	
8	valueOffset < 0 **valueOffset = -1**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	

	compareBuffer.length = 15 **compareOffset = 0** **compareLength = 1**		
9	compareLength > Text String length **valueOffset = 0** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 7**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
10	valueOffset + compareLength > Text String length **valueOffset = 2** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 5**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
11	Invalid parameter **occurrence = 0**	**ToolkitException.BAD_INPUT_PARAMETER is thrown**	
12	InitDisplayText()		
	Select a TLV (tag 02h)		
	findAndCompareValue() **tag = 0Dh** **occurrence =** 2	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the `getValueLength()` method	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
13	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Initialise compareBuffer **compareBuffer =** **04 00 01 … 0F**		
	findAndCompareValue() **tag = 0Dh, occurrence = 1** **valueOffset = 0** **compareOffset = 0** **compareLength = 17**	**Result is 00h**	
14	Verify current TLV **getValueLength()**	**Result is 17**	
15	Initialise compareBuffer **compareBuffer =** **04 00 01 …** 10		
	Compare buffers with same parameters	**Result is -1**	

16	Initialise compareBuffer **compareBuffer =** 03 **00 01 … 0F**		
	Compare buffers with same parameters	**Result is +1**	
17	Initialise compareBuffer **compareBuffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**		
	Compare buffers **valueOffset = 2** **compareOffset = 3** **compareLength = 12**	**Result is 00h**	
18	Initialise compareBuffer **compareBuffer =** **55 55 55** 02 01 **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is -1**	
19	Initialise compareBuffer **compareBuffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A** 0A 0D **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is +1**	
20	append a Text String TLV **tag = 0Dh** **buffer = 00 11 22 33 44 55**		
	Initialise compareBuffer **compareBuffer =** **04 00 01 … 0F**		
	findAndCompareValue() **tag = 0Dh, occurrence = 1** **valueOffset = 0** **compareOffset = 0** **compareLength = 17**	**Result is 00h**	

21	Initialise compareBuffer **compareBuffer =** **00 11 22 33 44 55**		
	findAndCompareValue() **tag = 0Dh, occurrence = 2** **valueOffset = 0** **compareOffset = 0** **compareLength = 6**	**Result is 00h**	
22	Initialise compareBuffer **compareBuffer =** **00 11 22 33 44** 66		
	findAndCompareValue() **tag = 0Dh, occurrence = 2** **valueOffset = 0** **compareOffset = 0** **compareLength = 6**	**Result is -1**	
23	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Initialise compareBuffer **CompareBuffer = 04 00 01 … 0F**		
	Successful call (with tag 8Dh) **tag = 8Dh** **compareBuffer.length = 17** **compareOffset = 0**	**Result is 00h**	
24	Append tag 0Fh **buffer = 00 01 … 0F**		
	Initialise compareBuffer **compareBuffer = 00 01 … 0F**		
	Successful call (with tag 8Fh) **tag = 8Fh** **compareBuffer.length = 16** **compareOffset = 0**	**Result is 00h**	

6.1.7.17.5 Test Coverage

CRR number	Test case number
N1	12
N2	14
N3	13, 17, 20, 21
N4	15, 18, 22
N5	16, 19
N6	23, 24

P1	**1**
P2	**2, 3, 4, 5, 6**
P3	**7, 8, 9, 10**
P4	**11**
C1	**Does not apply for Proactive Handler**

6.1.8 Class ProactiveResponseHandler

6.1.8.1 Method copyAdditionalInformation

6.1.8.1.1 Test Area Reference: API_2_PRH_CPAI_BSS

6.1.8.1.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short copyAdditionalInformation(byte[] dstBuffer,
                                       short dstOffset,
                                       short dstLength)
                                throws java.lang.NullPointerException,
                                java.lang.ArrayIndexOutOfBoundsException,
                                ToolkitException
```

Normal Execution

CRRN1: The `copyAdditionalInformation()` method shall copy a part of the additional information field from Result TLV element in `dstBuffer`, using `dstOffset` and `dstLength`.

CRRN2: `dstBuffer` shall only be modified from `dstOffset` to `(dstOffset + dstLength – 1)` (included).

CRRN3: The method returns (`dstOffset + dstLength`).

CRRN4: If a Result TLV element is available, it becomes the TLV selected after a call to the method.

CRRN5: The method shall copy from the first Result TLV.

Parameter Error

CRRP1: A `NullPointerException` shall be thrown if `dstBuffer` is `null`.

CRRP2: An `ArrayIndexOutOfBoundsException` shall be thrown if `dstOffset` or `dstLength` or both would cause access outside array bounds.

Context Error

CRRC1: A `ToolkitException.UNAVAILABLE_ELEMENT` shall be thrown in case of unavailable Result TLV element.

CRRC2: A `ToolkitException.OUT_OF_TLV_BOUNDARIES` shall be thrown if `dstLength` is greater than the value field of the available TLV.

6.1.8.1.3 Test Suite files

- Test Script: API_2_PRH_CPAI_BSS_1.scr
- Test Applet: API_2_PRH_CPAI_BSS_1.java
- Installation parameter: API_2_PRH_CPAI_BSS.install
- Load Script: API_2_PRH_CPAI_BSS.ldr
- Conversion parameter: API_2_PRH_CPAI_BSS.cnv

6.1.8.1.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Build and send a DISPLAY TEXT command** **qualifier = 0** **dcs = 4** **buffer = "Text"**		DISPLAY TEXT Pro command
	Terminal Response with 11 additional bytes **Result TLV = 03 0C 01** 01 23 45 67 89 AB CD EF 01 23 45		
	NULL as parameter to dstBuffer **dstBuffer = NULL**	`NullPointerException` is thrown	
2	**dstOffset ≥ dstBuffer.length** **dstBuffer.length = 10** **dstOffset = 10** **dstLength = 1**	`ArrayIndexOutOfBoundsException` is thrown	
3	**dstOffset < 0** **dstBuffer.length = 10** **dstOffset = -1** **dstLength = 1**	`ArrayIndexOutOfBoundsException` is thrown	
4	**dstLength > dstBuffer.length** **dstBuffer.length = 10** **dstOffset = 0** **dstLength = 11**	`ArrayIndexOutOfBoundsException` is thrown	
5	**dstOffset + dstLength > dstBuffer.length** **dstBuffer.length = 10** **dstOffset = 6** **dstLength = 5**	`ArrayIndexOutOfBoundsException` is thrown	
6	**dstLength < 0** **dstBuffer.length = 10** **dstOffset = 6** **dstLength = -1**	`ArrayIndexOutOfBoundsException` is thrown	
7	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**
	Terminal Response with 5 additional bytes **Result TLV = 03 06 01** 01 23 45 67 89		
	Successfull call, dstBuffer is the whole buffer **dstBuffer.length = 5** **dstOffset = 0** **dstLength = 5**	result of `copyAdditionalInformation()` is **05h.**	
8	**Compare dstBuffer using arrayCompare()** **src = {01, 23, 45, 67, 89}** **srcOffset = 00** **dest = dstBuffer** **destOffset = 0** **length = 5**	result of `arrayCompare()` is **00h**.	
9	**Call the getValueLength() method**	Result is **06h.**	

10	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**
	Terminal Response with 6 additional bytes **Result TLV = 03 07 01** AB CD EF FE DC BA		
	Successfull call, `dstBuffer` is part of a buffer **dstBuffer.length = 7** **dstOffset = 2** **dstLength = 5**	result of `copyAdditionalInformation()` is **07h.**	
11	**Compare `dstBuffer` using `arrayCompare()`** **src = {AB, CD, EF, FE, DC}** **srcOffset = 00** **dest = dstBuffer** **destOffset = 2** **length = 5**	result of `arrayCompare()` is **00h**.	
12	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**
	Terminal Response with 7 additional bytes **Result TLV = 03 08 01** FE DC BA 98 76 54 32		
	Successfull call, `dstBuffer` is part of a buffer **dstBuffer.length = 7** **dstOffset = 0** **dstLength = 5**	result of `copyAdditionalInformation()` is **05h.**	
13	**Compare `dstBuffer` using `arrayCompare()`** **src = {FE, DC, BA, 98, 76}** **srcOffset = 00** **dest = dstBuffer** **destOffset = 0** **length = 5**	result of `arrayCompare()` is **00h**.	
14	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**
	Terminal Response with 8 additional bytes **Result TLV = 03 09 01** 00 11 22 33 44 55 66 77		
	Successfull call, `dstBuffer` is the whole buffer **dstBuffer.length = 9** **dstOffset = 2** **dstLength = 5**	result of `copyAdditionalInformation()` is **07h.**	
15	**Compare `dstBuffer` using `arrayCompare()`** **src = {00, 11, 22, 33, 44}** **srcOffset = 00** **dest = dstBuffer** **destOffset = 2** **length = 5**	result of `arrayCompare()` is **00h**.	
16	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**

	Terminal Response with F2h additional bytes **Result TLV = 03 81 F3 01** 00 01 02 03…		
	Successfull call to the method **dstBuffer.length = F2h** **dstOffset = 0** **dstLength = F2h**	result of `copyAdditionalInformation()` is **F2h.**	
17	**Compare `dstBuffer` using `arrayCompare()`** **src = {00, 01, 02, 03, 04…}** **srcOffset = 00** **dest = dstBuffer** **destOffset = 0** **length = F2h**	result of `arrayCompare()` is **00h**.	
18	**Call the getValueLength() method**	Result is **F3h.**	
19	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**
	Terminal Response with 5 additional bytes **Result TLV = 03 06 01** 00 11 22 33 44		
	`dstLength` > data available **dstBuffer.length = 6** **dstOffset = 0** **dstLength = 6**	`OUT_OF_TLV_BOUNDARIES` `ToolkitException` is thrown	
20	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**
	Terminal Response with 5 additional bytes **Result TLV = 03 06 01** 00 11 22 33 44		
	Initialise `dstBuffer` **dstBuffer = {00h, 01h, 02h, 03h…}**		
	Call the `copyAdditionalInformation()` method **dstBuffer.length = 20** **dstOffset = 5** **dstLength = 5**		
	Compare `dstBuffer` using `arrayCompare()` **src = {** 00h, 01h, 02h, 03h, 04h, 00h, 11h, 22h, 33h, 44h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh, 10h, 11h, 12h, 13h} srcOffset = 0 dest = dstBuffer destOffset = 0 length = 20	result of `arrayCompare()` is **00h**	

21	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**
	Terminal Response with 2 Result TLV elements **1st Result TLV = 03 06 01** 01 23 45 67 89 **2nd Result TLV = 03 01 00**		
	Successfull call to `copyAdditionalInformation()` **`dstBuffer.length = 5`** **`dstOffset = 0`** **`dstLength = 5`**	result of `copyAdditionalInformation()` is **05h.**	
22	**Compare `dstBuffer` using `arrayCompare()`** **`src = {01, 23, 45, 67, 89}`** **`srcOffset = 00`** **`dest = dstBuffer`** **`destOffset = 0`** **`length = 5`**	result of `arrayCompare()` is **00h**.	
23	**Call the getValueLength() method**	Result is **06h.**	

6.1.8.1.5 Test Coverage

CRR number	Test case number
N1	**8, 11, 13, 15, 17, 20, 22**
N2	**20**
N3	**7, 10, 12, 14, 16, 21**
N4	**9, 18, 23**
N5	**21, 22, 23**
P1	**1**
P2	**2, 3, 4, 5, 6**
C1	
C2	**19**

6.1.8.2 Method copyTextString

6.1.8.2.1 Test Area Reference: API_2_PRH_CPTS_BS

6.1.8.2.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short copyTextString(byte[] dstBuffer,
                            short dstOffset)
                     throws java.lang.NullPointerException,
                            java.lang.ArrayIndexOutOfBoundsException,
                            ToolkitException
```

Normal Execution

CRRN1: The `copyTextString()` method copies the text string value from the first Text String TLV element, using `dstBuffer` and `dstOffset`.

CRRN2: If a Text String TLV element is available, it becomes the TLV selected.

CRRN3: The method returns (`dstOffset` + length of copied value).

Parameter Error

CRRP1: A `NullPointerException` shall be thrown if `dstBuffer` is `null`.

CRRP2: A `ArrayIndexOutOfBoundsException` shall be thrown if `dstOffset` would cause access outside array bounds.

Context Error

CRRC1: A `ToolkitException.UNAVAILABLE_ELEMENT` shall be thrown in case of unavailable Text String TLV element.

6.1.8.2.3 Test Suite files

- Test Script: API_2_PRH_CPTS_BS_1.scr
- Test Applet: API_2_PRH_CPTS_BS_1.java
- Installation parameter: API_2_PRH_CPTS_BS.install
- Load Script: API_2_PRH_CPTS_BS.ldr
- Conversion parameter: API_2_PRH_CPTS_BS.cnv

6.1.8.2.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	**Build and send a GET INPUT command**
	`qualifier = 00h`
	`dcs = 04h`
	`buffer = 'Text'`
	`minRespLength = 00h`
	`maxRespLength = FFh`
			GET INPUT Pro command

	Terminal Response **Text String TLV = 0D 02 04 41**		
	Get the Response ; call the copyTextString() method with a null dstBuffer **dstBuffer = null** **dstOffset = 0**	`NullPointerException` is thrown	
2	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response **Text String TLV = 0D 05 04 "ABC"**		
	dstOffset + text length > dstBuffer.length **dstBuffer.length =** 04h **dstOffset =** 02h	`ArrayIndexOutOfBoundsException` is thrown	

3	**dstOffset < 0** **dstBuffer.length =** 04h **dstOffset =** -1	ArrayIndexOutOfBoundsException is thrown	
4	**Build and send a DISPLAY TEXT command** qualifier = 00h dcs = 04h buffer = 'Text'		**DISPLAY TEXT Pro command**
	Terminal Response without Text String TLV		
	Get the Response ; call the copyTextString() method	UNAVAILABLE_ELEMENT ToolkitException is thrown	
5	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with a null Text String TLV **Text String TLV = 0D 00**		
	Initialise dstBuffer **dstBuffer = {00h, 01h, 02h, 03h}**		
	Call the copyTextString() method **dstBuffer.length =** 04h **dstOffset =** 02h	Result of copyTextString() is **02h**	
6	**Compare dstBuffer using arrayCompare()** **src = {00h, 01h, 02h, 03h}** srcOffset = 00h dest = dstBuffer destOffset = 00h length = 04h	Result of arrayCompare() is **00h**	
7	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = 01h **Text String TLV = 0D 02 04 41**		
	Initialise dstBuffer **dstBuffer = {00h, 01h, 02h, 03h}**		
	Call the copyTextString() method **dstBuffer.length =** 04h **dstOffset =** 00h	Result of copyTextString() is **01h**	
8	**Compare dstBuffer using arrayCompare()** **src = {41h, 01h, 02h, 03h}**	Result of arrayCompare() is **00h**	

	srcOffset = 00h dest = dstBuffer destOffset = 00h length = 04h		
9	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = 02h **Text String TLV = 0D 03 04 42 43**		
	Initialise dstBuffer **dstBuffer = {00h, 01h, 02h, 03h}**		
	Call the copyTextString() method **dstBuffer.length =** 04h **dstOffset =** 02h	Result of copyTextString() is **04h**	
10	**Compare dstBuffer using arrayCompare()** **src = {00h, 01h, 42h, 43h}** srcOffset = 00h dest = dstBuffer destOffset = 00h length = 04h	Result of arrayCompare() is **00h**	
11	**Call the getValueLength() method**	Result is **03h**	
12	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = 7Eh **Text String TLV = 0D 7F 04 01 02 … 7E**		
	Initialise dstBuffer **dstBuffer = {00h, 00h … 00h}**		
	Call the copyTextString() method **dstBuffer.length =** 7Eh **dstOffset =** 00h	Result of copyTextString() is **7Eh**	
13	**Compare dstBuffer using arrayCompare()** **src = {01h, …, 7Eh}** srcOffset = 00h dest = dstBuffer destOffset = 00h length = 7Eh	Result of arrayCompare() is **00h**	
14	**Call the getValueLength() method**	Result is **7Fh**	

15	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = 7Fh **Text String TLV = 0D 81 80 04 01 02 …7F**		
	Initialise dstBuffer **dstBuffer = {00h, 01h … FFh}**		
	Call the copyTextString() method **dstBuffer.length =** FFh **dstOffset =** 10h	Result of `copyTextString()` is **8Fh**	
16	**Compare dstBuffer using arrayCompare()** **src = {00h, 01h,… 0Fh, 01h, …7Fh, 8Fh, … FFh}** srcOffset = 00h dest = dstBuffer destOffset = 00h length = FFh	Result of `arrayCompare()` is **00h**	
17	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = EFh **Text String TLV = 0D 81 F0 04 01 02 … EF**		
	Initialise dstBuffer **dstBuffer = {00h, 00h … 00h}**		
	Call the copyTextString() method **dstBuffer.length =** FFh **dstOffset =** 00h	Result of `copyTextString()` is **EFh**	
18	**Compare dstBuffer using arrayCompare()** **src = {01h, …EFh, 00h … 00h }** srcOffset = 00h dest = dstBuffer destOffset = 00h length = FFh	Result of `arrayCompare()` is **00h**	
19	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with two Text String TLV **1st Text String TLV = 0D 03 04 42 43** **2nd Text String TLV = 0D 02 04 44**		
	Initialise dstBuffer **dstBuffer = {00h, 01h, 02h, 03h}**		

	Call the copyTextString() method **dstBuffer.length =** 04h **dstOffset =** 02h	Result of copyTextString() is **04h**	
20	**Compare dstBuffer using arrayCompare()** **src = {00h, 01h, 42h, 43h}** srcOffset = 00h dest = dstBuffer destOffset = 00h length = 04h	Result of arrayCompare() is **00h**	
21	**Call the getValueLength() method**	Result is **03h**	

6.1.8.2.5 Test Coverage

CRR number	Test case number
N1	6, 8, 10, 13, 16, 18, 20
N2	11, 14, 21
N3	5, 7, 9, 12, 15, 17, 19
P1	1
P2	2, 3
C1	4

6.1.8.3 Method getAdditionalInformationLength

6.1.8.3.1 Test Area Reference: API_2_PRH_GTIL

6.1.8.3.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short getAdditionalInformationLength()
                                throws ToolkitException
```

Normal Execution

CRR1: This method returns the length of the additional information field from the first Result TLV in the ProactiveResponseHandler.

CRR2: After a successful execution of the method, the Result TLV becomes the selected TLV of the ProactiveResponseHandler.

Parameter Error

No requirements

Context Error

CRR3: A ToolkitException.UNAVAILABLE_ELEMENT shall be thrown in case of unavailable Result TLV element.

6.1.8.3.3 Test Suite files

- Test Script: API_2_PRH_GTIL_1.scr
- Test Applet: API_2_PRH_GTIL _1.java
- Installation parameter: API_2_PRH_GTIL.install
- Load Script: API_2_PRH_GTIL.ldr
- Conversion parameter: API_2_PRH_GTIL.cnv

6.1.8.3.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	**Build and send a DISPLAY TEXT command**
	`qualifier = 00h`
	`dcs = 04h`
	`buffer = 'Text'`
			DISPLAY TEXT Pro command

	Terminal Response without additional information		
	`Get the Response ; call the getAdditionalInformationLength() method`	Result is **00h**	
2	**Call the `getValueLength()` method**	Result is **01h**	
3	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**
	Terminal Response with 1 additional byte **`Result TLV = 03 02 02 55`**		
	Get the Response ; call the `getAdditionalInformationLength()` method	Result is **01h**	
4	**Call the `getValueLength()` method**	Result is **02h**	
5	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**
	Terminal Response with 7Eh additional bytes `Result TLV = 03 7F 02 55 55 55 …`		
	Get the Response ; call the `getAdditionalInformationLength()` method	Result is **7Eh**	
6	**Call the `getValueLength()` method**	Result is **7Fh**	
7	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**
	Terminal Response with 7Fh additional bytes		

	`Result TLV = 03 81 80 02 55 55 55 ...`		
	Get the Response ; call the `getAdditionalInformationLength()` method	Result is **7Fh**	
8	**Call the `getValueLength()` method**	Result is **80h**	
9	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**
	Terminal Response with 80h additional bytes `Result TLV = 03 81 81 02 55 55 55 ...`		
	Get the Response ; call the `getAdditionalInformationLength()` method	Result is **80h**	
10	**Call the `getValueLength()` method**	Result is **81h**	
11	Build and send a DISPLAY TEXT command		DISPLAY TEXT Pro command
	Terminal Response with F2h additional bytes `Result TLV = 03 81 F3 02 55 55 55 ...`		
	Get the Response ; call the `getAdditionalInformationLength()` method	Result is **F2h**	
12	**Call the `getValueLength()` method**	Result is **F3h**	
13	Build and send a DISPLAY TEXT command		DISPLAY TEXT Pro command
	Terminal Response with 2 Result TLV 1st `Result TLV = 03 03 02 01 23` 2nd `Result TLV = 03 01 00`		
	Get the Response ; call the `getAdditionalInformationLength()` method	Result is **02h**	
14	**Call the `getValueLength()` method**	Result is 0**3h**	

6.1.8.3.5 Test Coverage

CRR number	Test case number
N1	**1, 3, 5, 7, 9, 11, 13**
N2	**2, 4, 6, 8, 10, 12, 14**
C1	

6.1.8.4 Method getGeneralResult

6.1.8.4.1 Test Area Reference: API_2_PRH_GTGR

6.1.8.4.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte getGeneralResult()
                        throws ToolkitException
```

Normal Execution

CRRN1: This method returns the general result of a proactive command.

CRRN2: After a successful execution of the method, the Result TLV becomes the selected TLV of the ProactiveResponseHandler.

Parameter Error

No requirements

Context Error

CRRC1: A `ToolkitException.UNAVAILABLE_ELEMENT` shall be thrown in case of unavailable Result TLV element.

6.1.8.4.3 Test Suite files

- Test Script: API_2_PRH_GTGR_1.scr
- Test Applet: API_2_PRH_GTGR_1.java
- Installation parameter: API_2_PRH_GTGR.install
- Load Script: API_2_PRH_GTGR.ldr
- Conversion parameter: API_2_PRH_GTGR.cnv

6.1.8.4.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	**Build and send a DISPLAY TEXT command**
	`qualifier = 00h`
	`dcs = 04h`
	`buffer = 'Text'`
			DISPLAY TEXT Pro command

	Terminal Response with General Result = 00 (command performed successfully)		
	`Get the Response` **`Call the getGeneralResult() method`**	Result of `getGeneralResult()` is **00h**	
2	**Call the `getValueLength()` method**	Result is **01h**	
3	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**

	Terminal Response with General Result = 01, without Additional information on result (command performed with partial comprehension)		
	`Get the Response` Call the `getGeneralResult()` method	Result of `getGeneralResult()` is **01h**	
4	**Call the `getValueLength()` method**	Result is **01h**	
5	Build and send a DISPLAY TEXT command		**DISPLAY TEXT Pro command**
	Terminal Response with General Result = 01, with Additional information on result `Result TLV = 03 02 01 55 (command performed with partial comprehension)`		
	`Get the Response` Call the `getGeneralResult()` method	Result of `getGeneralResult()` is **01h**	
6	**Call the `getValueLength()` method**	Result is **02h**	
7	**Build and send a DISPLAY TEXT command**		**DISPLAY TEXT Pro command**
	Terminal Response with General Result = 02 `Result TLV = 03 04 02 65 43 21 (Missing information)`		
	`Get the Response` Call the `getGeneralResult()` method	Result of `getGeneralResult()` is **02h**	
8	**Call the `getValueLength()` method**	Result is **04h**	
9	**Build and send a DISPLAY TEXT command**		**DISPLAY TEXT Pro command**
	Terminal Response with 7Fh additional bytes `Result TLV = 03 81 80 02 55 55 55 …`		
	Get the Response ; call the `getGeneralResult()` method	Result is **02h**	
10	**Call the `getValueLength()` method**	Result is **80h**	

11	**Build and send a DISPLAY TEXT command**		**DISPLAY TEXT Pro command**
	Terminal Response with 2 Result TLV **1st Result TLV = 03 02** 02 **12** 2nd Result TLV = 03 03 **03** 34 56		
	Get the Response ; call the `getGeneralResult()` method	Result is **02h**	
12	**Call the `getValueLength()` method**	Result is **02h**	

6.1.8.4.5 Test Coverage

CRR number	Test case number
N1	**1, 3, 5, 7, 9**
N2	**2, 4, 6, 8, 10**
C1	

6.1.8.5 Method getItemIdentifier

6.1.8.5.1 Test Area Reference: API_2_PRH_GTII

6.1.8.5.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte getItemIdentifier()
                        throws ToolkitException
```

Normal Execution

CRRN1: The method returns the item identifier byte value from the first Item Identifier TLV element.

CRRN2: If an Item Identifier TLV element is available, it becomes the TLV selected.

Parameter Error

No requirements

Context Error

CRRC1: A `ToolkitException.UNAVAILABLE_ELEMENT` shall be thrown in case of unavailable Item Identifier TLV element.

6.1.8.5.3 Test Suite files

- Test Script: API_2_PRH_GTII_1.scr
- Test Applet: API_2_PRH_GTII_1.java
- Installation parameter: API_2_PRH_GTII.install
- Load Script: API_2_PRH_GTII.ldr
- Conversion parameter: API_2_PRH_GTII.cnv

6.1.8.5.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Build and send a DISPLAY TEXT command**		**DISPLAY TEXT Pro command**
	Terminal Response (no Item Identifier TLV available)		
	`Call to getItemIdentifier() with unavailable Item Identifier TLV`	`UNAVAILABLE_ELEMENT ToolkitException` is thrown	
2	**Build and send a SELECT ITEM command with 2 items (ID=01, 02)**		**SELECT ITEM Pro command**
	Terminal Response with Item 1 selected Item Identifier TLV = `10 01 01`		
	Call the `getItemIdentifier()` method	Result is **01h**	
3	**Call the `getValueByte()` method** **`valueOffset = 00h`**	Result is **01h**	
4	**Build and send a SELECT ITEM command with 3 items (ID=03, 05, 07)**		**SELECT ITEM Pro command**
	Terminal Response with Item 5 selected Item Identifier TLV = `10 01 05`		
	Call the `getItemIdentifier()` method	Result is **05h**	
5	**Call the `getValueByte()` method** **`valueOffset = 00h`**	Result is **05h**	
6	**Build and send a SELECT ITEM command with 3 items (ID=FDh, FEh, FFh)**		**SELECT ITEM Pro command**
	Terminal Response with Item FFh selected Item Identifier TLV `= 10 01 FF`		
	Call the `getItemIdentifier()` method	Result is **FFh**	
7	**Call the `getValueByte()` method** **`valueOffset = 00h`**	Result is **FFh**	
8	**Build and send a SELECT ITEM command with 3 items (ID=FDh, FEh, FFh)**		**SELECT ITEM Pro command**

	Terminal Response with 2 Item Identifier TLV 1st Item Identifier TLV = `10 01 FFh` 2nd Item Identifier TLV = `10 01 FEh`		
	Call the `getItemIdentifier()` method	Result is **FFh**	
9	**Call the `getValueByte()` method** **`valueOffset = 00h`**	Result is **FFh**	

6.1.8.5.5 Test Coverage

CRR number	Test case number
N1	**2, 4, 6, 8**
N2	**3, 5, 7, 9**
C1	**1**

6.1.8.6 Method getTextStringCodingScheme

6.1.8.6.1 Test Area Reference: API_2_PRH_GTCS

6.1.8.6.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte getTextStringCodingScheme()
                                throws ToolkitException
```

Normal Execution

CRRN1: This method returns the data coding scheme from the first Text String TLV element.

CRRN2: If a Text String TLV element is available, it becomes the TLV selected.

Parameter Error

No requirements

Context Error

CRRC1: A `ToolkitException.UNAVAILABLE_ELEMENT` shall be thrown in case of unavailable Text String TLV element.

CRRC2: A `ToolkitException.OUT_OF_TLV_BOUNDARIES` shall be thrown if the Text String TLV is present with a length of 0.

6.1.8.6.3 Test Suite files

- Test Script: API_2_PRH_GTCS_1.scr
- Test Applet: API_2_PRH_GTCS_1.java
- Installation parameter: API_2_PRH_GTCS.install
- Load Script: API_2_PRH_GTCS.ldr
- Conversion parameter: API_2_PRH_GTCS.cnv

6.1.8.6.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Build and send a DISPLAY TEXT command**		**DISPLAY TEXT Pro command**
	Terminal Response (no Text String TLV element available)		
	`Call to getTextStringCodingScheme() with unavailable Text String TLV`	`UNAVAILABLE_ELEMENT ToolkitException` is thrown	
2	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with a null Text String TLV **`6.1.8.6.4.1.1 Text String TLV = 0D 00`**		
	Call the `getTextStringCodingScheme()` method	`OUT_OF_TLV_BOUNDARIES ToolkitException` is thrown	
3	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = 01h, DCS = 04h **`6.1.8.6.4.1.2 Text String TLV = 0D 02 04 "A"`**		
	Call the `getTextStringCodingScheme()` method	Result is **04h**	
4	**Call the `getValueLength()` method**	Result is **02h**	
5	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = 02h, DCS = 00h **`6.1.8.6.4.1.3 Text String TLV = 0D`**		

	03 00 "BB"		
	Call the `getTextStringCodingScheme()` method	Result is **00h**	
6	**Call the `getValueLength()` method**	Result is **03h**	
7	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = 7Eh, DCS = 08h **6.1.8.6.4.1.4 Text String TLV = 0D 7F 08 01 02 … 7E**		
	Call the `getTextStringCodingScheme()` method	Result is **08h**	
8	**Call the `getValueLength()` method**	Result is **7Fh**	
9	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = 7Fh, DCS = 04h **6.1.8.6.4.1.5 Text String TLV = 0D 81 80 04 01 02 … 7F**		
	Call the `getTextStringCodingScheme()` method	Result is **04h**	
10	**Call the `getValueLength()` method**	Result is **80h**	
11	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = EFh, DCS = 08h **6.1.8.6.4.1.6 Text String TLV = 0D 81 F0 08 01 02 … EE EF**		
	Call the `getTextStringCodingScheme()` method	Result is **08h**	

12	**Call the `getValueLength()` method**	Result is **F0h**	
13	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with 2 Text String TLV **`6.1.8.6.4.1.7`** **1st Text String TLV = `0D 02` `04` `41`** **`6.1.8.6.4.1.8`** **2nd Text String TLV = `0D 03` `08` `42 43`**		
	Call the `getTextStringCodingScheme()` method	Result is **04h**	
14	**Call the `getValueLength()` method**	Result is **02h**	

6.1.8.6.5 Test Coverage

CRR number	Test case number
N1	3, 5, 7, 9, 11, 13
N2	4, 6, 8, 10, 12, 14
C1	1
C2	2

6.1.8.7 Method GetTextStringLength

6.1.8.7.1 Test Area Reference: API_2_PRH_GTTL

6.1.8.7.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short getTextStringLength()
                        throws ToolkitException
```

Normal Execution

CRRN1: The `getTextStringLength()` method returns the text string length value from the first Text String TLV element.

CRRN2: If a Text String TLV element is available, it becomes the TLV selected.

Parameter Error

No requirements

Context Error

CRRC1: A `ToolkitException.UNAVAILABLE_ELEMENT` shall be thrown in case of unavailable Text String TLV element.

6.1.8.7.3 Test Suite files

- Test Script: API_2_PRH_GTTL_1.scr
- Test Applet: API_2_PRH_GTTL_1.java
- Installation parameter: API_2_PRH_GTTL.install
- Load Script: API_2_PRH_GTTL.ldr
- Conversion parameter: API_2_PRH_GTTL.cnv

6.1.8.7.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Build and send a DISPLAY TEXT command**		**DISPLAY TEXT Pro command**
	Terminal Response (no Text String TLV element available)		
	`Call to getTextStringLength() with unavailable Text String TLV`	`UNAVAILABLE_ELEMENT ToolkitException` is thrown	
2	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with a null Text String TLV **`6.1.8.7.4.1.1 Text String TLV = 0D 00`**		
	Call the `getTextStringLength()` method	Result is **00h**	
3	**Call the `getValueLength()` method**	Result is **00h**	
4	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = 01h, DCS = 04h **`6.1.8.7.4.1.2 Text String TLV = 0D 02 04 "A"`**		
	Call the `getTextStringLength()` method	Result is **01h**	
5	**Call the `getValueLength()` method**	Result is **02h**	
6	**Build and send a GET INPUT command**		**GET INPUT Pro command**

			command
	Terminal Response with text length = 02h, DCS = 00h **6.1.8.7.4.1.3 Text String TLV = 0D 03 00 "BB"**		
	Call the `getTextStringLength()` method	Result is **02h**	
7	**Call the `getValueLength()` method**	Result is **03h**	
8	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = 7Eh, DCS = 08h **6.1.8.7.4.1.4 Text String TLV = 0D 7F 08 01 02 … 7E**		
	Call the `getTextStringLength()` method	Result is **7Eh**	
9	**Call the `getValueLength()` method**	Result is **7Fh**	
10	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = 7Fh, DCS = 04h **6.1.8.7.4.1.5 Text String TLV = 0D 81 80 04 01 02 … 7F**		
	Call the `getTextStringLength()` method	Result is **7Fh**	
11	**Call the `getValueLength()` method**	Result is **80h**	
12	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with text length = EFh, DCS = 04h **6.1.8.7.4.1.6 Text String TLV = 0D**		

	81 F0 04 01 02 … EE EF		
	Call the `getTextStringLength()` method	Result is **EFh**	
13	**Call the `getValueLength()` method**	Result is **F0h**	
14	**Build and send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response with 2 Text String TLV **6.1.8.7.4.1.7 1st Text String TLV = `0D 02 04 41`** 2nd Text Sring TLV = `0D 03 08 42 43`		
	Call the `getTextStringLength()` method	Result is **01h**	
15	**Call the `getValueLength()` method**	Result is **02h**	

6.1.8.7.5 Test Coverage

CRR number	Test case number
1	**2, 4, 6, 8, 10, 12, 14**
2	**3, 5, 7, 9, 11, 13, 15**
3	**1**

6.1.8.8 Method getTheHandler

6.1.8.8.1 Test Area Reference: API_2_PRH_GTHD

6.1.8.8.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public static ProactiveResponseHandler getTheHandler()
                                        throws ToolkitException
```

Normal Execution

CRRN1: The method shall return the single system instance of the ProactiveHandler class.

Parameter Error

No requirements

Context Error

CRRC1: The method shall throw ToolkitException.HANDLER_NOT_AVAILABLE if the handler is busy.

6.1.8.8.3 Test Suite files

- Test Script: API_2_PRH_GTHD_1.scr
- Test Applet: API_2_PRH_GTHD_1.java

- Installation parameter: API_2_PRH_GTHD.install
- Load Script: API_2_PRH_GTHD.ldr
- Conversion parameter: API_2_PRH_GTHD.cnv

6.1.8.8.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**Trigger the applet with a PROFILE_DOWNLOAD**		
1	**Build and send a Proactive Command**		**Proactive Command**
	Terminal Response		
	getTheHandler() twice	**The returned objects shall be the same**	
2	getTheHandler()	The reference shall be a ProactiveResponseHandler	
3	getTheHandler()	The reference shall not be null	
4	Build and send a Proactive command **Do not send the Terminal Response**		Proactive Command
	Trig the applet one more time (CALL CONTROL) call the getTheHandler() method	HANDLER_NOT_AVAILABLE ToolkitException is thrown by getTheHandler()	

6.1.8.8.5 Test Coverage

CRR number	Test case number
N1	1, 2, 3
C1	4

6.1.8.9 Method getLength

6.1.8.9.1 Test Area Reference API_2_PRH_GLEN

6.1.8.9.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short getLength()
                    throws ToolkitException
```

Normal Execution

CRRN1: returns the length in bytes of the TLV list.

Parameter Error

No requirements

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.HANDLER_NOT_AVAILABLE.

6.1.8.9.3 Test Suite files

- Test Script: API_2_PRH_GLEN_1.scr
- Test Applet: API_2_PRH_GLEN_1.java
- Installation parameter: API_2_PRH_GLEN.install
- Load Script: API_2_PRH_GLEN.ldr
- Conversion parameter: API_2_PRH_GLEN.cnv

6.1.8.9.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Build and send a Display Text command**		**DISPLAY TEXT Pro command**
	Terminal Response without additional information in General Result TLV		
	Get the response **`getLength()`**	Result of `getLength()` is 12	
2	**Build and send a Display Text command**		**DISPLAY TEXT Pro command**
	Terminal Response with F2h additional information in General Result TLV		
	Get the response **`getLength()`**	Result of `getLength()` is FFh	

6.1.8.9.5 Test Coverage

CRR number	Test case number
N1	1, 2
C1	Does not apply for Proactive Response Handler

6.1.8.10 Method copy

6.1.8.10.1 Test Area Reference API_2_PRH_COPY_BSS

6.1.8.10.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short copy(byte[] dstBuffer,
                  short dstOffset,
                  short dstLength)
```

```
                    throws java.lang.NullPointerException,
                         java.lang.ArrayIndexOutOfBoundsException,
                         ToolkitException
```

Normal Execution

CRRN1: copies the simple TLV list contained in the handler to the destination byte array.

CRRN2: returns dstOffset + dstLength.

Parameter Error

CRRP1: if dstBuffer is null a NullPointerException is thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative, an ArrayIndexOutOfBoundsException is thrown.

CRRP3: if dstLength is grater than the length of the simple TLV List, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException. OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.8.10.3 Test Suite files

- Test Script: API_2_PRH_COPY_BSS_1.scr
- Test Applet: API_2_PRH_COPY_BSS_1.java
- Installation parameter: API_2_PRH_COPY_BSS.install
- Load Script: API_2_PRH_COPY_BSS.ldr
- Conversion parameter: API_2_PRH_COPY_BSS.cnv

6.1.8.10.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a DISPLAY TEXT command**		**DISPLAY TEXT Pro command**
	Terminal Response without Additional Information in General Result TLV		
	Get the response **`copy()` with NULL as parameter to dstBuffer**	`NullPointerException` is thrown	
2	**`dstOffset ≥ dstBuffer.length`** **`dstBuffer.length = 5`** `dstOffset = 5` **`dstLength = 1`**	`ArrayIndexOutOfBoundsException` is thrown	

3	**`dstOffset < 0`**
	`dstBuffer.length = 5`
	`dstOffset = -1`
	`dstLength = 1`
		`ArrayIndexOutOfBoundsException` is thrown	

4	**`dstLength > dstBuffer.length`** **`dstBuffer.length = 5`**	`ArrayIndexOutOfBoundsException` is thrown	

	dstOffset = 0 dstLength = 6		
5	**dstOffset + dstLength > dstBuffer.length** **dstBuffer.length = 5** dstOffset = 3 dstLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength < 0** **dstBuffer.length = 5** dstOffset = 0 dstLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**dstLength > length of the simple TLV list** **dstBuffer.length = 13** dstOffset = 0 dstLength = 13	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**Successful call, dstBuffer is the whole buffer** **dstBuffer.length = 12** dstOffset = 0 dstLength = 12	Result of copy() is 12	
9	**Compare the buffer**	Result of arrayCompare() is 0	
10	**Successful call, dstBuffer is part of a buffer** **dstBuffer.length = 20** dstOffset = 3 dstLength = 12	Result of copy() is 15	
11	**Compare the whole buffer**	Result of arrayCompare() is 0	
12	**Successful call, dstBuffer is part of a buffer** **dstBuffer.length = 20** dstOffset = 3 dstLength = 9	Result of copy() is 12	
13	**Compare the whole buffer**	Result of arrayCompare() is 0	

6.1.8.10.5 Test Coverage

CRR number	Test case number
N1	9, 11, 13
N2	8, 10, 12
P1	1
P2	2, 3, 4, 5, 6
P3	7
C1	Does not apply for Proactive

	Response Handler

6.1.8.11 Method findTLV

6.1.8.11.1 Test Area Reference API_2_PRH_FINDBB

6.1.8.11.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte findTLV(byte tag, byte occurrence)
              throws ToolkitException
```

Normal Execution

Looks for the indicated occurrence of a TLV element from the beginning of the TLV list (handler buffer) :

CRRN1: the method is successful if the required occurrence exists then the corresponding TLV becomes current.

CRRN2: if the method is successful then it returns TLV_FOUND_CR_SET when Comprehension Required flag is set.

CRRN3: if the method is successful then it returns TLV_FOUND_CR_NOT_SET when Comprehension Required flag is not set.

CRRN4: if the required occurrence of the TLV element does not exist, no TLV is selected and TLV_NOT_FOUND is returned.

CRRN5: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if an input parameter is not valid (e.g. occurrence = 0) an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException BAD_INPUT_PARAMETER. The current TLV is no longer defined.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.8.11.3 Test Suite files

- Test Script: API_2_PRH_FINDBB_1.scr
- Test Applet: API_2_PRH_FINDBB_1.java
- Installation parameter: API_2_PRH_FINDBB.install
- Load Script: API_2_PRH_FINDBB.ldr
- Conversion parameter: API_2_PRH_FINDBB.cnv

6.1.8.11.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a DISPLAY TEXT command**		**DISPLAY TEXT Pro command**
	Terminal Response with 2 General Result TLV **01 03 01 21 00**		

	82 02 82 81 03 01 00 03 02 01 12		
	findTLV() with Invalid input parameter **occurrence = 0**	ToolkitException.BAD_INPUT_PARAMETER is thrown	
2	**Search 1st TLV** **tag = 01h** occurrence = 1	Result is TLV_FOUND_CR_NOT_SET	
3	**Call the getValueLength() method**	**Result is 03h**	
4	**Search 2nd TLV** **tag = 02h** occurrence = 1	**Result is TLV_FOUND_CR_SET**	
5	**Call the getValueLength() method**	**Result is 02h**	
6	**Search a wrong tag** **tag = 04h** occurrence = 1	**Result is TLV_NOT_FOUND**	
7	**Call the getValueLength() method**	**ToolkitException.UNAVAILABLE_ELEMENT shall be thrown**	
8	**Search a tag with wrong occurrence** **tag = 01h** occurrence = 2	**Result is TLV_NOT_FOUND**	
9	**Call the getValueLength() method**	**ToolkitException.UNAVAILABLE_ELEMENT shall be thrown.**	
10	**Search 3rd TLV** **tag = 03h** occurrence = **1**	Result is TLV_FOUND_CR_NOT_SET	
11	**Call the getValueLength() method**	**Result is 01h**	
12	**Search 3rd TLV** **tag = 03h** occurrence = **2**	Result is TLV_FOUND_CR_NOT_SET	
13	**Call the getValueLength() method**	**Result is 02h**	
14	**Search tag 81h** **tag = 81h** occurrence = **1**	Result is TLV_FOUND_CR_NOT_SET	
15	**Search tag 82h** **tag = 82h** occurrence = **1**	Result is TLV_FOUND_CR_SET	

6.1.8.11.5 Test Coverage

CRR number	Test case number
N1	3, 5, 11, 13
N2	2, 4

N3	**10, 12**
N4	**6, 7,8, 9**
N5	**14,15**
P1	**1**
C1	**Does not apply for Proactive Response Handler**

6.1.8.12 Method getValueLength

6.1.8.12.1 Test Area Reference: API_2_PRH_GVLEN

6.1.8.12.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short getValueLength()
                        throws ToolkitException
```

Normal Execution

CRRN1: gets and returns the binary length of the value field for the last TLV element which has been found in the handler.

Parameter Error

No requirements

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.8.12.3 Test Suite files

- Test Script: API_2_PRH_GVLEN_1.scr
- Test Applet: API_2_PRH_GVLEN_1.java
- Installation parameter: API_2_PRH_GVLEN.install
- Load Script: API_2_PRH_GVLEN.ldr
- Conversion parameter: API_2_PRH_GVLEN.cnv

6.1.8.12.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response **6.1.8.12.4.1.1Text String TLV = 0D 00**		
	Get the response	ToolkitException.UNAVAILABLE_ELEMENT is thrown	

	getValueLength()	LE_ELEMENT is thrown	
2	Search TLV 0Dh		
	getValueLength()	**Result is 00h**	
3	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response **Text String TLV = 0D 01 04 41**		
	Search TLV 0Dh (Text String TLV)		
	getValueLength()	**Result is 02h**	
4	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 7Eh **Text String TLV = 0D 7F 04 01 02 … 7E**		
	Search TLV 0Dh (Text String TLV)		
	getValueLength()	**Result is 7Fh**	
5	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 7Fh Text String TLV = 0D 81 80 04 01 02 … 7E 7F		
	Search TLV 0Dh (Text String TLV)		
	getValueLength()	**Result is 80h**	
6	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = EFh **Text String TLV = 0D 81 F0 04 01 02 … EF**		
	Search TLV 0Dh (Text String TLV)		
	getValueLength()	**Result is F0h**	

6.1.8.12.5 Test Coverage

CRR number	Test case number
N1	2, 3, 4, 5, 6
C1	Does not apply for Proactive Response

	Handler
C2	1

6.1.8.13 Method getValueByte

6.1.8.13.1 Test Area Reference API_2_PRH_GVBYTS

6.1.8.13.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte getValueByte(short valueOffset)
                         throws ToolkitException
```

Normal Execution

CRRN1: Gets a byte from the last TLV element which has been found in the handler and returns its value (1 byte).

Parameter Error

CRRP1: if valueOffset is out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.8.13.3 Test Suite files

- Test Script: API_2_PRH_GVBYTS_1.scr
- Test Applet: API_2_PRH_GVBYTS_1.java
- Installation parameter: API_2_PRH_GVBYTS.install
- Load Script: API_2_PRH_GVBYTS.ldr
- Conversion parameter: API_2_PRH_GVBYTS.cnv

6.1.8.13.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a GET INPUT command**		**GET INPUT Pro command**
	Terminal Response, Text String length = 7Eh **Text String TLV = 0D 7F 04 01 02 … 7E**		
	Get the response		
	`getValueByte(0)`	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	Search TLV `01h` (Command Details TLV)		
	getValueByte(3)	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	

3	Search TLV 01h (Command Details TLV)		
	getValueByte(2)	**Result is 00h (qualifier)**	
4	Search TLV 02h (Device Identities TLV)		
	getValueByte(0)	**Result is 82h (Source)**	
5	Search TLV 0Dh (Text String TLV)		
	getValueByte(7E)	**Result is 7Eh**	
6	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = EFh Text String TLV = 0D 81 F0 04 01 02 … 7E 7F … EF		
	getValueByte(7E)	**Result is 7Eh**	
7	getValueByte(7F)	**Result is 7Fh**	
8	getValueByte(EF)	**Result is EFh**	

6.1.8.13.5 Test Coverage

CRR number	Test case number
N1	3, 4, 5, 6, 7, 8
P1	2
C1	Does not apply for Proactive Response Handler
C2	1

6.1.8.14 Method copyValue

6.1.8.14.1 Test Area Reference API_2_PRH_CPYVS_BSS

6.1.8.14.2 Conformance Requirement

The method with following prototype shall be compliant with its definition in the API.

```
public short copyValue(short valueOffset,
                       byte[] dstBuffer,
                       short dstOffset,
                       short dstLength)
                throws java.lang.NullPointerException,
                       java.lang.ArrayIndexOutOfBoundsException,
                       ToolkitException
```

Normal Execution

CRRN1: copies a part of the last TLV element which has been found, into a destination. buffer.

CRRN2: returns dstOffset + dstLength.

Parameter Error

CRRP1: if dstBuffer is null NullPointerException is thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative ArrayIndexOutOfBoundsException is thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.8.14.3 Test Suite files

- Test Script: API_2_PRH_CPYVS_BSS_1.scr
- Test Applet: API_2_PRH_CPYVS_BSS_1.java
- Installation parameter: API_2_PRH_CPYVS_BSS.install
- Load Script: API_2_PRH_CPYVS_BSS.ldr
- Conversion parameter: API_2_PRH_CPYVS_BSS.cnv

6.1.8.14.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 5 **Text String TLV = 0D 06 04 01 02 … 05**		
	Get the response Select Text String TLV		
	copyValue() with a null dstBuffer	`NullPointerException` is thrown	
2	dstOffset ≥ dstBuffer.length **dstBuffer.length = 5** **dstOffset = 5** **dstLength = 1**	`ArrayIndexOutOfBoundsException` is thrown	
3	dstOffset < 0 **dstBuffer.length = 5** **dstOffset = -1**	`ArrayIndexOutOfBoundsException` is thrown	

	dstLength = 1		
4	dstLength >dstBuffer.length **dstBuffer.length = 5** **dstOffset = 0** **dstLength = 6**	ArrayIndexOutOfBoundsException is thrown	
5	dstOffset + dstLength >dstBuffer.length **dstBuffer.length = 5** **dstOffset = 3** **dstLength = 3**	ArrayIndexOutOfBoundsException is thrown	
6	dstLength < 0 **dstBuffer.length = 5** **dstOffset = 0** **dstLength = -1**	ArrayIndexOutOfBoundsException is thrown	
7	valueOffset ≥ Text String Length **valueOffset = 6** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	valueOffset < 0 **valueOffset = -1** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
9	dstLength > Text String length **valueOffset = 0** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 7**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
10	valueOffset + dstLength > Text String length **valueOffset = 2** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 5**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
11	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 16 **Text String TLV = 0D 11 04 00 01 … 0F**		
	Get the response		
	copyValue()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
12	Select Text String TLV		

	Successful call **valueOffset = 0** **dstBuffer.length = 17** **dstOffset = 0** **dstLength = 17**	**Result of copyValue() is 17**	
13	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
14	initialise dstBuffer **dstBuffer = 55 55 … 55**		
	Successful call **valueOffset = 2** **dstBuffer.length = 20** **dstOffset = 3** **dstLength = 12**	**Result of copyValue() is 15**	
15	Compare buffer **buffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**	Result is 00h	

6.1.8.14.5 Test Coverage

CRR number	Test case number
N1	13, 15
N2	12, 14
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for Proactive Response Handler
C2	11

6.1.8.15 Method compareValue

6.1.8.15.1 Test Area Reference: API_2_PRH_CPRVS_BSS

6.1.8.15.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte compareValue(short valueOffset,
                         byte[] compareBuffer,
                         short compareOffset,
                         short compareLength)
                  throws java.lang.NullPointerException,
```

`java.lang.ArrayIndexOutOfBoundsException,`

`ToolkitException`

Normal Execution

Compares the last found TLV element with a buffer :

CRRN1: returns 0 if identical.

CRRN2: returns -1 if the first miscomparing byte in simple TLV List is less than that in compareBuffer.

CRRN3: returns 1 if the first miscomparing byte in simple TLV List is greater than that in compareBuffer.

Parameter Error

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset or compareLength or both would cause access outside array bounds, or if compareLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.8.15.3 Test Suite files

- Test Script: API_2_PRH_CPRVS_BSS_1.scr
- Test Applet: API_2_PRH_CPRVS_BSS_1.java
- Installation parameter: API_2_PRH_CPRVS_BSS.install
- Load Script: API_2_PRH_CPRVS_BSS.ldr
- Conversion parameter: API_2_PRH_CPRVS_BSS.cnv

6.1.8.15.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 5 **`Text String TLV = 0D 06 04 01 02 … 05`**		
	Get the response Select Text String TLV		
	`compareValue()` with a null compareBuffer	`NullPointerException` is thrown	
2	compareOffset ≥ compareBuffer.length **`compareBuffer.length = 5`** **`compareOffset = 5`** **`compareLength = 1`**	**`ArrayIndexOutOfBoundsException` is thrown**	

3	compareOffset < 0 **compareBuffer.length = 5** **compareOffset = -1** **compareLength = 1**	`ArrayIndexOutOfBoundsException` is thrown	
4	compareLength >compareBuffer.length **compareBuffer.length = 5** **compareOffset = 0** **compareLength = 6**	`ArrayIndexOutOfBoundsException` is thrown	
5	compareOffset + compareLength >compareBuffer.length **compareBuffer.length = 5** **compareOffset = 3** **compareLength = 3**	`ArrayIndexOutOfBoundsException` is thrown	
6	compareLength < 0 **compareBuffer.length = 5** **compareOffset = 0** **compareLength = -1**	`ArrayIndexOutOfBoundsException` is thrown	
7	valueOffset ≥ Text String Length **valueOffset = 6** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown	
8	valueOffset < 0 **valueOffset = -1** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	
9	compareLength > Text String length **valueOffset = 0** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 7**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	
10	valueOffset + compareLength > Text String length **valueOffset = 2** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 5**	**`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown**	
11	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 16 **Text String TLV = 0D 11 04 00 01 … 0F**		

	Get the response		
	compareValue()	`ToolkitException.UNAVAILABLE_ELEMENT` is thrown	
12	Select Text String TLV		
	Initialise compareBuffer **`compareBuffer =`** **`04 00 01 … 0F`**		
	Compare buffers **`valueOffset = 0`** **`compareOffset = 0`** **`compareLength = 17`**	**Result is 00h**	
13	Initialise compareBuffer **`compareBuffer =`** **`04 00 01 …`** `10`		
	Compare buffers with same parameters	**Result is -1**	
14	Initialise compareBuffer **`compareBuffer =`** `03` **`00 01 … 0F`**		
	Compare buffers with same parameters	**Result is +1**	
15	Initialise compareBuffer **`compareBuffer =`** **`55 55 55 01 02`** **`03 04 05 06 07`** **`08 09 0A 0B 0C`** **`55 55 55 55 55`**		
	Compare buffers **`valueOffset = 2`** **`compareOffset = 3`** **`compareLength = 12`**	**Result is 00h**	
16	Initialise compareBuffer **`compareBuffer =`** **`55 55 55`** `02 01` **`03 04 05 06 07`** **`08 09 0A 0B 0C`** **`55 55 55 55 55`**		
	Compare buffers with same parameters	**Result is -1**	
17	Initialise compareBuffer **`compareBuffer =`**		

	55 55 55 01 02 **03 04 05 06 07** **08 09 0A** 0A 0D **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is +1**	

6.1.8.15.5 Test Coverage

CRR number	Test case number
N1	12, 15
N2	13, 16
N3	14, 17
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for Proactive Response Handler
C2	11

6.1.8.16 Method findAndCopyValue

6.1.8.16.1 Test Area Reference: API_2_PRH_FACYB_BS

6.1.8.16.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short findAndCopyValue(byte tag,
                              byte[] dstBuffer,
                              short dstOffset)
                throws    java.lang.NullPointerException,
                          java.lang.ArrayIndexOutOfBoundsException,
                          ToolkitException
```

Normal Execution

CRRN1: looks for the first occurrence of a TLV element from the beginning of a TLV list and copy its value into a destination buffer.

CRRN2: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN3: if the method is successful then the corresponding TLV becomes current and dstOffset + length of the copied value is returned.

CRRN4: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if dstBuffer is null NullPointerException shall be thrown.

CRRP2: if dstOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.8.16.3 Test Suite files

- Test Script: API_2_PRH_FACYB_BS_1.scr
- Test Applet: API_2_PRH_FACYB_BS_1.java
- Installation parameter: API_2_PRH_FACYB_BS.install
- Load Script: API_2_PRH_FACYB_BS.ldr
- Conversion parameter: API_2_PRH_FACYB_BS.cnv

6.1.8.16.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 15 **Text String TLV = 0D 10 04 01 02 … 0F**		
	Get the response		
	findAndCopyValue() with a null `dstBuffer`	`NullPointerException` is thrown	
2	dstOffset ≥ dstBuffer.length **tag = 0Dh** **dstBuffer.length = 20** **dstOffset = 20**	`ArrayIndexOutOfBoundsException` is thrown	
3	dstOffset < 0 **dstBuffer.length = 20** **dstOffset = -1**	`ArrayIndexOutOfBoundsException` is thrown	
4	length > dstBuffer.length **dstBuffer.length = 15** **dstOffset = 0**	`ArrayIndexOutOfBoundsException` is thrown	
5	dstOffset + length >dstBuffer.length **dstBuffer.length = 20** **dstOffset = 5**	`ArrayIndexOutOfBoundsException` is thrown	
6	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 16 **Text String TLV = 0D 10 04 00 01 … 0F**		
	Get the response		
	Select a TLV (tag 02h)		

	findAndCopyValue() **tag = 04h**	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the getValueLength() method	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
7	Successful call **tag = 0Dh** **dstBuffer.length = 17** **dstOffset = 0**	Result of copyValue() is 17	
8	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
9	initialise dstBuffer **dstBuffer = 55 55 … 55**		
	Successful call **dstBuffer.length = 20** **dstOffset = 2**	**Result of copyValue() is 19**	
10	Compare buffer **buffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**	Result is 00h	
11	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, with 2 Text String TLV **0D 10 04 00 01 … 0F** **0D 02 04 41**		
	Get the response		
	Successful call **tag = 0Dh** **dstBuffer.length = 17** **dstOffset = 0**	**Result of copyValue() is 17**	
12	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
13	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 16 **Text String TLV = 0D 10 04 00 01 … 0F**		
	Get the response		
	Successful call (with tag 8Dh) **tag = 8Dh** **dstBuffer.length = 17**	Result of copyValue() is 17	

	dstOffset = 0		
14	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	

6.1.8.16.5 Test Coverage

CRR number	Test case number
N1	8, 10, 12
N2	6
N3	7, 9, 11
N4	13, 14
P1	1
P2	2, 3, 4, 5
C1	Does not apply for Proactive Response Handler

6.1.8.17 Method findAndCopyValue

6.1.8.17.1 Test Area Reference: API_2_PRH_FACYBS_BSS

6.1.8.17.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short findAndCopyValue(byte tag,
                              byte occurence,
                             short valueOffset,
                             byte[] dstBuffer,
                             short dstOffset,
                             short dstLength)
                      throws java.lang.NullPointerException,
                             java.lang.ArrayIndexOutOfBoundsException,
                             ToolkitException
```

Normal Execution

CRRN1: looks for the indicated occurrence of a TLV element from the beginning of a TLV list and copy its value into a destination buffer.

CRRN2: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN3: if the method is successful then the corresponding TLV becomes current and dstOffset + dstLength is returned.

CRRN4: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if dstBuffer is null NullPointerException shall be thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.8.17.3 Test Suite files

- Test Script: API_2_PRH_FACYBS_BSS_1.scr
- Test Applet: API_2_PRH_FACYBS_BSS_1.java
- Installation parameter: API_2_PRH_FACYBS_BSS.install
- Load Script: API_2_PRH_FACYBS_BSS.ldr
- Conversion parameter: API_2_PRH_FACYBS_BSS.cnv

6.1.8.17.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 15 **`Text String TLV = 0D 10 04 01 02 … 0F`**		
	Get the response		
	findAndCopyValue() with a null `dstBuffer`	`NullPointerException` is thrown	
2	dstOffset ≥ dstBuffer.length **`tag = 0Dh, occurrence = 1`** **`valueOffset = 0`** **`dstBuffer.length = 5`** **`dstOffset = 5`** **`dstLength = 1`**	`ArrayIndexOutOfBoundsException` is thrown	
3	dstOffset < 0 **`dstBuffer.length = 5`** **`dstOffset = -1`** **`dstLength = 1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	dstLength > dstBuffer.length **`dstBuffer.length = 5`** **`dstOffset = 0`** **`dstLength = 6`**	`ArrayIndexOutOfBoundsException` is thrown	
5	dstOffset + dstLength > dstBuffer.length **`dstBuffer.length = 5`** **`dstOffset = 3`** **`dstLength = 3`**	`ArrayIndexOutOfBoundsException` is thrown	
6	dstLength < 0 **`dstBuffer.length = 5`**	`ArrayIndexOutOfBoundsException` is thrown	

	dstOffset = 0 **dstLength = -1**		
7	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 5 **Text String TLV = 0D 06 04 01 02 … 05**		
	Get the response		
	valueOffset ≥ Text String Length **tag = 0Dh, occurrence = 1** **valueOffset = 6** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
8	valueOffset < 0 **valueOffset = -1** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
9	dstLength > Text String length **valueOffset = 0** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 7**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
10	valueOffset + dstLength > Text String length **valueOffset = 2** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 5**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
11	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 16 **Text String TLV = 0D 10 04 00 01 … 0F**		
	Get the response		
	Select a TLV (tag 02h)		
	findAndCopyValue() **tag = 0Dh** **occurrence =** 2	`ToolkitException.UNAVAILABLE_ELEMENT` is thrown	
	Call the `getValueLength()` method	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
12	Successful call	Result of `findAndCopyValue()` is 17	

	tag = 0Dh, occurrence = 1 **valueOffset = 0** **dstBuffer.length = 17** **dstOffset = 0** **dstLength = 17**	`findAndCopyValue()` is 17	
13	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
14	initialise dstBuffer **dstBuffer = 55 55 … 55**		
	Successful call **tag = 0Dh, occurrence = 1** **valueOffset = 2** **dstBuffer.length = 20** **dstOffset = 3** **dstLength = 12**	**Result of `copyValue()` is 15**	
15	Compare buffer **buffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**	Result is 00h	
16	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, with 2 Text String TLV **0D 10 04 00 01 02 … 0F** **0D 00 11 22 33 44 55 (no specific DCS byte)**		
	Get the response		
	Successful call **tag = 0Dh, occurrence = 1** **valueOffset = 0** **dstBuffer.length = 17** **dstOffset = 0** **dstLength = 17**	**Result of `findAndCopyValue()` is 17**	
17	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
18	Successful call **tag = 0Dh, occurrence = 2** **valueOffset = 0** **dstBuffer.length = 6** **dstOffset = 0** **dstLength = 6**	**Result of `findAndCopyValue()` is 6**	

19	Compare buffer **buffer = 00 11 22 33 44 55**	Result is 00h	
20	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 16 **Text String TLV = 0D 10 04 00 01 … 0F**		
	Get the response		
	Successful call (with tag 8Dh) **tag = 8Dh** **dstBuffer.length = 17** **dstOffset = 0**	Result of `copyValue()` is 17	
21	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	

6.1.8.17.5 Test Coverage

CRR number	Test case number
N1	13, 15, 17, 19
N2	11
N3	12, 14, 16, 18
N4	20, 21
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for Proactive Response Handler

6.1.8.18 Method findAndCompareValue

6.1.8.18.1 Test Area Reference: API_2_PRH_FACRB_BS

6.1.8.18.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte findAndCompareValue(byte tag,
                                byte[] compareBuffer,
                                short compareOffset)
                throws     java.lang.NullPointerException,
                           java.lang.ArrayIndexOutOfBoundsException,
                     ToolkitException
```

Normal Execution

Looks for the first occurrence of a TLV element from beginning of a TLV list and compare its value with a buffer :

CRRN1: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN2: if the method is successful then the corresponding TLV becomes current.

CRRN3: if identical returns 0.

CRRN4: if the first miscomparing byte in simple TLV is less than that in compareBuffer returns -1.

CRRN5: if the first miscomparing byte in simple TLV is greater than that in compareBuffer returns 1.

CRRN6: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.8.18.3 Test Suite files

- Test Script: API_2_PRH_FACRB_BS_1.scr
- Test Applet: API_2_PRH_FACRB_BS_1.java
- Installation parameter: API_2_PRH_FACRB_BS.install
- Load Script: API_2_PRH_FACRB_BS.ldr
- Conversion parameter: API_2_PRH_FACRB_BS.cnv

6.1.8.18.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 15 **`Text String TLV = 0D 10 04 01 02 … 0F`**		
	Get the response		
	findAndCompareValue() with a null `dstBuffer`	`NullPointerException` is thrown	
2	compareOffset ≥ compareBuffer.length **`tag = 0Dh`** **`compareBuffer.length = 20`** **`compareOffset = 20`**	`ArrayIndexOutOfBoundsException` is thrown	
3	compareOffset < 0 **`compareBuffer.length = 20`** **`compareOffset = -1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	length > compareBuffer.length	`ArrayIndexOutOfBoundsException` is thrown	

	compareBuffer.length = 15 **compareOffset = 0**		
5	compareOffset + length > compareBuffer.length **compareBuffer.length = 20** **compareOffset = 5**	`ArrayIndexOutOfBoundsException` is thrown	
6	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 16 **Text String TLV = 0D 10 04 00 01 … 0F**		
	Get the response		
	Select a TLV (tag 02h)		
	findAndCompareValue() **tag = 04h**	`ToolkitException.UNAVAILABLE_ELEMENT` is thrown	
	Call the `getValueLength()` method	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
7	Initialise compareBuffer **compareBuffer =** **04 00 01 … 0F**		
	Compare buffers **tag = 0Dh** **compareOffset = 0**	**Result is 00h**	
8	Verify current TLV **getValueLength()**	**Result is 17**	
9	Initialise compareBuffer **compareBuffer =** **04 00 01 …** 10		
	Compare buffers with same parameters	**Result is -1**	
10	Initialise compareBuffer **compareBuffer =** 03 **00 01 … 0F**		
	Compare buffers with same parameters	**Result is +1**	
11	Initialise compareBuffer **compareBuffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**		
	Compare buffers **compareOffset = 2**	**Result is 00h**	

12	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, with 2 Text String TLV **0D 10 04 00 01 … 0F** **0D 06 00 11 22 33 44 55**		
	Get the response		
	Initialise compareBuffer **compareBuffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**		
	Compare buffers **compareOffset = 2**	**Result is 00h**	
13	Initialise compareBuffer **compareBuffer =** **55 55 04** 01 **01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**		
	Compare buffers **compareOffset = 2**	**Result is -1**	
14	Initialise compareBuffer **compareBuffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D** 0D 10 **55**		
	Compare buffers **compareOffset = 2**	**Result is +1**	
15	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 16 **Text String TLV = 0D 10 04 00 01 … 0F**		
	Get the response		
	Initialise compareBuffer **CompareBuffer =** **04 00 01 … 0F**		
	Compare buffers (with tag 8Dh)	**Result is 00h**	

	tag = 8Dh **compareOffset = 0**		

6.1.8.18.5 Test Coverage

CRR number	Test case number
N1	6
N2	8
N3	7, 11, 12
N4	9, 13
N5	10, 14
N6	15
P1	1
P2	2, 3, 4, 5
C1	Does not apply for Proactive Response Handler

6.1.8.19 Method findAndCompareValue

6.1.8.19.1 Test Area Reference: API_2_PRH_FACRBBS_BSS

6.1.8.19.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte findAndCompareValue(byte tag,
                                byte occurence,
                                short valueOffset,
                                byte[] compareBuffer,
                                short compareOffset,
                                short compareLength)
                         throws java.lang.NullPointerException,
                                java.lang.ArrayIndexOutOfBoundsException,
                                ToolkitException
```

Normal Execution

Looks for the indicated occurrence of a TLV element from the beginning of a TLV list and compare its value with a buffer:

CRRN1: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN2: if the method is successful then the corresponding TLV becomes current.

CRRN3: if identical 0 is returned.

CRRN4: if the first miscomparing byte in simple TLV is less than that in compareBuffer -1 is returned.

CRRN5: if the first miscomparing byte in simple TLV is greater than that in compareBuffer 1 is returned

CRRN6: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset or compareLength or both would cause access outside array bounds, or if compareLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, compareLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

CRRP4: if an input parameter is not valid (e.g. occurence = 0) an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException BAD_INPUT_PARAMETER.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.8.19.3 Test Suite files

- Test Script: API_2_PRH_FACRBBS_BSS_1.scr
- Test Applet: API_2_PRH_FACRBBS_BSS_1.java
- Installation parameter: API_2_PRH_FACRBBS_BSS.install
- Load Script: API_2_PRH_FACRBBS_BSS.ldr
- Conversion parameter: API_2_PRH_FACRBBS_BSS.cnv

6.1.8.19.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 15 **`Text String TLV = 0D 10 04 01 02 … 0F`**		
	Get the response		
	`findAndCompareValue()` with a null compareBuffer	`NullPointerException` is thrown	
2	compareOffset ≥ compareBuffer.length **`tag = 0Dh, occurrence = 1`** **`valueOffset = 0`** **`compareBuffer.length = 5`** **`compareOffset = 5`** **`compareLength = 1`**	`ArrayIndexOutOfBoundsException` is thrown	
3	compareOffset < 0 **`compareBuffer.length = 5`** **`compareOffset = -1`** **`compareLength = 1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	compareLength >compareBuffer.length **`compareBuffer.length = 5`** **`compareOffset = 0`**	`ArrayIndexOutOfBoundsException` is thrown	

	compareLength = 6		
5	compareOffset + compareLength >compareBuffer.length **compareBuffer.length = 5** **compareOffset = 3** **compareLength = 3**	`ArrayIndexOutOfBoundsException` is thrown	
6	compareLength < 0 **compareBuffer.length = 5** **compareOffset = 0** **compareLength = -1**	`ArrayIndexOutOfBoundsException` is thrown	
7	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 16 **Text String TLV = 0D 10 04 00 01 … 0F**		
	Get the response		
	valueOffset ≥ Text String Length **tag = 0Dh, occurrence = 1** **valueOffset = 6** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
8	valueOffset < 0 **valueOffset = -1** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
9	compareLength > Text String length **valueOffset = 0** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 7**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
10	valueOffset + compareLength > Text String length **valueOffset = 2** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 5**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
11	Invalid parameter **occurrence = 0**	**ToolkitException.BAD_INPUT_PARAMETER is thrown**	
12	Send a GET INPUT command		**GET INPUT Pro command**

	Terminal Response, Text String length = 16 **Text String TLV = 0D 10 04 00 01 … 0F**		
	Get the response		
	Select a TLV (tag 02h)		
	findAndCompareValue() **tag = 0Dh** **occurrence =** 2	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the getValueLength() method	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
13	Initialise compareBuffer **compareBuffer =** **04 00 01 … 0F**		
	findAndCompareValue() **tag = 0Dh, occurrence = 1** **valueOffset = 0** **compareOffset = 0** **compareLength = 17**	**Result is 00h**	
14	Verify current TLV **getValueLength()**	**Result is 17**	
15	Initialise compareBuffer **compareBuffer =** **04 00 01 …** 10		
	Compare buffers with same parameters	**Result is -1**	
16	Initialise compareBuffer **compareBuffer =** 03 **00 01 … 0F**		
	Compare buffers with same parameters	**Result is +1**	
17	Initialise compareBuffer **compareBuffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**		
	Compare buffers **valueOffset = 2** **compareOffset = 3** **compareLength = 12**	**Result is 00h**	
18	Initialise compareBuffer		

	compareBuffer = **55 55 55** 02 01 **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is -1**	
19	Initialise compareBuffer **compareBuffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A** 0A 0D **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is +1**	
20	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, with 2 Text String TLV **0D 10 04 00 01 … 0F** **0D 06 00 11 22 33 44 55**		
	Get the response		
	Initialise compareBuffer **compareBuffer =** **04 00 01 … 0F**		
	findAndCompareValue() **tag = 0Dh, occurrence = 1** **valueOffset = 0** **compareOffset = 0** **compareLength = 17**	**Result is 00h**	
21	Initialise compareBuffer **compareBuffer =** **00 11 22 33 44 55**		
	findAndCompareValue() **tag = 0Dh, occurrence = 2** **valueOffset = 0** **compareOffset = 0** **compareLength = 6**	**Result is 00h**	
22	Initialise compareBuffer **compareBuffer =** **00 11 22 33 44** 66		
	findAndCompareValue() **tag = 0Dh, occurrence = 2**	**Result is -1**	

	valueOffset = 0 **compareOffset = 0** **compareLength = 6**		
23	Send a GET INPUT command		**GET INPUT Pro command**
	Terminal Response, Text String length = 16 **Text String TLV = 0D 10 04 00 01 … 0F**		
	Get the response		
	Initialise compareBuffer **CompareBuffer =** **04 00 01 … 0F**		
	Compare buffers (with tag 8Dh) **tag = 8Dh** **compareOffset = 0**	**Result is 00h**	

6.1.8.19.5 Test Coverage

CRR number	Test case number
N1	12
N2	14
N3	13, 17, 20, 21
N4	15, 18, 22
N5	16, 19
N6	23
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
P4	11
C1	Does not apply for Proactive Response Handler

6.1.9 Class EnvelopeResponseHandler

6.1.9.1 Method getTheHandler

6.1.9.1.1 Test Area Reference: API_2_ERH_GTHD

6.1.9.1.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public static EnvelopeResponseHandler getTheHandler()
              throws ToolkitException
```

Normal Execution

CRRN1: The method shall return the single system instance of the EnvelopeResponseHandler class.

CRRN2: The EnvelopeResponseHandler is a Temporary JCRE Entry Point Object (see ref X

Parameters error

No requirements

Context error

CRRC1: The method shall thrown ToolkitException (HANDLER_NOT_AVAILABLE) if the handler is busy.

CRRC2: After the first invocation of the ProactiveHandler.send method the EnvelopeResponseHandler is no more available

6.1.9.1.3 Test suite files:

- Test Script: API_2_ERH_GTHD_1.scr
- Test Applet: API_2_ERH_GTHD_1.java
- Installation parameter: API_2_ERH_GTHD.install (same as default applet)
- Load Script: API_2_ERH_GTHD.ldr
- Conversion parameter: API_2_ERH_GTHD.cnv

6.1.9.1.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	getTheHandler twice	The returned objects shall be the same	
2	GetTheHandler	The reference returned shall be an EnvelopeResponseHandler (checkcast)	
3	GetTheHandler	The reference returned shall not be null.	
4	getTheHandler and store it in a static field of the toolkit applet	SecurityException is thrown	
5	getTheHandler and store it in a field of the toolkit applet	SecurityException is thrown	
6	getTheHandler, then send a proactive command, and then, appendTLV	ToolkitException HANDLER_NOT_AVAILABLE is thrown	

6.1.9.1.5 Test Coverage

CRR number	*Test case number*
N1	***1, 2, 3***

N2	***4, 5***
C1	***To be checked in Framework tests and insert here cross reference***
C2	***6***

6.1.9.2 Method post

6.1.9.2.1 Test Area Reference: API_2_ERH_POST_B

6.1.9.2.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public void post(byte statusType)
         throws ToolkitException
```

Normal Execution

CRRN1: When the method is called, the toolkit applet can continue it's processing (e.g. prepare a proactive command).

CRRN2: The byte statusType is SW1 of the status.

CRRN3: If the send method is called after a post method, the posted data are the first sent to the ME.

CRRN4: The SIM Toolkit Framework shall take the optional Application Data posted by the triggered toolkit applet if present, secure and send the response packet the SIM Toolkit Framework will return the response APDU defined by the toolkit applet.

Parameters error

No requirements

Context error

CRRC1: The method shall thrown ToolkitException (HANDLER_NOT_AVAILABLE) if the handler is busy.

6.1.9.2.3 Test suite files:

- Test Script: API_2_ERH_POST_B_1.scr

  All SMS PP sent to the applet are not secured,

- Test Applet: API_2_ERH_POST_B_!.java
- Installation parameter: API_2_ERH_POST_B.install (Same as default applet)
- Load Script: API_2_ERH_POST_B.ldr
- Conversion parameter: API_2_ERH_POST_B.cnv

6.1.9.2.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**getTheHandler and then post**		**9000**
2	**Fill the handler (appendTLV to have 256 bytes in it)and then post data with status 9F**		**9F00 data are retrieved with GET RESPONSE command**
3	**appendTLV, post and then appendTLV**	**? : question sent to T3**	
4	**construct the response and post it with status 9E and then send a display text**		**9EXX and posted data retreived by a GET RESPONSE with status 91XX and display text retreived by a FETCH**

5	**getTheHandler and post , then send a display text**		**91XX and display text is retreived by a FETCH**
6	**getTheHandler, appendTLV, send a display text, post.**	**ToolkitException UNAVAILABLE_ELEMENT is thrown**	
7	**getTheHandler, appendTLV, post with status 9EXX, post with status 9FXX**		**? linked to test case 3, question sent to T3, can we modify a posted handler?**

6.1.9.2.5 Test Coverage

CRR number	***Test case number***
N1	***3, 4, 7***
N2	***1, 2, 4, 7***
N3	***4, 5***
N4	***To be checked in Framework tests and insert here cross reference***
C1	***6***

6.1.9.3 Method postAsBERTLV

6.1.9.3.1 Test Area Reference: API_2_ERH_POST_BB

6.1.9.3.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public void postAsBERTLV(byte statusType,
                         byte tag)
                  throws ToolkitException
```

Normal Execution

CRRN1: When the method is called, the toolkit applet can continue it's processing (e.g. prepare a proactive command) the SIM Toolkit Framework will return the response APDU defined by the toolkit applet.

CRRN2: The byte statusType is SW1 of the status

CRRN3: If the send method is called after a postAsBERTLV method, the posted data are the first sent to the ME.

CRRN4: The byte tag is the BER Tag at the beginning of the simple TLV list.

Parameters error

No requirements

Context error

CRRC1: The method shall thrown ToolkitException (HANDLER_NOT_AVAILABLE) if the handler is busy.

6.1.9.3.3 Test suite files:

Specific triggering :

Call control

- Test Script: API_2_ERH_POST_BB.scr
- Test Applet: API_2_ERH_POST_BB.java
- Installation parameter: API_2_ERH_POST_BB.install (Same as default applet)

- Load Script: API_2_ERH_POST_BB.ldr
- Conversion parameter: API_2_ERH_POST_BB.cnv

6.1.9.3.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	getTheHandler and then postAsBERTLV		9000
2	Fill the handler and then postAsBERTLV the data with status 9F, and tag 33		9F00 data are retreived with GET RESPONSE command, the tag shall be 33
3	appendTLV, postAsBERTLV and then appendTLV	? : question sent to T3	
4	construct the response and postAsBERTLV it with status 9E, tag 75 and then send a display text		9EXX and posted data retreived by a GET RESPONSE the tag shall be 75 with status 91XX and display text retreived by a FETCH
5	getTheHandler and postAsBERTLV, then send a display text		91XX and display text is retreived by a FETCH
6	GetTheHandler, appendTLV, send a display text, postAsBERTLV.	ToolkitException UNAVAILABLE_ELEMENT is thrown	
7	GetTheHandler, appendTLV, postAsBERTLV with status 9EXX, tag 56, postAsBERTLV with status 9FXX, tag 28		? linked to test case 3, question sent to T3, can we modify the posted handler ?

6.1.9.3.5 Test Coverage

CRR number	*Test case number*
N1	*3, 4, 7*
N2	*1, 2, 4, 7*
N3	*4, 5*
N4	*2, 4, 7*
C1	*6*

6.1.9.4 Method appendArray

6.1.9.4.1 Test Area Reference: API_2_EDH_APDA

6.1.9.4.2 Conformance Requirement:

The method with the following method header shall comply to its definition in the API.

```
void appendArray(byte[] buffer, short offset, short length)
                 throws java.lang.NullPointerException,
                        java.lang.ArrayIndexOutOfBoundsException,
                        ToolkitException
```

Normal Execution

CRRN1: appends a buffer into the Edithandler buffer

CRRN2: a successful append does not modify the TLV selected

Parameters error

CRRP1: if buffer is null, a java.lang.NullPointerException is thrown

CRRP2: if offset or length or both would cause access outside the array bounds, or if length is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.

Context error

CRRC1: if the EditHandler buffer is too small to append the requested data, a ToolkitException is thrown with reason code HANDLER_OVERFLOW

CRRC2: if the EditHandler buffer is busy, a ToolkitException is thrown with reason code HANDLER_NOT_AVAILABLE

6.1.9.4.3 Test suite files:

- Test Script : API_2_EDH_APDA_1.scr
- Test Applet: API_2_EDH_APDA_1.java
- Installation parameter: API_2_EDH_APDA.install (Same as default applet).
- Load Script: API_2_EDH_APDA.ldr
- Conversion parameter: API_2_EDH_APDA.cnv

6.1.9.4.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			

6.1.9.4.5 Test Coverage

CRR number	*Test case number*

6.1.9.5 Method appendTLV

6.1.9.5.1 Test Area Reference: API_2_EDH_APTLVBB

6.1.9.5.2 Conformance Requirement:

The method with the following method header shall comply to its definition in the API.

```
void appendTLV(byte tag, byte value)
               throws ToolkitException
```

Normal Execution

CRRN1 : Appends a TLV element to the current TLV list (1-byte element).

CRRN2 : A successful append does not modify the TLV selected.

Parameters error

None

Context error

CRRC1: if the EditHandler buffer is too small to append the requested data, a ToolkitException is thrown with reason code HANDLER_OVERFLOW

CRRC2: if the EditHandler buffer is busy, a ToolkitException is thrown with reason code HANDLER_NOT_AVAILABLE

6.1.9.5.3 Test suite files:

- Test Script: API_2_EDH_APTLVBB_1.scr
- Test Applet: API_2_EDH_APTLVBB_1.java
- Installation parameter: API_2_EDH_APTLVBB.install (Same as default applet).
- Load Script: API_2_EDH_APTLVBB.ldr
- Conversion parameter: API_2_EDH_APTLVBB.cnv

6.1.9.5.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			

6.1.9.5.5 Test Coverage

CRR number	*Test case number*

6.1.9.6 Method appendTLV

6.1.9.6.1 Test Area Reference: API_2_EDH_APTLVBBB

6.1.9.6.2 Conformance Requirements:

The method with the following method header shall comply to its definition in the API.

```
void appendTLV (byte tag, byte value1, byte value2)
               throws ToolkitException
```

Normal Execution

CRRN1: Appends a TLV element to the current TLV list (2-byte element).

CRRN2: A successful append does not modify the TLV selected.

Parameters error

None

Context error

CRRC1: if the EditHandler buffer is too small to append the requested data, a ToolkitException is thrown with reason code HANDLER_OVERFLOW

CRRC2: if the EditHandler buffer is busy, a ToolkitException is thrown with reason code HANDLER_NOT_AVAILABLE

6.1.9.6.3 Test suite files:

- Test Script: API_2_EDH_APTLVBBB_1.scr
- Test Applet: API_2_EDH_APTLVBBB_1.java
- Installation parameter: API_2_EDH_APTLVBBB.install
- Load Script: API_2_EDH_APTLVBBB.ldr

- Conversion parameter: API_2_EDH_APTLVBBB.cnv

6.1.9.6.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			

6.1.9.6.5 Test Coverage

CRR number	*Test case number*

6.1.9.7 Method appendTLV

6.1.9.7.1 Test Area Reference: API_2_EDH_APTLVB_BSS

6.1.9.7.2 Conformance Requirement:

The method with the following method header shall comply to its definition in the API.

```
void appendTLV(byte tag,
               byte[] value,
               short valueoffset,
               short valuelength)
              throws java.lang.NullPointerException,

                     java.lang.ArrayIndexOutOfBoundsException,

                     ToolkitException
```

Normal Execution

CRRN1: Appends a TLV element to the current TLV list (byte-array element).

CRRN2: A successful append does not modify the TLV selected.

Parameters error

CRRP1: if value is null, a java.lang.NullPointerException is thrown

CRRP2: if valueoffset or valuelength or both would cause access outside the array bounds, or if length is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.

Context error

CRRC1: if the EditHandler buffer is too small to append the requested data, a ToolkitException is thrown with reason code HANDLER_OVERFLOW

CRRC2: if the EditHandler buffer is busy, a ToolkitException is thrown with reason code HANDLER_NOT_AVAILABLE

CRRC3: if valuelength is greater than 255, a ToolkitException is thrown with reason code BAD_INPUT_PARAMETER

6.1.9.7.3 Test suite files:

- Test Script: API_2_EDH_APTLVB_BSS_1.scr
- Test Applet: API_2_EDH_APTLVB_BSS_1.java
- Installation parameter: API_2_EDH_APTLVB_BSS.install (Same as default applet).
- Load Script: API_2_EDH_APTLVB_BSS.ldr

- Conversion parameter: API_2_EDH_APTLVB_BSS.cnv

6.1.9.7.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			

6.1.9.7.5 Test Coverage

CRR number	*Test case number*

6.1.9.8 Method appendTLV

6.1.9.8.1 Test Area Reference: API_2_EDH_APTLVBB_BSS

6.1.9.8.2 Conformance Requirement:

The method with the following method header shall comply to its definition in the API.

```
void appendTLV(byte tag,
               byte value1
               byte[] value2,
               short value2offset,
               short value2length)
              throws java.lang.NullPointerException,

                     java.lang.ArrayIndexOutOfBoundsException,

                     ToolkitException
```

Normal Execution

CRRN1 : Appends a TLV element to the current TLV list (1 byte and a byte-array element).

CRRN2 : A successful append does not modify the TLV selected.

Parameters error

CRRP1: if value2 is null, a java.lang.NullPointerException is thrown

CRRP2: if value2offset or value2length or both would cause access outside the array bounds, or if length is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.

Context error

CRRC1: if the EditHandler buffer is too small to append the requested data, a ToolkitException is thrown with reason code HANDLER_OVERFLOW

CRRC2: if the EditHandler buffer is busy, a ToolkitException is thrown with reason code HANDLER_NOT_AVAILABLE

CRRC3: if valuelength is greater than 255, a ToolkitException is thrown with reason code BAD_INPUT_PARAMETER

6.1.9.8.3 Test suite files:

- Test Script: API_2_EDH_APTLVBB_BSS_1.scr
- Test Applet: API_2_EDH_APTLVBB_BSS_1.java
- Installation parameter: API_2_EDH_APTLVBB_BSS.install (Same as default applet)
- Load Script: API_2_EDH_APTLVBB_BSS.ldr

- Conversion parameter: API_2_EDH_APTLVBB_BSS.cnv

6.1.9.8.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			

6.1.9.8.5 Test Coverage

CRR number	*Test case number*

6.1.9.9 Method clear

6.1.9.9.1 Test Area Reference: API_2_EDH_CLR

6.1.9.9.2 Conformance Requirement:

The method with the following method header shall comply to its definition in the API.

```
void clear()
          throws ToolkitException
```

Normal Execution

CRRN1 : Clears the TLV list of an EditHandler and resets the current TLV selected.

Parameters error

No requirements

Context error

CRRC1 : if the EditHandler buffer is busy, a ToolkitException is thrown with reason code `HANDLER_NOT_AVAILABLE`

6.1.9.9.3 Test suite files:

- Test Script: API_2_EDH_CLR_1.scr
- Test Applet: API_2_EDH_CLR _1.java
- Installation parameter: API_2_EDH_CLR.install (Same as default applet)
- Load Script: API_2_EDH_CLR.ldr
- Conversion parameter: API_2_EDH_CLR.cnv

6.1.9.9.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			

6.1.9.9.5 Test Coverage

CRR number	*Test case number*

6.1.9.10 Method getLength

6.1.9.10.1 Test Area Reference API_2_PAH_GLEN

6.1.9.10.2 Conformance Requirement

The method with following header shall be compliant to its definition in the API.

```
public short getLength()
                    throws ToolkitException
```

Normal Execution

CRRN1: returns the length in bytes of the TLV list.

Parameter Error

No requirements

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.HANDLER_NOT_AVAILABLE.

6.1.9.10.3 Test Suite files

- Test Script: API_2_PAH_GLEN_1.scr
- Test Applet: API_2_PAH_GLEN_1.java
- Installation parameter: API_2_PAH_GLEN.install
- Load Script: API_2_PAH_GLEN.ldr
- Conversion parameter: API_2_PAH_GLEN.cnv

6.1.9.10.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Clear the handler** **`getLength()`**	Result of `getLength()` is 0	
2	**Call the `init()` method** `getLength()`	Result of `getLength()` is 9	
3	**`Call the initDisplayText() method, with buffer length = 240`** **`getLength()`**	Result of `getLength()` is 253	
4	**`Build a 7Fh Proactive Handler`** **`getLength()`**	Result of `getLength()` is 7Fh	
5	**`Build a 80h Proactive Handler`**	Result of `getLength()` is 80h	

	getLength()		

6.1.9.10.5 Test Coverage

CRR number	Test case number
N1	1, 2, 3
C1	Does not apply for Proactive Handler

6.1.9.11 Method copy

6.1.9.11.1 Test Area Reference API_2_PAH_COPY_BSS

6.1.9.11.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short copy(byte[] dstBuffer,
                  short dstOffset,
                  short dstLength)
                   throws java.lang.NullPointerException,
                        java.lang.ArrayIndexOutOfBoundsException,
                        ToolkitException
```

Normal Execution

CRRN1: copies the simple TLV list contained in the handler to the destination byte array.

CRRN2: returns dstOffset + dstLength.

Parameter Error

CRRP1: if dstBuffer is null a NullPointerException is thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative, an ArrayIndexOutOfBoundsException is thrown.

CRRP3: if dstLength is grater than the length of the simple TLV List, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException. OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.9.11.3 Test Suite files

- Test Script: API_2_PAH_COPY_BSS_1.scr
- Test Applet: API_2_PAH_COPY_BSS_1.java
- Installation parameter: API_2_PAH_COPY_BSS.install
- Load Script: API_2_PAH_COPY_BSS.ldr
- Conversion parameter: API_2_PAH_COPY_BSS.cnv

6.1.9.11.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**NULL as parameter to dstBuffer**	`NullPointerException` is thrown	
2	Call the `init()` method		
	dstOffset ≥ dstBuffer.length **dstBuffer.length = 5** dstOffset = 5 dstLength = 1	`ArrayIndexOutOfBoundsException` is thrown	
3	**dstOffset < 0** **dstBuffer.length = 5** dstOffset = -1 dstLength = 1	`ArrayIndexOutOfBoundsException` is thrown	
4	**dstLength > dstBuffer.length** **dstBuffer.length = 5** dstOffset = 0 dstLength = 6	`ArrayIndexOutOfBoundsException` is thrown	
5	**dstOffset + dstLength > dstBuffer.length** **dstBuffer.length = 5** dstOffset = 3 dstLength = 3	`ArrayIndexOutOfBoundsException` is thrown	
6	**dstLength < 0** **dstBuffer.length = 5** dstOffset = 0 dstLength = -1	`ArrayIndexOutOfBoundsException` is thrown	
7	**dstLength > length of the simple TLV list** **dstBuffer.length = 10** dstOffset = 0 dstLength = 10	`ToolkitException.OUT_OF_TLV_BOUNDARIES` is thrown	
8	**Successful call, dstBuffer is the whole buffer** **dstBuffer.length = 9** dstOffset = 0 dstLength = 9	Result of `copy()` is 9	
9	**Compare the buffer**	Result of `arrayCompare()` is 0	
10	**Successful call, dstBuffer is part of a buffer** **dstBuffer.length = 15** dstOffset = 3 dstLength = 9	Result of `copy()` is 12	

11	**Compare the whole buffer**	Result of `arrayCompare()` is 0	
12	**Successful call, dstBuffer is part of a buffer** **dstBuffer.length = 15** dstOffset = 3 dstLength = 6	Result of `copy()` is 9	
13	**Compare the whole buffer**	Result of `arrayCompare()` is 0	

6.1.9.11.5 Test Coverage

CRR number	Test case number
N1	**9, 11, 13**
N2	**8, 10, 12**
P1	**1**
P2	**2, 3, 4, 5, 6**
P3	**7**
C1	**Does not apply for Proactive Handler**

6.1.9.12 Method findTLV

6.1.9.12.1 Test Area Reference API_2_PAH_FINDBB

6.1.9.12.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte findTLV(byte tag, byte occurrence)
              throws ToolkitException
```

Normal Execution

Looks for the indicated occurrence of a TLV element from the beginning of the TLV list (handler buffer):

CRRN1: the method is successful if the required occurrence exists then the corresponding TLV becomes current.

CRRN2: if the method is successful then it returns TLV_FOUND_CR_SET when Comprehension Required flag is set.

CRRN3: if the method is successful then it returns TLV_FOUND_CR_NOT_SET when Comprehension Required flag is not set.

CRRN4: if the required occurrence of the TLV element does not exist, no TLV is selected and TLV_NOT_FOUND is returned.

CRRN5: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if an input parameter is not valid (e.g. occurrence = 0) an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException BAD_INPUT_PARAMETER. The current TLV is no longer defined.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.9.12.3 Test Suite files

- Test Script: API_2_PAH_FINDBB_1.scr
- Test Applet: API_2_PAH_FINDBB_1.java
- Installation parameter: API_2_PAH_FINDBB.install
- Load Script: API_2_PAH_FINDBB.ldr
- Conversion parameter: API_2_PAH_FINDBB.cnv

6.1.9.12.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialise the handler**		
	Invalid input parameter **`Occurrence = 0`**	ToolkitException.BAD_INPUT_PARAMETER is thrown	
2	**Call the `init()` method**		
	Search 1st TLV **`Tag = 01h`** `Occurrence = 1`	**Result is `TLV_FOUND_CR_SET`**	
3	**Call the `getValueLength()` method**	**Result is 03h**	
4	**Search 2nd TLV** **`Tag = 02h`** `Occurrence = 1`	**Result is `TLV_FOUND_CR_SET`**	
5	**Call the `getValueLength()` method**	**Result is 02h**	
6	**Search a wrong tag** **`Tag = 03h`** `Occurrence = 1`	**Result is `TLV_NOT_FOUND`**	
7	**Call the `getValueLength()` method**	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
8	**Search a tag with wrong occurrence** **`Tag = 01h`** `Occurrence = 2`	**Result is `TLV_NOT_FOUND`**	
9	**Call the `getValueLength()` method**	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
10	**Append a TLV with tag=02h**		
	Search the TLV **`Tag = 02h`** `Occurrence = 2`	**Result is `TLV_FOUND_CR_NOT_SET`**	
11	**Append a TLV with tag=04h**		
	Search the TLV **`Tag = 04h`** `Occurrence = 1`	**Result is `TLV_FOUND_CR_NOT_SET`**	
12	**Search tag 81h** **`Tag = 81h`** `Occurrence = 1`	**Result is `TLV_FOUND_CR_SET`**	

13	**Search tag 84h** **Tag = 84h** Occurrence = 1	**Result is** **TLV_FOUND_CR_NOT_SET**	

6.1.9.12.5 Test Coverage

CRR number	Test case number
N1	3, 5
N2	2, 4
N3	10, 11
N4	6, 7,8, 9
N5	12, 13
P1	1
C1	Does not apply for Proactive Handler

6.1.9.13 Method getValueLength

6.1.9.13.1 Test Area Reference API_2_PAH_GVLEN

6.1.9.13.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short getValueLength()
                        throws ToolkitException
```

Normal Execution

CRRN1: gets and returns the binary length of the value field for the last TLV element which has been found in the handler.

Parameter Error

No requirements

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.9.13.3 Test Suite files

- Test Script: API_2_PAH_GVLEN_1.scr
- Test Applet: API_2_PAH_GVLEN_1.java
- Installation parameter: API_2_PAH_GVLEN.install
- Load Script: API_2_PAH_GVLEN.ldr
- Conversion parameter: API_2_PAH_GVLEN.cnv

6.1.9.13.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	**Call the `init()` method**		
	getValueLength()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	**Call the `initDisplayText()` method** **`length = 0`**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `00h`**	
3	**Call the `initDisplayText()` method** **`length = 1 (+ dcs byte)`**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `02h`**	
4	**Call the `initDisplayText()` method** **`length = 7Eh (+ dcs byte)`**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `7Fh`**	
5	**Call the `initDisplayText()` method** **`length = 7Fh (+ dcs byte)`**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `80h`**	
6	**Call the `initDisplayText()` method** **`length = F0h` (maximum text length)**		
	Search TLV `0Dh` (Text String TLV)		
	getValueLength()	**Result is `F1h`**	

6.1.9.13.5 Test Coverage

CRR number	Test case number
N1	**2, 3, 4, 5, 6**
C1	**Does not apply for Proactive Handler**
C2	**1**

6.1.9.14 Method getValueByte

6.1.9.14.1 Test Area Reference API_2_PAH_GVBYTS

6.1.9.14.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte getValueByte(short valueOffset)
```

```
                    throws ToolkitException
```

Normal Execution

CRRN1: Gets a byte from the last TLV element which has been found in the handler and returns its value (1 byte).

Parameter Error

CRRP1: if valueOffset is out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.9.14.3 Test Suite files

- Test Script: API_2_PAH_GVBYTS_1.scr
- Test Applet: API_2_PAH_GVBYTS_1.java
- Installation parameter: API_2_PAH_GVBYTS.install
- Load Script: API_2_PAH_GVBYTS.ldr
- Conversion parameter: API_2_PAH_GVBYTS.cnv

6.1.9.14.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	**Call the `init()` method**
	`type = FFh`
	`qualifier = FEh`
	`destination = FDh`		

	getValueByte(0)	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	Search TLV `01h` (Command Details TLV)		
	getValueByte(3)	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
3	Search TLV `01h` (Command Details TLV)		
	getValueByte(2)	**Result is `FEh` (qualifier)**	
4	Search TLV `02h` (Device Identities TLV)		
	getValueByte(0)	**Result is `81h` (Source)**	
5	initDisplayText() **`buffer = 00 01 … 7D`** **`length = 7Eh`** Search TLV `0Dh` (Text String TLV)		
	getValueByte(7E)	**Result is `7Dh`**	

6	initDisplayText() **buffer = 00 01 … 7D 7E** **length = 7Fh** Search TLV 0Dh (Text String TLV)		
	getValueByte(7E)	**Result is 7Dh**	
7	getValueByte(7F)	**Result is 7Eh**	
8	initDisplayText() **buffer = 00 01 … EF** **length = F0h** Search TLV 0Dh (Text String TLV)		
	getValueByte(F0)	**Result is EFh**	

6.1.9.14.5 Test Coverage

CRR number	Test case number
N1	**3, 4, 5, 6, 7, 8**
P1	**2**
C1	**Does not apply for Proactive Handler**
C2	**1**

6.1.9.15 Method copyValue

6.1.9.15.1 Test Area Reference API_2_PAH_CPYVS_BSS

6.1.9.15.2 Conformance Requirement

The method with following prototype shall be compliant with its definition in the API.

```
public short copyValue(short valueOffset,
                       byte[] dstBuffer,
                       short dstOffset,
                       short dstLength)
                throws java.lang.NullPointerException,
                       java.lang.ArrayIndexOutOfBoundsException,
                       ToolkitException
```

Normal Execution

CRRN1: copies a part of the last TLV element which has been found, into a destination. buffer.

CRRN2: returns dstOffset + dstLength.

Parameter Error

CRRP1: if dstBuffer is null NullPointerException is thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative ArrayIndexOutOfBoundsException is thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.9.15.3 Test Suite files

- Test Script: API_2_PAH_CPYVS_BSS _1.scr
- Test Applet: API_2_PAH_GVBYTS_1.java
- Installation parameter: API_2_PAH_GVBYTS.install
- Load Script: API_2_PAH_GVBYTS.ldr
- Conversion parameter: API_2_PAH_GVBYTS.cnv

6.1.9.15.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler Select a TLV		
	copyValue() with a null dstBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15 Select Text String TLV		
	dstOffset ≥ dstBuffer.length **`dstBuffer.length = 5`** **`dstOffset = 5`** **`dstLength = 1`**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	dstOffset < 0 **`dstBuffer.length = 5`** **`dstOffset = -1`** **`dstLength = 1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	dstLength >dstBuffer.length **`dstBuffer.length = 5`** **`dstOffset = 0`** **`dstLength = 6`**	`ArrayIndexOutOfBoundsException` is thrown	
5	dstOffset + dstLength >dstBuffer.length **`dstBuffer.length = 5`** **`dstOffset = 3`** **`dstLength = 3`**	`ArrayIndexOutOfBoundsException` is thrown	
6	dstLength < 0 **`dstBuffer.length = 5`** **`dstOffset = 0`** **`dstLength = -1`**	`ArrayIndexOutOfBoundsException` is thrown	

7	initDisplayText() with length = 5 Select Text String TLV		
	valueOffset ≥ Text String Length **valueOffset = 6** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
8	valueOffset < 0 **valueOffset = -1** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
9	dstLength > Text String length **valueOffset = 0** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 7**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
10	valueOffset + dstLength > Text String length **valueOffset = 2** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 5**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
11	Initialise the handler		
	copyValue()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
12	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F** Select Text String TLV		
	Successful call **valueOffset = 0** **dstBuffer.length = 17** **dstOffset = 0** **dstLength = 17**	**Result of copyValue() is 17**	
13	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
14	initialise dstBuffer **dstBuffer = 55 55 … 55**		
	Successful call **valueOffset = 2** **dstBuffer.length = 20**	**Result of copyValue() is 15**	

	dstOffset = 3 **dstLength = 12**		
15	Compare buffer **buffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**	Result is 00h	

6.1.9.15.5 Test Coverage

CRR number	Test case number
N1	13, 15
N2	12, 14
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for Proactive Handler
C2	11

6.1.9.16 Method compareValue

6.1.9.16.1 Test Area Reference API_2_PAH_CPRVS_BSS

6.1.9.16.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte compareValue(short valueOffset,
                         byte[] compareBuffer,
                         short compareOffset,
                         short compareLength)
                  throws java.lang.NullPointerException,
                         java.lang.ArrayIndexOutOfBoundsException,
                         ToolkitException
```

Normal Execution

Compares the last found TLV element with a buffer:

CRRN1: returns 0 if identical.

CRRN2: returns -1 if the first miscomparing byte in simple TLV List is less than that in compareBuffer.

CRRN3: returns 1 if the first miscomparing byte in simple TLV List is greater than that in compareBuffer.

Parameter Error

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset or compareLength or both would cause access outside array bounds, or if compareLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.1.9.16.3 Test Suite files

- Test Script: API_2_PAH_CPRVS_BSS_1.scr
- Test Applet: API_2_PAH_CPRVS_BSS_1.java
- Installation parameter: API_2_PAH_CPRVS_BSS.install
- Load Script: API_2_PAH_CPRVS_BSS.ldr
- Conversion parameter: API_2_PAH_CPRVS_BSS.cnv

6.1.9.16.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler Select a TLV		
	compareValue() with a null compareBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15 Select Text String TLV		
	compareOffset ≥ compareBuffer.length **`compareBuffer.length = 5`** **`compareOffset = 5`** **`compareLength = 1`**	**`ArrayIndexOutOfBoundsException is thrown`**	
3	compareOffset < 0 **`compareBuffer.length = 5`** **`compareOffset = -1`** **`compareLength = 1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	compareLength >compareBuffer.length **`compareBuffer.length = 5`** **`compareOffset = 0`** **`compareLength = 6`**	`ArrayIndexOutOfBoundsException` is thrown	
5	compareOffset + compareLength >compareBuffer.length **`compareBuffer.length = 5`** **`compareOffset = 3`** **`compareLength = 3`**	`ArrayIndexOutOfBoundsException` is thrown	
6	compareLength < 0 **`compareBuffer.length = 5`** **`compareOffset = 0`**	`ArrayIndexOutOfBoundsException` is thrown	

	compareLength = -1		
7	initDisplayText() with length = 5 Select Text String TLV		
	valueOffset ≥ Text String Length **valueOffset = 6** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
8	valueOffset < 0 **valueOffset = -1** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
9	compareLength > Text String length **valueOffset = 0** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 7**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
10	valueOffset + compareLength > Text String length **valueOffset = 2** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 5**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
11	Initialise the handler		
	compareValue()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
12	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F** Select Text String TLV		
	Initialise compareBuffer **compareBuffer =** **04 00 01 … 0F**		
	Compare buffers **valueOffset = 0** **compareOffset = 0** **compareLength = 17**	**Result is 00h**	
13	Initialise compareBuffer **compareBuffer =** **04 00 01 02 03** **04 05 06 07 08**		

	05 0A 0B 0C 0D **0E** 10		
	Compare buffers with same parameters	**Result is -1**	
14	Initialise compareBuffer **compareBuffer =** 03 **00 01 … 0F**		
	Compare buffers with same parameters	**Result is +1**	
15	Initialise compareBuffer **compareBuffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**		
	Compare buffers **valueOffset = 2** **compareOffset = 3** **compareLength = 12**	**Result is 00h**	
16	Initialise compareBuffer **compareBuffer =** **55 55 55** 02 01 **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is -1**	
17	Initialise compareBuffer **compareBuffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A** 0A 0D **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is +1**	

6.1.9.16.5 Test Coverage

CRR number	Test case number
N1	12, 15
N2	13, 16
N3	14, 17
P1	1
P2	2, 3, 4, 5, 6

P3	**7, 8, 9, 10**
C1	**Does not apply for Proactive Handler**
C2	**11**

6.1.9.17 Method findAndCopyValue

6.1.9.17.1 Test Area Reference API_2_PAH_FACYB_BS

6.1.9.17.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short findAndCopyValue(byte tag,
                              byte[] dstBuffer,
                              short dstOffset)
                throws    java.lang.NullPointerException,
                          java.lang.ArrayIndexOutOfBoundsException,
                             ToolkitException
```

Normal Execution

CRRN1: looks for the first occurrence of a TLV element from the beginning of a TLV list and copy its value into a destination buffer.

CRRN2: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN3: if the method is successful then the corresponding TLV becomes current and dstOffset + length of the copied value is returned.

CRRN4: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if dstBuffer is null NullPointerException shall be thrown.

CRRP2: if dstOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.9.17.3 Test Suite files

- Test Script: API_2_PAH_FACYB_BS_1.scr
- Test Applet: API_2_PAH_FACYB_BS_1.java
- Installation parameter: API_2_PAH_FACYB_BS.install
- Load Script: API_2_PAH_FACYB_BS.ldr
- Conversion parameter: API_2_PAH_FACYB_BS.cnv

6.1.9.17.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	Initialise the handler		
	FindAndCopyValue() with a null dstBuffer	`NullPointerException` is thrown	
2	InitDisplayText() with length = 15		
	dstOffset ≥ dstBuffer.length **`tag = 0Dh`** **`dstBuffer.length = 20`** **`dstOffset = 20`**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	dstOffset < 0 **`dstBuffer.length = 20`** **`dstOffset = -1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	length > dstBuffer.length **`dstBuffer.length = 15`** **`dstOffset = 0`**	`ArrayIndexOutOfBoundsException` is thrown	
5	DstOffset + length >dstBuffer.length **`DstBuffer.length = 20`** **`DstOffset = 5`**	`ArrayIndexOutOfBoundsException` is thrown	
6	initDisplayText()		
	Select a TLV (tag 02h)		
	findAndCopyValue() **`tag = 03h`**	`ToolkitException.UNAVAILABLE_ELEMENT` is thrown	
	Call the `getValueLength()` method	**`ToolkitException.UNAVAILABLE_ELEMENT` is thrown.**	
7	initDisplayText() **`dcs = 4`** **`buffer = 00 01 … 0F`**		
	Successful call **`Tag = 0Dh`** **`DstBuffer.length = 17`** **`DstOffset = 0`**	**Result of `copyValue()` is 17**	
8	Compare buffer **`buffer = 04 00 01 … 0F`**	Result is 00h	
9	initialise dstBuffer **`dstBuffer = 55 55 … 55`**		
	Successful call **`dstBuffer.length = 20`** **`dstOffset = 2`**	**Result of `copyValue()` is 19**	
10	Compare buffer **`buffer =`** **`55 55 04 00 01`** **`02 03 04 05 06`**	Result is 00h	

	07 08 09 0A 0B **0C 0D 0E 0F 55**		
11	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	append a 2nd Text String TLV		
	Successful call **tag = 0Dh** **dstBuffer.length = 17** **dstOffset = 0**	**Result of copyValue() is 17**	
12	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
13	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Successful call (with tag 8Dh) **tag = 8Dh** **dstBuffer.length = 17** **dstOffset = 0**	**Result of copyValue() is 17**	
14	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
15	Append tag 0Fh **buffer = 00 01 … 0F**		
	Successful call (with tag 8Fh) **tag = 8Fh** **dstBuffer.length = 16** **dstOffset = 0**	**Result of copyValue() is 16**	
16	Compare buffer **buffer = 00 01 … 0F**	Result is 00h	

6.1.9.17.5 Test Coverage

CRR number	Test case number
N1	8, 10, 12
N2	6
N3	7, 9, 11
N4	13, 14, 15, 16
P1	1
P2	2, 3, 4, 5
C1	Does not apply for Proactive Handler

6.1.9.18 Method findAndCopyValue

6.1.9.18.1 Test Area Reference API_2_PAH_FACYBS_BSS

6.1.9.18.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public short findAndCopyValue(byte tag,
                              byte occurence,
                             short valueOffset,
                             byte[] dstBuffer,
                             short dstOffset,
                             short dstLength)
                      throws java.lang.NullPointerException,
                            java.lang.ArrayIndexOutOfBoundsException,
                             ToolkitException
```

Normal Execution

CRRN1: looks for the indicated occurrence of a TLV element from the beginning of a TLV list and copy its value into a destination buffer.

CRRN2: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN3: if the method is successful then the corresponding TLV becomes current and dstOffset + dstLength is returned.

CRRN4: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if dstBuffer is null NullPointerException shall be thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.9.18.3 Test Suite files

- Test Script: API_2_PAH_FACYBS_BSS_1.scr
- Test Applet: API_2_PAH_FACYBS_BSS_1.java
- Installation parameter: API_2_PAH_FACYBS_BSS.install
- Load Script: API_2_PAH_FACYBS_BSS.ldr
- Conversion parameter: API_2_PAH_FACYBS_BSS.cnv

6.1.9.18.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler		
	findAndCopyValue() with a null dstBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15		
	dstOffset ≥ dstBuffer.length **tag = 0Dh, occurrence = 1** **valueOffset = 0** **dstBuffer.length = 5** **dstOffset = 5** **dstLength = 1**	**ArrayIndexOutOfBoundsException is thrown**	
3	dstOffset < 0 **dstBuffer.length = 5** **dstOffset = -1** **dstLength = 1**	`ArrayIndexOutOfBoundsException` is thrown	
4	dstLength >dstBuffer.length **dstBuffer.length = 5** **dstOffset = 0** **dstLength = 6**	`ArrayIndexOutOfBoundsException` is thrown	
5	dstOffset + dstLength >dstBuffer.length **dstBuffer.length = 5** **dstOffset = 3** **dstLength = 3**	`ArrayIndexOutOfBoundsException` is thrown	
6	dstLength < 0 **dstBuffer.length = 5** **dstOffset = 0** **dstLength = -1**	`ArrayIndexOutOfBoundsException` is thrown	
7	initDisplayText() with length = 5		
	valueOffset ≥ Text String Length **tag = 0Dh, occurrence = 1** **valueOffset = 6** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
8	valueOffset < 0 **valueOffset = -1** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
9	dstLength > Text String length	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	

	valueOffset = 0 **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 7**		
10	valueOffset + dstLength > Text String length **valueOffset = 2** **dstBuffer.length = 15** **dstOffset = 0** **dstLength = 5**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
11	InitDisplayText()		
	Select a TLV (tag 02h)		
	findAndCopyValue() **tag = 0Dh** **occurrence =** 2	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the `getValueLength()` method	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
12	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Successful call **tag = 0Dh, occurrence = 1** **valueOffset = 0** **dstBuffer.length = 17** **dstOffset = 0** **dstLength = 17**	**Result of `findAndCopyValue()` is 17**	
13	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
14	initialise dstBuffer **dstBuffer = 55 55 … 55**		
	Successful call **tag = 0Dh, occurrence = 1** **valueOffset = 2** **dstBuffer.length = 20** **dstOffset = 3** **dstLength = 12**	**Result of `copyValue()` is 15**	
15	Compare buffer **buffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**	Result is 00h	
16	Append a Text String TLV		

	tag = 0D **buffer = 00 11 22 33 44 55 (no specific DCS byte)**		
	Successful call **tag = 0Dh, occurrence = 1** **valueOffset = 0** **dstBuffer.length = 17** **dstOffset = 0** **dstLength = 17**	**Result of findAndCopyValue() is 17**	
17	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
18	Successful call **tag = 0Dh, occurrence = 2** **valueOffset = 0** **dstBuffer.length = 6** **dstOffset = 0** **dstLength = 6**	**Result of findAndCopyValue() is 6**	
19	Compare buffer **buffer = 00 11 22 33 44 55**	Result is 00h	
20	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Successful call (with tag 8Dh) **tag = 8Dh** **dstBuffer.length = 17** **dstOffset = 0**	**Result of copyValue() is 17**	
21	Compare buffer **buffer = 04 00 01 … 0F**	Result is 00h	
22	Append tag 0Fh **buffer = 00 01 … 0F**		
	Successful call (with tag 8Fh) **tag = 8Fh** **dstBuffer.length = 16** **dstOffset = 0**	**Result of copyValue() is 16**	
23	Compare buffer **buffer = 00 01 … 0F**	Result is 00h	

6.1.9.18.5 Test Coverage

CRR number	Test case number
N1	13, 15, 17, 19
N2	11

N3	**12, 14, 16, 18**
N4	**20, 21, 22, 23**
P1	**1**
P2	**2, 3, 4, 5, 6**
P3	**7, 8, 9, 10**
C1	**Does not apply for Proactive Handler**

6.1.9.19 Method findAndCompareValue

6.1.9.19.1 Test Area Reference API_2_PAH_FACRB_BS

6.1.9.19.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte findAndCompareValue(byte tag,
                                byte[] compareBuffer,
                                short compareOffset)
                    throws     java.lang.NullPointerException,
                               java.lang.ArrayIndexOutOfBoundsException,
                          ToolkitException
```

Normal Execution

Looks for the first occurrence of a TLV element from beginning of a TLV list and compare its value with a buffer :

CRRN1: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN2: if the method is successful then the corresponding TLV becomes current.

CRRN3: if identical returns 0.

CRRN4: if the first miscomparing byte in simple TLV is less than that in compareBuffer returns -1.

CRRN5: if the first miscomparing byte in simple TLV is greater than that in compareBuffer returns 1.

CRRN6: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.9.19.3 Test Suite files

- Test Script: API_2_PAH_FACRB_BS_1.scr
- Test Applet: API_2_PAH_FACRB_BS_1.java
- Installation parameter: API_2_PAH_FACRB_BS.install

- Load Script: API_2_PAH_FACRB_BS.ldr
- Conversion parameter: API_2_PAH_FACRB_BS.cnv

6.1.9.19.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler		
	findAndCompareValue() with a null dstBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15		
	compareOffset ≥ compareBuffer.length **`tag = 0Dh`** **`compareBuffer.length = 20`** **`compareOffset = 20`**	**`ArrayIndexOutOfBoundsException` is thrown**	
3	compareOffset < 0 **`compareBuffer.length = 20`** **`compareOffset = -1`**	`ArrayIndexOutOfBoundsException` is thrown	
4	length > compareBuffer.length **`compareBuffer.length = 15`** **`compareOffset = 0`**	`ArrayIndexOutOfBoundsException` is thrown	
5	compareOffset + length > compareBuffer.length **`compareBuffer.length = 20`** **`compareOffset = 5`**	`ArrayIndexOutOfBoundsException` is thrown	
6	InitDisplayText()		
	Select a TLV (tag 02h)		
	findAndCompareValue() **`tag = 03h`**	`ToolkitException.UNAVAILABLE_ELEMENT` is thrown	
	Call the `getValueLength()` method	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
7	initDisplayText() **`dcs = 4`** **`buffer = 00 01 … 0F`**		
	Initialise compareBuffer **`compareBuffer =`** **`04 00 01 … 0F`**		
	Compare buffers **`tag = 0Dh`** **`compareOffset = 0`**	**Result is 00h**	
8	Verify current TLV **`getValueLength()`**	**Result is 17**	
9	Initialise compareBuffer **`compareBuffer =`** **`04 00 01 …`** `10`		

	Compare buffers with same parameters	**Result is -1**	
10	Initialise compareBuffer **compareBuffer =** 03 **00 01 … 0F**		
	Compare buffers with same parameters	**Result is +1**	
11	Initialise compareBuffer **compareBuffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**		
	Compare buffers **compareOffset = 2**	**Result is 00h**	
12	append a Text String TLV **tag = 0Dh** **buffer = 00 11 22 33 44 55**		
	Initialise compareBuffer **compareBuffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**		
	Compare buffers **compareOffset = 2**	**Result is 00h**	
13	Initialise compareBuffer **compareBuffer =** **55 55 04** 01 **01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D 0E 0F 55**		
	Compare buffers **compareOffset = 2**	**Result is -1**	
14	Initialise compareBuffer **compareBuffer =** **55 55 04 00 01** **02 03 04 05 06** **07 08 09 0A 0B** **0C 0D** 0D 10 **55**		
	Compare buffers	**Result is +1**	

	compareOffset = 2		
15	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Initialise compareBuffer **CompareBuffer = 04 00 01 … 0F**		
	Successful call (with tag 8Dh) **tag = 8Dh** **compareBuffer.length = 17** **compareOffset = 0**	**Result is 00h**	
16	Append tag 0Fh **buffer = 00 01 … 0F**		
	Initialise compareBuffer **compareBuffer = 00 01 … 0F**		
	Successful call (with tag 8Fh) **tag = 8Fh** **compareBuffer.length = 16** **compareOffset = 0**	**Result is 00h**	

6.1.9.19.5 Test Coverage

CRR number	Test case number
N1	6
N2	8
N3	7, 11, 12
N4	9, 13
N5	10, 14
N6	15, 16
P1	1
P2	2, 3, 4, 5
C1	Does not apply for Proactive Handler

6.1.9.20 Method findAndCompareValue

6.1.9.20.1 Test Area Reference API_2_PAH_FACRBBS_BSS

6.1.9.20.2 Conformance Requirement

The method with following prototype shall be compliant to its definition in the API.

```
public byte findAndCompareValue(byte tag,
                                byte occurence,
                                short valueOffset,
                                byte[] compareBuffer,
                                short compareOffset,
```

```
                                    short compareLength)
                            throws java.lang.NullPointerException,
                                   java.lang.ArrayIndexOutOfBoundsException,
                                    ToolkitException
```

Normal Execution

Looks for the indicated occurrence of a TLV element from the beginning of a TLV list and compare its value with a buffer:

CRRN1: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN2: if the method is successful then the corresponding TLV becomes current.

CRRN3: if identical 0 is returned.

CRRN4: if the first miscomparing byte in simple TLV is less than that in compareBuffer -1 is returned.

CRRN5: if the first miscomparing byte in simple TLV is greater than that in compareBuffer 1 is returned

CRRN6: The search method is comprehension required flag independent.

Parameter Error

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset or compareLength or both would cause access outside array bounds, or if compareLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, compareLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

CRRP4: if an input parameter is not valid (e.g. occurence = 0) an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException BAD_INPUT_PARAMETER.

Context Error

CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException HANDLER_NOT_AVAILABLE.

6.1.9.20.3 Test Suite files

- Test Script: API_2_PAH_FACRBBS_BSS_1.scr
- Test Applet: API_2_PAH_FACRBBS_BSS_1.java
- Installation parameter: API_2_PAH_FACRBBS_BSS.install
- Load Script: API_2_PAH_FACRBBS_BSS.ldr
- Conversion parameter: API_2_PAH_FACRBBS_BSS.cnv

6.1.9.20.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Initialise the handler		
	findAndCompareValue() with a null compareBuffer	`NullPointerException` is thrown	
2	initDisplayText() with length = 15		

	compareOffset ≥ compareBuffer.length **tag = 0Dh, occurrence = 1** **valueOffset = 0** **compareBuffer.length = 5** **compareOffset = 5** **compareLength = 1**	**ArrayIndexOutOfBoundsException is thrown**	
3	compareOffset < 0 **compareBuffer.length = 5** **compareOffset = -1** **compareLength = 1**	`ArrayIndexOutOfBoundsException` is thrown	
4	compareLength >compareBuffer.length **compareBuffer.length = 5** **compareOffset = 0** **compareLength = 6**	`ArrayIndexOutOfBoundsException` is thrown	
5	compareOffset + compareLength >compareBuffer.length **compareBuffer.length = 5** **compareOffset = 3** **compareLength = 3**	`ArrayIndexOutOfBoundsException` is thrown	
6	compareLength < 0 **compareBuffer.length = 5** **compareOffset = 0** **compareLength = -1**	`ArrayIndexOutOfBoundsException` is thrown	
7	initDisplayText() with length = 5		
	valueOffset ≥ Text String Length **tag = 0Dh, occurrence = 1** **valueOffset = 6** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
8	valueOffset < 0 **valueOffset = -1** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 1**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
9	compareLength > Text String length **valueOffset = 0** **compareBuffer.length = 15** **compareOffset = 0** **compareLength = 7**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	
10	valueOffset + compareLength > Text String length **valueOffset = 2**	**ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown**	

	compareBuffer.length = 15 **compareOffset = 0** **compareLength = 5**		
11	Invalid parameter **occurrence = 0**	**ToolkitException.BAD_INPUT_PARAMETER is thrown**	
12	InitDisplayText()		
	Select a TLV (tag 02h)		
	findAndCompareValue() **tag = 0Dh** **occurrence =** 2	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the `getValueLength()` method	**ToolkitException.UNAVAILABLE_ELEMENT is thrown.**	
13	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Initialise compareBuffer **compareBuffer =** **04 00 01 … 0F**		
	findAndCompareValue() **tag = 0Dh, occurrence = 1** **valueOffset = 0** **compareOffset = 0** **compareLength = 17**	**Result is 00h**	
14	Verify current TLV **getValueLength()**	**Result is 17**	
15	Initialise compareBuffer **compareBuffer =** **04 00 01 …** 10		
	Compare buffers with same parameters	**Result is -1**	
16	Initialise compareBuffer **compareBuffer =** 03 **00 01 … 0F**		
	Compare buffers with same parameters	**Result is +1**	
17	Initialise compareBuffer **compareBuffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A 0B 0C**		

	55 55 55 55 55		
	Compare buffers **valueOffset = 2** **compareOffset = 3** **compareLength = 12**	**Result is 00h**	
18	Initialise compareBuffer **compareBuffer =** **55 55 55** 02 01 **03 04 05 06 07** **08 09 0A 0B 0C** **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is -1**	
19	Initialise compareBuffer **compareBuffer =** **55 55 55 01 02** **03 04 05 06 07** **08 09 0A** 0A 0D **55 55 55 55 55**		
	Compare buffers with same parameters	**Result is +1**	
20	append a Text String TLV **tag = 0Dh** **buffer = 00 11 22 33 44 55**		
	Initialise compareBuffer **compareBuffer =** **04 00 01 … 0F**		
	findAndCompareValue() **tag = 0Dh, occurrence = 1** **valueOffset = 0** **compareOffset = 0** **compareLength = 17**	**Result is 00h**	
21	Initialise compareBuffer **compareBuffer =** **00 11 22 33 44 55**		
	findAndCompareValue() **tag = 0Dh, occurrence = 2** **valueOffset = 0** **compareOffset = 0** **compareLength = 6**	**Result is 00h**	
22	Initialise compareBuffer		

	compareBuffer = **00 11 22 33 44** 66		
	findAndCompareValue() **tag = 0Dh, occurrence = 2** **valueOffset = 0** **compareOffset = 0** **compareLength = 6**	**Result is -1**	
23	initDisplayText() **dcs = 4** **buffer = 00 01 … 0F**		
	Initialise compareBuffer **CompareBuffer = 04 00 01 … 0F**		
	Successful call (with tag 8Dh) **tag = 8Dh** **compareBuffer.length = 17** **compareOffset = 0**	**Result is 00h**	
24	Append tag 0Fh **buffer = 00 01 … 0F**		
	Initialise compareBuffer **compareBuffer = 00 01 … 0F**		
	Successful call (with tag 8Fh) **tag = 8Fh** **compareBuffer.length = 16** **compareOffset = 0**	**Result is 00h**	

6.1.9.20.5 Test Coverage

CRR number	Test case number
N1	12
N2	14
N3	13, 17, 20, 21
N4	15, 18, 22
N5	16, 19
N6	23, 24
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
P4	11
C1	Does not apply for Proactive Handler

6.1.10 Class ToolkitRegistry

6.1.10.1 Method allocateTimer

6.1.10.1.1 Test Area Reference: API_2_TKR_ATIM

6.1.10.1.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public byte allocateTimer()
                        throws ToolkitException
```

Normal Execution

CRRN1: the returned timer identifier shall be between 01 and 08 inclusive.

CRRN2: the returned timer identifier shall be different from a previously allocated but not released one.

CRRN3: The SIM Toolkit Framework shall trigger the applet when receiving an ENVELOPE(TIMER EXPIRATION) command for the allocated timer.

CRRN4: A call to isEventSet() method for EVENT_TIMER_EXPIRATION should return true if the applet has at least one timer allocated.

Parameters error

No requirements

Context error

CRRC1: Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE if all the timers are allocated.

CRRC2: Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE if the maximum number of timers have been allocated to this applet according to installation parameter.

6.1.10.1.3 Test suite files:

- Test Script: API_2_TKR_ATIM_1.scr
- Test Applet: API_2_TKR_ATIM_1.java
- Installation parameter: API_2_TKR_ATIM.install

  Same as default applet but with:

  - one instance with Maximum number of timers: 0

  - one instance with Maximum number of timers: 4

  - one instance with Maximum number of timers: 8

- Load Script : API_2_TKR_ATIM.ldr

  The load script installs the 3 instances.

- Conversion parameter:API_2_TKR_ATIM.cnv

6.1.10.1.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Allocates up to 8 timers** (Applet was installed with max timer parameter equals to 8).	1. Shall return false. 2. No exception shall be thrown. Timer ID returned shall be between 01 and 08 inclusive. It	

	1. IsEventSet(EVENT_TIMER_EXPIRATION) 2. for 0 to 7, allocateTimer(). 3. IsEventSet(EVENT_TIMER_EXPIRATION)	shall be different after each call. 3. Shall return true.	
2	**Allocate timers more than the maximum** (Applet was installed with max timer parameter equals to 8). The applet allocates 1 more timer.	Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE.	
3	**Check applet is Triggered by ENVELOPE(TIMER_EXPIRATION) command** Try ENVELOPE(TIMER EXPIRATION) with all timers id. Calls releaseTimer(id) each time a timer expires.	Shall trigger each time an ENVELOPE(TIMER EXPIRATION) is sent to the SIM, for Timer ID = 01 to 08.	
4	**Allocate up to 4 timers** (Applet was installed with max timer parameter equals to 4). 1. IsEventSet(EVENT_TIMER_EXPIRATION) 2. for 0 to 3, allocateTimer(). 3. IsEventSet(EVENT_TIMER_EXPIRATION)	1- Shall return false. 2- No exception shall be thrown. Each time, the returned timer identifier shall be between 01 and 08 inclusive. It shall be different after each call. 3- Shall return true.	
5	**Allocate timers more than the maximum** (Applet was installed with max timer parameter equals to 4). 1- The applet allocates 1 more timer. 2- Releases the 4 allocated timers by calling releaseTimer(id)	1- Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE.	
6	**Allocate timers more than the maximum** (Applet was installed with max timer parameter equals to 0). The applet allocates 1 more timers.	Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE.	
7	**Check allocation of timers with more than 1 applet** (One Applet was installed with max timer parameter equals to 8, Another was installed with max timer parameter equals to 4). 1- The 1st applet allocates 7 timers by calling allocateTimer() 2- The 2d applet allocates 1 timer by calling allocateTimer() 3- The 2d applet allocates 1 timer by calling allocate timer	1- Shall not throw an exception 2- Shall not throw an exception 3- Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE.	

6.1.10.1.5 Test Coverage

CRR number	***Test case number***

N1	***1,4***
N2	***1,4***
N3	***3***
N4	***1,4***
C1	***2***
C2	***5***

6.1.10.2 Method changeMenuEntry

6.1.10.2.1 Test Area Reference: API_2_TKR_CMETB_BSSBZBS

6.1.10.2.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public void changeMenuEntry(byte id,
                            byte[] menuEntry,
                            short offset,
                            short length,
                            byte nextAction,
                            boolean helpSupported,
                            byte iconQualifier,
                            short iconIdentifier)
                    throws java.lang.NullPointerException,
                           java.lang.ArrayIndexOutOfBoundsException,
                           ToolkitException
```

Normal Execution

CRRN1: The SIM Toolkit Framework shall automatically update the menu stored in the ME by issuing a SET UP MENU proactive command.The later will reflect the changes done for the entry. The SIM Toolkit Framework shall use the data of the EF sume file in order to build the SET UP MENU command.

CRRN2: If the entry was in ‘disabled’ state then the SIM Toolkit Framework shall enable it.

CRRN3: a call to isEventSet() method on EVENT_MENU_SELECTION shall return true before and after the call.

CRRN4: if method was called with length equals to 0 then in the next SETUP MENU command, the SIM Toolkit Framework shall issue, the corresponding Item Data Object TLV will contain only the ID of the entry.

CRRN5: if helpSupported was true then a call to isEventSet() method on EVENT_MENU_SELECTION_HELP_REQUEST event shall return true.

CRRN6: if helpSupported was true then after the completion of the SETUP MENU command, if an ENVELOPE(MENU_SELECTION_HELP_REQUEST) command is received by the SIM for this entry, then the SIM Toolkit framework shall trigger the applet.

CRRN7: if help supported was true, the SIM Toolkit Framework shall issue a SETUP MENU command with command qualifier = 0x80

CRRN8: if helpSupported was false and if no entries is supporting help then a call to isEventSet() method on EVENT_MENU_SELECTION_HELP_REQUEST event shall return false .

CRRN9: if helpSupported was false and if no entries is supporting help then after the completion of the SETUP MENU command, if an ENVELOPE(MENU_SELECTION_HELP_REQUEST) command is received by the SIM, then the SIM Toolkit framework shall not trigger the applet.

CRRN10:The SIM Toolkit Framework shall supply in the SET UP MENU command with the icon identifier provided in the icon identifier list within the item icon identifier list Simple TLV if all the applets registered to the EVENT_MENU_SELECTION provide it.

CRRN11: The SIM Toolkit Framework shall set in the SET UP MENU command with the Icon list qualifier transmitted to the ME as 'icon is not self explanatory' if one of the applet registered prefers this qualifier.

CRRN12: If Next Action Indicator was different from 0x00, the SIM Toolkit Framework shall issue a SETUP MENU proactive command containing an Items Next Action Indicator simple TLV with the comprehension flag set to 0.

Parameters error

CRRP1: Shall throw java.lang.NullPointerException - if menuEntry is null

CRRP2: Shall throw java.lang.ArrayIndexOutOfBoundsException - if offset would cause access outside array bounds

CRRP3: Shall throw java.lang.ArrayIndexOutOfBoundsException - if length would cause access outside array bounds

CRRP4: Shall throw java.lang.ArrayIndexOutOfBoundsException - if both offset and length would cause access outside array bounds

CRRP5: Shall throw a ToolkitException MENU_ENTRY_NOT_FOUND if Item Identifier that corresponds to id parameter, is 0x00.

Context error

CRRC1: Shall throw a ToolkitException with MENU_ENTRY_NOT_FOUND reason if the Menu Identifier is different from 0x00 but isn't associated to the calling applet instance.

CRRC2: Shall throw ALLOWED_LENGTH_EXCEEDED if the menu entry string is bigger than the allocated space

6.1.10.2.3 Test suite files:

Additional requirements for the GSM personalisation:

- contain the definition of 10 Icons
- content of EF sume shall be :
  - Title Alpha Identifier: "TOOLKIT TEST"
  - Title Icon Identifier: 0xFF

- Test Script: API_2_TKR_CMETB_BBSSBZBS_1.scr
- Test Applet: API_2_TKR_CMETB_BBSSBZBS_1.java

  entry 01 is « Init1 »

  entry 02 is « Init2 »

- Installation parameter: API_2_TKR_CMETB_BBSSBZBS.install

  Same as default applet but with:

  - Maximum text length for a menu entry: 15

  - Maximum number of menu entries: 2

  - Position / Identifier for each menu entry: 01/01, 02/02

- Load Script : API_2_TKR_CMETB_BBSSBZBS.ldr
- Conversion parameter: API_2_TKR_CMETB_BBSSBZBS.cnv

6.1.10.2.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Applet changes the entry's title by menuEntry buffer**	1- No exception shall be thrown.	On the next SIM Initialisation, The SIM shall

	1- changeMenuEntry()with parameters: Id = 0x02 MenuEntry = “UseAllBuffer” Offset = 0 Length = menuEntry.length NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0. 2- isEventSet(EVENT_MENU_SELECTION). 3- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST).	2- shall return true. 3- shall return false.	issue a SETUP MENU proactive command which contains the new text for entry ID 02.
2	**Changing the title with part of menuEntry buffer** 1- changeMenuEntry()with parameters: Id = 0x01 MenuEntry = “UsePartOfBuffer” Offset = 3 Length = 12 NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0. 2- isEventSet(EVENT_MENU_SELECTION). 3- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	1- No exception shall be thrown. 2- Shall return true. 3- Shall return false.	On the next SIM Initialisation, The SIM shall issue a SETUP MENU proactive command which contains the new text for entry ID 01.
3	**Length = 0** 1- changeMenuEntry() for entry 0x01 and entry 0x02, with parameters: Id = 0x01/0x02 MenuEntry = “LengthEquals0” Offset = 0 Length = 0 NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0. 2- isEventSet(EVENT_MENU_SELECTION). 3- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST).	1- No exception shall be thrown. 2- Shall return true. 3- shall return false.	On the next SIM Initialisation, The SIM shall issue a SETUP MENU proactive command which contains for entry 0x01and entry 0x02, no text part.
4	**Setting a next action indicator != 0** 1- changeMenuEntry()with parameters: Id = 0x02 MenuEntry = “NextActionIndic” Offset = 0 Length = menuEntry.length NextAction = 0x10 (SETUP CALL) HelpSupported = false IconQualifier = 0 IconIdentifier = 0 2- isEventSet(EVENT_MENU_SELECTION). 3- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST).	1- No exception shall be thrown. 2- Shall return true. 3- Shall return false.	On the next SIM Initialisation, The SIM shall issue a SETUP MENU proactive command which contains an Items Next Action Indicator list.
5	**Checking applet isn’t triggered by a**		

	MENU_SELECTION_HELP_REQUEST		
6	**help supported=true** 1- changeMenuEntry()with parameters: Id = 0x02 MenuEntry = “HelpSupported” Offset = 0 Length = menuEntry.length NextAction = 0 HelpSupported = true IconQualifier = 0 IconIdentifier = 0 2- isEventSet(EVENT_MENU_SELECTION). 3- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST).	1- No exception shall be thrown. 2- Shall return true. 3- Shall return true.	On the next SIM Initialisation, The SIM shall issue a SETUP MENU proactive command which contains a command qualifier 0x80.
7	**Checking applet is triggered by a MENU_SELECTION_HELP_REQUEST**		
8	**Setting icons** 1- ChangeMenuEntry() for entries 0x01,0x02, with parameters: Id = 0x01/0x02 MenuEntry = “IconQualifier” Offset = 0 Length = menuEntry.length NextAction = 0 HelpSupported = false IconQualifier = 0x01 IconIdentifier = 0x01/0x02 2- isEventSet(EVENT_MENU_SELECTION). 3- IsEventSet(EVENT_MENU_SELECTION_HELP_REQUEST).	1- No exception shall be thrown. 2- Shall return true. 3- Shall return false.	On the next SIM Initialisation, The SIM shall issue a SETUP MENU proactive command which contains an Icon Identifier List.
9	**MenuEntry is disabled** 1- disableEntry(0x01). 2- changeMenuEntry()with parameters: Id = 0x01 MenuEntry = “EnableEntry” Offset = 0 Length = menuEntry.length NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0 3- isEventSet(EVENT_MENU_SELECTION). 4- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST).	1- No exception shall be thrown. 2- No exception shall be thrown. 3- Shall return true. 4- Shall return false.	On the next SIM Initialisation, The SIM shall issue a SETUP MENU proactive command which contains the entry.
10	**MenuEntry is null**	Shall throw java.lang.NullPointerException.	
11	**Offset causes access outside array bounds** Id = 0x01 MenuEntry = “OffsetViolation” Offset = menuEntry.length Length = 1 NextAction = 0 HelpSupported = false	Shall throw java.lang.ArrayIndexOutOfBounds Exception.	

	IconQualifier = 0 IconIdentifier = 0		
12	**Big Offset causes access outside array bounds** Id = 0x01 MenuEntry = “OffsetViolation” Offset = 255 Length = 1 NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0	Shall throw java.lang.ArrayIndexOutOfBounds Exception.	
13	**Offset < 0 causes access outside array bounds** Id = 0x01 MenuEntry = “OffsetViolation” Offset = -1 Length = 1 NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0	Shall throw java.lang.ArrayIndexOutOfBounds Exception.	
14	**Length causes access outside array bounds** Id = 0x01 MenuEntry = “LengthViolation” Offset = 0 Length = MenuEntry.length + 1 NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0.	Shall throw java.lang.ArrayIndexOutOfBounds Exception.	
15	**Big Length causes access outside array bounds** Id = 0x01 MenuEntry = “LengthViolation” Offset = 0 Length = 255 NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0.	Shall throw java.lang.ArrayIndexOutOfBounds Exception.	
16	**Length < 0 causes access outside array bounds** Id = 0x01 MenuEntry = “LengthViolation” Offset = 0 Length = -1 NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0.	Shall throw java.lang.ArrayIndexOutOfBounds Exception.	
17	**Both offset and length causes access outside array bounds** Id = 0x01 MenuEntry = “BothViolation” Offset = 4 Length = 10 NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0	Shall throw java.lang.ArrayIndexOutOfBounds Exception. .	

18	**Invalid ID used** Id = 0x00 MenuEntry = contains text, != null Offset = 0 Length = menuEntry.length < 16 NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0	Shall throw a ToolkitException with MENU_ENTRY_NOT_FOUND reason code.	
19	**ID isn't allocated to a menu entry of this applet instance** Id = 0x0A MenuEntry = contains text, != null Offset = 0 Length = menuEntry.length < 16 NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0	Shall throw a ToolkitException with reason code : MENU_ENTRY_NOT_FOUND.	
20	**The text is bigger than the allocated space** Id = 0x02 MenuEntry = contains text, != null Offset = 0 Length = menuEntry.length > 15 NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0	Shall throw a ToolkitException with reason code : ALLOWED_LENGTH_EXCEEDED.	

6.1.10.2.5 Test Coverage

CRR number	***Test case number***
N1	***1,2,3,4,6,8,9***
N2	***9***
N3	***1,2,3,4,6,8,9***
N4	***3***
N5	***6***
N6	***7***
N7	***6***
N8	***1,2,3,4,8,9***
N9	***5***
N10	***8***
N11	***8***
N12	***4***
P1	***10***
P2	***11,12,13***
P3	***14,15,16***
P4	***17***
P5	***18***
C1	***19***
C2	***20***

6.1.10.3 Method clearEvent

6.1.10.3.1 Test Area Reference: API_2_TKR_CEVTB

6.1.10.3.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public void clearEvent(byte event)
                throws ToolkitException
```

Normal Execution

CRRN1: A call to isEventSet() method for a cleared event should return false after a call to clearEvent.

CRRN2: The SIM Toolkit Framework shall not trigger the applet on the occurrence of the cleared event anymore.

CRRN3: if event was EVENT_CALL_CONTROL_BY_SIM and after the call, no applet is registered to it, The SIM Toolkit Framework shall allow an applet to register to this event.

CRRN4: if event was EVENT_CALL_CONTROL_BY_SIM and one applet is still registered to these event, The SIM Toolkit Framework shall not allow an applet to register to this event.

CRRN5: if event was EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM and after the call, no applet is registered to it, The SIM Toolkit Framework shall allow an applet to register to this event.

CRRN6: if event was EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM and one applet is still registered to these event, The SIM Toolkit Framework shall not allow an applet to set this event.

CRRN7: if event was one among EVENT_EVENT_DOWNLOAD event and event was previously set and now no applet instance is registered to this event, then the SIM Toolkit Framework shall issue a SETUP EVENT LIST command with the event removed from the event list.

CRRN8: if event is not defined in gsm 0319 then no exception shall be thrown.

Parameters error

CRRP1: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event was EVENT_MENU_SELECTION.

CRRP2: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event was EVENT_MENU_SELECTION_HELP_REQUEST.

CRRP3: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event was EVENT_TIMER_EXPIRATION.

CRRP4: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event was EVENT_STATUS_COMMAND.

Context error

No requirements

6.1.10.3.3 Test suite files:

- Test Script: API_2_TKR_CEVTB_1.scr
- Test Applet: API_2_TKR_ CEVTB _1.java

  As default but applet registers to an event list which contains all defined events in GSM 0319 excepted those that aren't allowed or supported by setEvent().
- Installation parameter: API_2_TKR_ CEVTB.install
- Load Script: API_2_TKR_ CEVTB.ldr

- Conversion parameter: API_2_TKR_ CEVTB.cnv

6.1.10.3.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Clear ALLOWED unregistered events** For events ranging from –128 to 127 excepted those that aren't allowed, the applet calls : 1- clearEvent() method 2- isEventSet() method.	1- No exception is thrown each time. 2- Shall return false each time.	
2	**Clear registered events** 1- For each ALLOWED and SUPPORTED events, the applet calls setEvent() method. 2- For events ranging from –128 to 127 excepted those that aren't allowed, the applet calls : 2.1- ClearEvent() method 2.2- isEventSet() method	1- No exception shall be thrown. 2.1- No exception shall be thrown. 2.2- Shall return false.	
3	**Clearing NOT ALLOWED events** For each event among: EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION, EVENT_STATUS_COMMAND 1- The applet calls clearEvent(event) method.	1- Each time, clearEvent shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED.	
4	**Check that applet is still registered to EVENT_MENU_SELECTION**		
5	**SIM initialisation** RESET + TERMINAL PROFILE		A SETUP EVENT LIST proactive command with no events.
6	**Checking applet isn't triggered by an ENVELOPE(SMS-PP DOWNLOAD) command** An ENVELOPE(SMS-PP DOWNLOAD) is sent with a TAR, referencing applet.		

6.1.10.3.5 Test Coverage

CRR number	*Test case number*
N1	***1,2***
N2	***6***
N3	***Framework***
N4	***Framework***
N5	***Framework***
N6	***Framework***
N7	***5***
N8	***1,2***
P1	***3***
P2	***3***
P3	***3***
P4	***3***

6.1.10.4 Method disableMenuEntry

6.1.10.4.1 Test Area Reference: API_2_TKR_DMETB

6.1.10.4.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public void disableMenuEntry(byte id)
                    throws ToolkitException
```

Normal Execution

CRRN1: A call to isEventSet() method on EVENT_MENU_SELECTION shall return the same result before and after the call to disableMenuEntry() method.

CRRN2: A call to isEventSet() method on EVENT_MENU_SELECTION_HELP_REQUEST shall return the same result before and after the call to disableMenuEntry() method.

CRRN3: After invocation of this method the SIM Toolkit Framework should automatically update the menu stored in the ME .CRR

CRRN4: After invocation of this method,If there is no more enabled menu entries then the SIM Toolkit framework shall issue a SETUP MENU proactive command containing Item Data Object for Item 1 TLV with a length of zero and no value part.

Parameters error

CRRP1: shall throw a ToolkitException MENU_ENTRY_NOT_FOUND if Item Identifier, that corresponds to id parameter, is 0x00.

Context error

CRRC1: shall throw a ToolkitException with reason = ENTRY_NOT_FOUND if the menu entry doesn't exist for this applet

6.1.10.4.3 Test suite files:

Additional requirements for the GSM personalisation:

contain the definition of 10 Icons

content of EF sume shall be :

- Title Alpha Identifier: "TOOLKIT TEST"
- Title Icon Identifier: 0xFF

- Test Script: API_2_TKR_DMETB_1.scr
- Test Applet: API_2_TKR_DMETB_1.java
- Installation parameter: API_2_TKR_DMETB.install

  Same as default applet but with:

  - Maximum text length for a menu entry: 15
  - Maximum number of menu entries: 1
  - Position / Identifier for each menu entry: 01/01
- Load Script: API_2_TKR_DMETB.ldr
- Conversion parameter: API_2_TKR_DMETB.cnv

6.1.10.4.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Disabling a previously enabled entry** 1- IsEventSet(EVENT_MENU_SELECTION) 2- IsEventSet(EVENT_MENU_SELECTION_HELP_REQUEST) 3- EnableMenuEntry(0x01) 4- DisableMenuEntry(0x01). 5- IsEventSet(EVENT_MENU_SELECTION) 6- IsEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	1- Shall return true 2- Shall return false 3- No exception shall be thrown. 4- No exception shall be thrown. 5- Shall return true. 6- Shall return false.	
2	**Disabling invalid entries** For ID ranging from 0x00 to 0xFF except 0x01, the applet calls disableMenuEntry(ID) method.	Each time a Toolkit Exception with MENU_ENTRY_NOT_FOUND reason code shall be thrown.	
3	**SIM INITIALISATION** Reset + Terminal Profile.		SETUP MENU proactive command which contains a 1st ITEM Data object TLV with a length of 0.

6.1.10.4.5 Test Coverage

CRR number	*Test case number*
N1	***1***
N2	***1***
N3	***3***
N4	***3***
P1	***2***
C1	***2***

6.1.10.5 Method enableMenuEntry

6.1.10.5.1 Test Area Reference: API_2_TKR_EMETB

6.1.10.5.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public void enableMenuEntry(byte id)
                throws ToolkitException
```

Normal Execution

CRRN1: A call to isEventSet() method on EVENT_MENU_SELECTION shall return the same result before and after the call to enableMenuEntry() method.

CRRN2: A call to isEventSet() method on EVENT_MENU_SELECTION_HELP_REQUEST shall return the same result before and after the call to enableMenuEntry() method.

CRRN3: The SIM Toolkit Framework should automatically issue a SETUP MENU proactive command which does contain an ITEM SIMPLE TLV object for this entry.

Parameters error

CRRP1: shall throw a ToolkitException MENU_ENTRY_NOT_FOUND if Item Identifier, that corresponds to id parameter, is 0x00.

Context error

CRRC1 : shall throw a ToolkitException with reason = MENU_ENTRY_NOT_FOUND if the menu entry doesn't exist for this applet

6.1.10.5.3 Test suite files:

Additional requirements for the GSM personalisation:

contain the definition of 10 Icons

content of EF sume shall be :

- Title Alpha Identifier: "TOOLKIT TEST"
- Title Icon Identifier: 0xFF

- Test Script: API_2_TKR_EMETB_1.scr
- Test Applet: API_2_TKR_EMETB_1.java
- Installation parameter: API_2_TKR_EMETB.install

  Same as default applet but with:

  - Maximum text length for a menu entry: 15
  - Maximum number of menu entries: 1
  - Position / Identifier for each menu entry: 01/01
- Load Script: API_2_TKR_EMETB.ldr
- Conversion parameter: API_2_TKR_EMETB.cnv

6.1.10.5.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**enabling a previously disabled entry** 7- IsEventSet(EVENT_MENU_SELECTION) 8- IsEventSet(EVENT_MENU_SELECTION_HELP_REQUEST) 9- disableMenuEntry(0x01) 10- enableMenuEntry(0x01). 11- IsEventSet(EVENT_MENU_SELECTION) 12- IsEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	7- Shall return true 8- Shall return false 9- No exception shall be thrown. 10- No exception shall be thrown. 11- Shall return true. 12- Shall return false.	
2	**Disabling invalid entries** For ID ranging from 0x00 to 0xFF except 0x01, the applet calls enableMenuEntry(ID) method.	Each time a Toolkit Exception with MENU_ENTRY_NOT_FOUND reason code shall be thrown.	
3	**SIM Initialisation** Reset + Terminal Profile.		SETUP MENU proactive command which contains a 1st ITEM Data object TLV with a length of 0.

6.1.10.5.5 Test Coverage

CR number	*Test case number*
N1	***1***
N2	***1***
N3	***3***
P1	***2***
C1	***2***

6.1.10.6 Method getEntry

6.1.10.6.1 Test Area Reference: API_2_TKR_GETY

6.1.10.6.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public static ToolkitRegistry getEntry()
                                throws ToolkitException
```

Normal Execution

CRRN1: returns a reference to the applet ToolkitRegistry object of the calling applet.

CRRN2: Each successive call to getEntry() method shall return the same object.

Parameters error

No requirements

Context error

CRRC1: shall throw a ToolkitException with reason REGISTRY_ERROR if the caller isn't a Toolkit Applet.

6.1.10.6.3 Test suite files:

- Test Script: API_2_TKR_GETY_1.scr
- Test Applet: API_2_TKR_GETY_1.java
- Installation parameter: API_2_TKR_GETY.install
- Load Script: API_2_TKR_GETY.ldr
- Conversion parameter: API_2_TKR_GETY.cnv

6.1.10.6.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Installalation** In the constructor, the applet instance calls the getEntry() method.	Returns a not null ToolkitRegistry instance.	
2	**Check it returns the same entry** The applet calls the getEntry() method again.	Returns the same as 1, not null ToolkitRegistry instance.	

6.1.10.6.5 Test Coverage

CR number	*Test case number*
N1	***1***
N2	***2***
C1	

6.1.10.7 Method getPollInterval

6.1.10.7.1 Test Area Reference: API_2_TKR_GPOL

6.1.10.7.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public short getPollInterval()
```

Normal Execution

CRN1 : shall return a value between 1 and 15300 if applet is registered to EVENT_STATUS_COMMAND event.

CRN2 : shall return POLL_NO_DURATION value (0) if the toolkit applet is not registered to EVENT_STATUS_COMMAND event.

Parameters error

No requirements.

Context error

No requirements.

6.1.10.7.3 Test suite files:

- Test Script: API_2_TKR_GPOL_1.scr
- Test Applet: API_2_TKR_ GPOL_1.java
- Installation parameter: API_2_TKR_ GPOL.install
- Load Script: API_2_TKR_ GPOL.ldr
- Conversion parameter: API_2_TKR_ GPOL.cnv

6.1.10.7.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Applet isn't registered to EVENT_STATUS_COMMAND** `getPollInterval()`.	Shall returns 0.	
2	**Requesting max duration** 1- `RequestPollInterval(15300)` 2- `GetPollInterval`.	1- No exception shall be thrown. 2- Shall return a value between 1 and 15300.	
3	**Requesting System Duration** 1- `RequestPollInterval(POLL_SYSTEM_DURATION)` 2- `GetPollInterval()`.	1- No exception shall be thrown. 2- Shall return a value between 1 and 15300.	
4	**Requesting no Duration** 1- `RequestPollInterval(POLL_NO_DURATION)` 2- `GetPollInterval()`.	1- No exception shall be thrown. 2- Shall return 0.	

6.1.10.7.5 Test Coverage

CR number	*Test case number*
N1	*2,3*
N2	*1,4*

6.1.10.8 Method initMenuEntry

6.1.10.8.1 Test Area Reference: API_2_TKR_IMET_BSSBZBS

6.1.10.8.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public byte initMenuEntry(byte[] menuEntry,
                          short offset,
                          short length,
                          byte nextAction,
                          boolean helpSupported,
                          byte iconQualifier,
                          short iconIdentifier)
                   throws  java.lang.NullPointerException,
                          java.lang.ArrayIndexOutOfBoundsException,
                          ToolkitException
```

Normal Execution

CRRN1: The SIM Toolkit Framework shall automatically update the menu stored in the ME by issuing a SETUP MENU proactive command.The later will reflect the changes done for the entry. The SIM Toolkit Framework shall use the data of the EF sume file in order to build the SET UP MENU command.

CRRN2: a call to isEventSet() method on EVENT_MENU_SELECTION shall return true after the 1st successful call (without an exception).

CRRN3: if helpSupported was true then a following call to isEventSet() method on EVENT_MENU_SELECTION_HELP_REQUEST event shall return true .

CRRN4: if helpSupported was true then after the completion of the SETUP MENU command, if an ENVELOPE(MENU_SELECTION_HELP_REQUEST) command is received by the SIM for this entry, then the SIM Toolkit framework shall trigger the applet.

CRRN5: if help supported was true, the SIM Toolkit Framework shall issue a SETUP MENU command with command qualifier = 0x80

CRRN6: if helpSupported was false and there isn't any menu entry supporting help then a call to isEventSet() method on EVENT_MENU_SELECTION_HELP_REQUEST event shall return false.

CRRN7:The SIM Toolkit Framework shall supply in the SET UP MENU command with the icon identifier provided in the icon identifier list within the item icon identifier list Simple TLV if all the applets registered to the EVENT_MENU_SELECTION provide it.

CRRN8: The SIM Toolkit Framework shall set in the SET UP MENU command with the Icon list qualifier transmitted to the ME as 'icon is not self explanatory' if one of the applet registered prefers this qualifier.

CRRN9: If Next Action Indicator was different from 0x00, the SIM Toolkit Framework shall issue a SETUP MENU proactive command containing an Items Next Action Indicator simple TLV with the comprehension flag set to 0.

Parameters error

CRRP1: Shall throw java.lang.NullPointerException - if menuEntry is null

CRRP2: Shall throw java.lang.ArrayIndexOutOfBoundsException - if offset would cause access outside array bounds

CRRP3: Shall throw java.lang.ArrayIndexOutOfBoundsException - if length would cause access outside array bounds

CRRP4: Shall throw java.lang.ArrayIndexOutOfBoundsException - if both offset and length would cause access outside array bounds

Context error

CRRC1: Shall throw ALLOWED_LENGTH_EXCEEDED if the menu entry string is bigger than the allocated space

CRRC2: Shall throw REGISTRY_ERROR if the menu entry cannot be initialised (eg no more item data in applet loading parameter)

6.1.10.8.3 Test suite files:

Additional requirements for the GSM personalisation:

contain the definition of 10 Icons

content of EF sume shall be :

- Title Alpha Identifier: "TOOLKIT TEST"
- Title Icon Identifier: 0xFF
- Test case trigger: 1- Applet instantiation

  2- Menu selection

  3- Menu selection Help Supported

- Test Script : API_2_TKR_IMET_BSSBZBS_1.scr
- Test Applet: API_2_TKR_IMET_BSSBZBS_1.java
- Installation parameter: API_2_TKR_IMET_BSSBZBS.install

Same as default applet but with:

- Maximum text length for a menu entry: 15
- Maximum number of menu entries: 5
- Position / Identifier for each menu entry: 01/01, 02/02, 03/03, 04/04, and 05/05
- Load Script: API_2_TKR_IMET_BSSBZBS.ldr

At the end of the script the applet is loaded but not instantiated.

- Conversion parameter: API_2_TKR_IMET_BSSBZBS.cnv

6.1.10.8.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**NULL as parameter to menuEntry**	Shall throw a java.lang.NullPointerException.	
2	**Offset = menuEntry.length** MenuEntry = “ToolkitTest” Offset = 11	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
3	**Offset < 0** MenuEntry = “ToolkitTest” Offset = -1	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
4	**Offset = 255**		

	MenuEntry = “ToolkitTest” Offset = 255	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
5	**Length = menuEntry.length+1** MenuEntry = “ToolkitTest” Offset = 0 Length = 12	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
6	**Length < 0** MenuEntry = “ToolkitTest” Offset = 0 Length = -1	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
7	**Length = 255** MenuEntry = “ToolkitTest” Offset = 0 Length = 255	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
8	**Offset + length > menuEntry.length** MenuEntry = “ToolkitTest” Offset = 5 Length = 7	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
9	**MenuEntry.length > size allocated at loading for each menu entry** MenuEntry = “ToolkitTest impossible” Offset = 0 Length = 16	ALLOWED_LENGTH_EXCEEDED ToolkitException is thrown.	
10	**Successful call, menuEntry is the whole buffer** 1- initMenuEntry() MenuEntry = “TOOLKIT TEST 1” Offset = 0 Length = 14 NextAction = 0x00 HelpSupported = false IconQualifier = 0x00 IconIdentifier = 0 2- isEventSet(EVENT_MENU_SELECTION)	1- No exception shall be thrown, Shall return ID 0x01. 2- Shall return true.	
11	**Successful call, menuEntry part of a buffer** 1- initMenuEntry() MenuEntry = “1234567TOOLKIT TEST 2” Offset = 7 Length = 14 NextAction = 0x00 HelpSupported = false IconQualifier = 0x00 IconIdentifier = 0 2- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	1- No exception shall be thrown,Shall return ID 0x02. 2- Shall return false.	
12	**Successful call, menuEntry with help supported** 1- initMenuEntry() MenuEntry = “TOOLKIT TEST 3” Offset = 0 Length = 14 NextAction = 0x00	1- No exception shall be thrown, Shall return ID 0x03 2- Shall return true.	

	HelpSupported = true IconQualifier = 0x00 IconIdentifier = 0 2- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)		
13	**Successful call, menuEntry with an Icon** MenuEntry = “TOOLKIT TEST 4” Offset = 0 Length = 14 NextAction = 0x00 HelpSupported = false IconQualifier = 0x01 [icon not self explanatory] IconIdentifier = 1	1- No exception shall be thrown. 2- Shall return ID 0x04	
14	**Successful call, menuEntry with a next action indication** MenuEntry = “TOOLKIT TEST 5” Offset = 0 Length = 14 **NextAction = 0x24 [Select Item]** HelpSupported = false IconQualifier = 0x00 IconIdentifier = 0	1- No exception shall be thrown. 2- Shall return ID 0x05	
15	**Initialize more entry than allocated at loading** MenuEntry = “ToolkitTest” Offset = 0 Length = 11	REGISTRY_ERROR ToolkitException is thrown.	
16	**Reset, followed by an initialization procedure** ATR, Terminal Profile Fetch		Card shall Send a SetUpMenu Proactive command: [CommandQualifier]=help supported [AlphaId]="TOOLKIT TEST" [ItemId=1] = "TOOLKIT TEST 1" [ItemId=2] = "TOOLKIT TEST 2" [ItemId=3] = "TOOLKIT TEST 3" [ItemId=4] = "TOOLKIT TEST 4" [ItemId=5] = "TOOLKIT TEST 5" [ItemsNextAction]=0500000000024
17	**Check Applet is triggered by ENVELOPE(MENU_SELECTION) command** Menu Entry ID = 0x01		
18	**Check Applet is triggered by ENVELOPE(MENU_SELECTION) command** Menu Entry ID = 0x02		
19	**Check Applet is triggered by ENVELOPE(MENU_SELECTION) command** Menu Entry ID = 0x03		
20	**Check Applet is triggered by ENVELOPE(MENU_SELECTION) command** Menu Entry ID = 0x04		

21	**Check Applet is triggered by ENVELOPE(MENU_SELECTION) command** Menu Entry ID = 0x05		
22	**Check Applet is triggered by ENVELOPE(MENU_SELECTION_HELP_REQUEST) command** Menu Entry ID = 0x03		

6.1.10.8.5 Test Coverage

CRR number	***Test case number***
N1	***16***
N2	***10***
N3	***12***
N4	***22***
N5	***11***
N6	***12,16***
N7	***13***
N8	***13***
N9	***14,16***
P1	***1***
P2	***2,3,4***
P3	***5,6,7***
P4	***8***
C1	***9***
C2	***15***

6.1.10.9 Method isEventSet

6.1.10.9.1 Test Area Reference: API_2_TKR_IEVSB

6.1.10.9.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public boolean isEventSet(byte event)
```

Normal Execution

CRRN1: shall return true if the event is set in the Toolkit Registry for the applet

CRRN2: shall return false if the event isn't set in the Toolkit Registry for the applet

Parameters error

No requirements

Context error

No requirements

6.1.10.9.3 Test suite files:

- Test Script: API_2_TKR_IEVSB_1.scr
- Test Applet: API_2_TKR_ IEVSB _1.java
- Installation parameter: API_2_TKR_ IEVSB.install

- Load Script: API_2_TKR_ IEVSB.ldr
- Conversion parameter: API_2_TKR_ IEVSB.cnv

6.1.10.9.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Events aren't set** `Applet calls isEventSet() for each events ranging from –128 to + 127 excepted EVENT_FORMATTED_SMS_PP_ENV.`	Shall return false each time.	
2	**For EVENT_FORMATTED_SMS_PP_ENV** `isEventSet(EVENT_FORMATTED_SMS_PP_ENV)`	Shall return true.	

3	**After clearing EVENT_FORMATTED_SMS_PP_ENV**
	
	`1- clearEvent(EVENT_FORMATTED_SMS_PP_ENV)`
	
	`2- isEventSet(EVENT_FORMATTED_SMS_PP_ENV).`
		1- No exception shall be thrown.
		
		2- Shall return false.	

4	**Setting events**
	
	`1- For each SUPPORTED and ALLOWED events, applet calls :`
	
	`1.1- setEvent() method`
	
	`1.2- isEventSet() method.`
		1.1- No exception shall be thrown.
		
		1.2- Shall return true each time.	

6.1.10.9.5 Test Coverage

CRR number	*Test case number*
N1	***2,4***
N2	***1,3***

6.1.10.10 Method releaseTimer

6.1.10.10.1 Test Area Reference: API_2_TKR_RTIMB

6.1.10.10.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public void releaseTimer(byte timerIdentifier)
                throws ToolkitException
```

Normal Execution

CRRN1: if it was the last allocated timer for the applet then a following call to isEventSet() method for EVENT_TIMER_EXPIRATION should return false.

CRRN2: if applet has timers allocated then a call to isEventSet(EVENT_TIMER_EXPIRATION) shall return true.

CRRN3: The SIM Toolkit Framework shall set released timer as available for allocation after the call.

CRRN4: The SIM Toolkit Framework shall not trigger the applet when receiving an ENVELOPE(TIMER EXPIRATION) command for this timer anymore.

Parameters error

CRRP1: shall throw a ToolkitException with INVALID_TIMER_ID reason if the timer identifier isn't between 1 and 8.

Context error

CRRC1: shall throw a ToolkitException with INVALID_TIMER_ID reason if the timer is valid but isn't allocated to this applet.

6.1.10.10.3 Test suite files:

- Test Script: API_2_TKR_RTIMB_1.scr
- Test Applet: API_2_TKR_RTIMB_1.java
- Installation parameter: API_2_TKR_RTIMB.install

  As Default except, max timer which is set to 8.
- Load Script: API_2_TKR_RTIMB.ldr
- Conversion parameter: API_2_TKR_RTIMB.cnv

6.1.10.10.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Releasing not allocated timers** `For each timer ID ranging from 0x00 to 0xFF, applet calls releaseTimer(ID).`	Each time, method shall throw a ToolkitException with reason code INVALID_TIMER_ID.	

2	**Releasing allocated timers**
	`1- For 0 to 7, allocateTimer().`
	`2- For 1 to 7,`
	`2.1- ReleaseTimer(id).`
	`2.2IsEventSet(EVENT_TIMER_EXPIRATION)`
		1- No exception shall be thrown.
		2.1- Each time, no exception shall be thrown.
		2.2- Each time shall return true
		2-	

3	**Releasing invalid timer ID**
	`1- ReleaseTimer(0xFF) method`
	`2- IsEventSet(EVENT_TIMER_EXPIRATION.`
		1- Shall throw a ToolkitException with INVALID_TIMER_ID reason code.
		2- Shall return true.	

4	**Releasing last timer**
	`1- releaseTimer(0x01)`
	`2- isEventSet(EVENT_TIMER_EXPIRATION)`
		1- No exception shall be thrown.
		2- Shall return false.	

5	**Checking we can allocate timers after they have been released** `For 0 to 7, allocateTimer().`	No exception shall be thrown.	
6	**Releasing all timers.** `For 1 to 8, releaseTimer(id).`	No exception shall be thrown.	
7	**Checking applet isn't triggered by ENVELOPE(TIMER_EXPIRATION) command**		

6.1.10.10.5 Test Coverage

CRR number	*Test case number*
N1	***4***
N2	***2,3***
N3	***5***
N4	***7***
P1	***1***
C1	***1***

6.1.10.11 Method requestPollInterval

6.1.10.11.1 Test Area Reference: API_2_TKR_RPOLS

6.1.10.11.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public void requestPollInterval(short duration)
                    throws ToolkitException
```

Normal Execution

CRRN1: a successive call to isEventSet() method on EVENT_STATUS_COMMAND shall return true if duration was between 1 and 15300.

CRRN2: the SIM Toolkit Framework should issue a Poll Interval proactive command with a value if duration's value was between 1 and 15300 inclusive.

CRRN3: The SIM Toolkit Framework shall trigger the applet when receiving a STATUS command if duration's value was between 1 and 15300 inclusive.

CRRN4: a successive call to isEventSet() method on EVENT_STATUS_COMMAND shall return true if duration was POLL_SYSTEM_DURATION.

CRRN5: the SIM Toolkit Framework should issue a Poll Interval proactive command with a value if duration's value was POLL_SYSTEM_DURATION.

CRRN6: The SIM Toolkit Framework shall trigger the applet when receiving a STATUS command if duration's value was POLL_SYSTEM_DURATION.

CRRN7: a successive call to isEventSet() method on EVENT_STATUS_COMMAND shall return false if duration was POLL_NO_DURATION.

CRRN8: the SIM Toolkit Framework should issue a Polling Off proactive command if value was POLL_NO_DURATION.

CRRN9: The SIM Toolkit Framework shall not trigger the applet when receiving a STATUS command if duration was POLL_NO_DURATION.

Parameters error

CRRP1: the method should throw a ToolkitException with REGISTRY_ERROR reason if duration was > 15300.

CRRP2: the method should throw a ToolkitException with REGISTRY_ERROR reason if duration was < 0.

Context error

No Requirements

6.1.10.11.3 Test suite files:

- Test Script: API_2_TKR_RPOLS_1.scr

- Test Applet: API_2_TKR_RPOLS_1.java
- Installation parameter: API_2_TKR_RPOLS.install
- Load Script: API_2_TKR_RPOLS.ldr
- Conversion parameter: API_2_TKR_RPOLS.cnv

6.1.10.11.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Requesting a value between 1 and 15300 s** 1- isEventSet(EVENT_STATUS_COMMMAND) 2- For duration ranging from 1 to 15300, RequestPollInterval(duration). 3- IsEventSet(EVENT_STATUS_COMMAND).	1- Shall return false. 2- No exception shall be thrown. 3- Shall return true.	
2	**SIM Initialisation**		The SIM shall issue a POLL INTERVAL proactive command with a duration between 1 and 15300.
3	**Check Applet is triggered by a STATUS command**		
4	**Requesting POLL SYSTEM DURATION** 1- isEventSet(EVENT_STATUS_COMMMAND). 2- RequestPollInterval(POLL_SYSTEM_DURATION). 3- IsEventSet(EVENT_STATUS_COMMAND).	1- Shall return true. 2- No exception shall be thrown. 3- Shall return true.	
5	**SIM Initialisation**		The SIM shall issue a POLL INTERVAL proactive command with a duration between 1 and 15300.
6	**Check Applet is triggered by a STATUS command**		
7	**Requesting invalid duration** For duration ranging from 0x3BC5 to 0xFFFF, RequestPollInterval(duration)	Each time, a ToolkitException with REGISTRY_ERROR reason code, shall be thrown.	
8	**Requesting POLL NO DURATION** 1- isEventSet(EVENT_STATUS_COMMMAND) 2- RequestPollInterval(POLL_NO_DURATION) 3- IsEventSet(EVENT_STATUS_COMMAND)	1- Shall return true. 2- No exception shall be thrown. 3- Shall return false.	
9	**SIM Initialisation**		The SIM shall issue a POLLING OFF.
10	**Check Applet isn't triggered by an STATUS command.**		

6.1.10.11.5 Test Coverage

CRR number	*Test case number*
N1	***1***
N2	***2***
N3	***3***

N4	***4***
N5	***5***
N6	***6***
N7	***8***
N8	***9***
N9	***10***
P1	***7***
P2	***7***

6.1.10.12 Method setEvent

6.1.10.12.1 Test Area Reference: API_2_TKR_SEVTB

6.1.10.12.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public void setEvent(byte id)
            throws ToolkitException
```

Normal Execution

CRRN1: a following call to isEventSet() method with the same event id shall answer true for the applet.

CRRN2: the SIM Toolkit Framework shall trigger the applet if an occurrence of the set event happens.

CRRN3: this method shall accept all the events defined in GSM 0319 excepted : EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION , EVENT_STATUS_COMMAND

Parameters error

CRRP1: shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason if event is 0 or event ranges from 20 to 127.

CRRP2: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if event is EVENT_MENU_SELECTION.

CRRP3: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if event is EVENT_MENU_SELECTION_HELP_REQUEST.

CRRP4: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if event is EVENT_TIMER_EXPIRATION.

CRRP5: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if event is EVENT_STATUS_COMMAND.

Context error

CRRC1: shall throw a ToolkitException with EVENT_ALREADY_REGISTERED if event is EVENT_CALL_CONTROL_BY_SIM but another applet is already registered to it.

CRRC2: shall throw a ToolkitException with EVENT_ALREADY_REGISTERED if event is EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM but another applet is already registered to it.

6.1.10.12.3 Test suite files:

- Test Script: API_2_TKR_SEVTB_1.scr
- Test Applet: API_2_TKR_SEVTB_1.java
- Installation parameter: API_2_TKR_SEVTB.install
- Load Script: API_2_TKR_SEVTB.ldr
- Conversion parameter: API_2_TKR_SEVTB.cnv

6.1.10.12.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Applet is triggered by ENVELOPE(SMS_PP_DOWNLOAD) command.** Because applet does call setEvent() during installation, if it is triggered, it means setEvent() works for this event. So Applet records it as result of test 1.		
2	**Setting ALLOWED and SUPPORTED events** 1- For all events defined in GSM 0319 and allowed: 1.1- clearEvent(event) 1.2- isEventSet(event) 1.3- setEvent(event) 1.4- isEventSet(event) 1.5- clearEvent(event)	1.1- No exception shall be thrown. 1.2- Shall return false. 1.3- No exception shall be thrown. 1.4- Shall return true. 1.5- No exception shall be thrown.	
3	**Event 0**	Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason code.	
4	**Events from 20 to 127**	Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason code.	
5	**Setting EVENT_MENU_SELECTION**	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
6	**Setting EVENT_MENU_SELECTION_HELP_REQUEST**	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
7	**Setting EVENT_TIMER_EXPIRATION**	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
8	**Setting EVENT_STATUS_COMMAND**	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
9	**Setting EVENT_CALL_CONTROL_BY_SIM 2 times** 1- SetEvent(EVENT_CALL_CONTROL_BY_SIM) 2- SetEvent(EVENT_CALL_CONTROL_BY_SIM)	1- Shall not throw an exception 2- Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	
10	**Setting EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM 2 times**	1- Shall not throw an exception 2- Shall throw a ToolkitException with	

	1- SetEvent(EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM) 2- SetEvent(EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM)	EVENT_ALREADY_REGISTERED reason code.	
11	**Check applet is triggered by an ENVELOPE(CALL_CONTROL)**		
12	**Check applet is triggered by an ENVELOPE(MO_SHORT_MESSAGE_CONTROL)**		

6.1.10.12.5 Test Coverage

CRR number	*Test case number*
N1	***2***
N2	***1,11,12***
N3	***2***
P1	***3,4***
P2	***5***
P3	***6***
P4	***7***
P5	***8***
C1	***9***
C2	***10***

6.1.10.13 Method setEventList

6.1.10.13.1 Test Area Reference: API_2_TKR_SEVL_BSS

6.1.10.13.2 Conformance Requirement:

The method with following header shall be compliant to its definition in the API.

```
public void setEventList(byte[] eventList,
                         short offset,
                         short length)
                  throws java.lang.NullPointerException,
                         java.lang.ArrayIndexOutOfBoundsException,
                         ToolkitException
```

Normal Execution

CRRN1: for all events set successfully by this method, a call to isEventSet() method should return true.

CRRN2: the SIM Toolkit Framework shall trigger the applet if an occurrence of one of the successfully registered events happens.

CRRN3: this method shall accept all the events defined in GSM 0319 excepted : EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION , EVENT_STATUS_COMMAND.

Parameters error

CRRP1: shall throw a java.lang.NullPointerException if eventList is null.

CRRP2: shall throw a java.lang.ArrayIndexOutOfBoundsException if offset would cause access outside array bounds.

CRRP3: shall throw a java.lang.ArrayIndexOutOfBoundsException if length would cause access outside array bounds.

CRRP4: shall throw a java.lang.ArrayIndexOutOfBoundsException if both offset and length would cause access outside array bounds.

CRRP5: shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason if event is 0 or event ranges from 20 to 127.

CRRP6: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if eventList contains EVENT_MENU_SELECTION.

CRRP7: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if eventList contains EVENT_MENU_SELECTION_HELP_REQUEST.

CRRP8: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if eventList contains EVENT_TIMER_EXPIRATION.

CRRP9: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if eventList contains EVENT_STATUS_COMMAND.

Context error

CRRC1: shall throw a ToolkitException with EVENT_ALREADY_REGISTERED if eventList contains EVENT_CALL_CONTROL_BY_SIM but another applet is already registered to it.

CRRC2: shall throw a ToolkitException with EVENT_ALREADY_REGISTERED if eventList contains EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM but another applet is already registered to it.

6.1.10.13.3 Test suite files:

- Test Script: API_2_TKR_SEVL_BSS_1.scr
- Test Applet: API_2_TKR_ SEVL_BSS _1.java
- Installation parameter: API_2_TKR_ SEVL_BSS.install
- Load Script: API_2_TKR_ SEVL_BSS.ldr
- Conversion parameter: API_2_TKR_ SEVL_BSS.cnv

6.1.10.13.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	**Registering all eventList buffer**
	
	`EventList = all allowed events defined in GSM 0319.`
	
	`1- For each event in EventList ClearEvent(event)`
	
	`2- SetEventList(eventList)`
	
	`Offset = 0`
	`Length = eventList.lentgh`
	
	`3- For all events in eventList IsEventSet(event)`
	
	`4- For each event in EventList ClearEvent(event)`
		1- No exception shall be thrown.
		
		2- No exception shall be thrown.
		
		3- Each time shall return true.
		
		4- No exception shall be thrown.	

2	**Registering part of eventList buffer**	1- No exception shall be thrown.	

	EventList = all allowed events defined in GSM 0319. 1- For each event in EventList ClearEvent(event) 2- SetEventList(eventList, offset, length) Offset > 0 Length = eventList.lentgh – offset 3- For all events in eventList IsEventSet(event) 4- For each event in EventList ClearEvent(event)	2- No exception shall be thrown. 3- Each time shall return true for events ranging from offset to offset+length else shall return false. 4- No exception shall be thrown.	
3	**Null buffer** EventList = null	Shall throw a java.lang.NullPointerException Exception	
4	**Out of bounds offset** Offset = eventList.length Length = 1	Shall throw a java.lang.ArrayIndexOutOfBounds Exception	
5	**Out of bounds and big offset** Offset = 255 Length = 1	Shall throw a java.lang.ArrayIndexOutOfBounds Exception	
6	**Offset < 0** Offset = -1 Length = 1	Shall throw a java.lang.ArrayIndexOutOfBounds Exception	
7	**Out of bounds length** Offset = 0 Length = eventList.length + 1	Shall throw a java.lang.ArrayIndexOutOfBounds Exception	
8	**Out of bounds and big length** Offset = 0 Length = 255	Shall throw a java.lang.ArrayIndexOutOfBounds Exception	
9	**Length < 0** Offset = 0 Length = -1	Shall throw a java.lang.ArrayIndexOutOfBounds Exception	
10	**Out of bounds offset + Length** Offset + length > eventList.length + 1	Shall throw a java.lang.ArrayIndexOutOfBounds Exception	
11	**Event 0**	Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason code.	
12	**Events from 20 to 127**	Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason code.	
13	**EVENT_MENU_SELECTION**	Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	
14	**EVENT_MENU_SELECTION_HELP_REQUEST**		

		Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	
15	**EVENT_TIMER_EXPIRATION**	Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	
16	**EVENT_STATUS_COMMAND**	Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	
17	**EVENT_CALL_CONTROL_BY_SIM already registered.** 1- call to setEventList() with eventList containing EVENT_CALL_CONTROL_BY_SIM 2- re-call setEventList() with EVENT_CALL_CONTROL_BY_SIM	1- No exception shall be thrown 2- Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	
18	**EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM already registered** 1- call to setEventList() with eventList containing EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM 2- re-call setEventList() with EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM	1- No exception shall be thrown 2- Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	
19	**Check Triggering On EVENT_CALL_CONTROL_BY_SIM** An ENVELOPE(CALL_CONTROL) is sent to the SIM which has to trigger applet. Applet records this triggering as result of test 19.		
20	**Check Triggering On EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM** An ENVELOPE(MO_SHORT_MESSAGE_CONTROL) is sent to the SIM which has to trigger applet. Applet records this triggering as result of test 20.		

6.1.10.13.5 Test Coverage

CRR number	***Test case number***
N1	***1,2***
N2	***19,20***
N3	***1,2***
P1	***3***
P2	***4,5,6***
P3	***7,8,9***
P4	***10***
P5	***11,12***

P6	***13***
P7	***14***
P8	***15***
P9	***16***
C1	***17***
C2	***18***

6.1.11 Class ToolkitException

6.1.11.1 Exception Constants

6.1.11.1.1 Test Area Reference: API_2_TKE_CONS

6.1.11.1.2 Conformance Requirement:

There is no API, only constants.

Normal Execution

CRRN1 : The Constants of the class ToolkitException shall all have the same name and value defined in the GSM03.19 .

Parameters error

No requirements

Context error

No requirements

6.1.11.1.3 Test suite files:

No additional requirements for the GSM personalisation

- Test Script: API_2_TKE_CONS_1.scr
- Test Applet: API_2_TKE_CONS_1.java
- Installation parameter: API_2_TKE_CONS.install (Same as default applet)
- Load Script: API_2_TKE_CONS.ldr

  At the end of the script the applet is loaded but not instantiated.
- Conversion parameter: API_2_TKE_CONS.cnv

6.1.11.1.4 Test Procedure

Id	Description
01	Check constant HANDLER_OVERFLOW=1
03	Check constant UNAVAILABLE_ELEMENT=3
04	Check constant MENU_ENTRY_NOT_FOUND=4
05	Check constant REGISTRY_ERROR=5
06	Check constant EVENT_NOT_SUPPORTED=6
07	Check constant EVENT_ALREADY_REGISTERED=7
08	Check constant OUT_OF_TLV_BOUDARIES=8
09	Check constant ME_PROFILE_NOT_AVAILABLE=9
10	Check constant ALLOWED_LENGTH_EXCEEDED=10
11	Check constant NO_TIMER_AVAILABLE=11
12	Check constant INVALID_TIMER_ID=12
13	Check constant EVENT_NOT_ALLOWED=13
14	Check constant BAD_INPUT_PARAMETER=14

6.1.11.1.5 Test Coverage

CR number	*Test case number*
N1	***1-14***

6.1.11.2 Constructor ToolkitException

6.1.11.2.1 Test Area Reference: API_2_TKE_TOOES

6.1.11.2.2 Conformance Requirement:

The constructor with following headershall compliant to its definition in the API.

```
public ToolkitException(short reason)
```

Normal Execution

CRRN1: Construct a ToolkitException instance with the specified reason.

Parameters error

No requirements

Context error

No requirements

6.1.11.2.3 Test suite files:

No additional requirements for the GSM personalisation:

Test Script: API_2_TKE_ TOOES.scr

Test Applet: API_2_TKE_ TOOES.java

Installation parameter: API_2_TKE_ TOOES.install (Same as default applet)

Load Script: API_2_TKE_ TOOES.ldr

Conversion parameter: API_1_TKE_ TOOES.cnv

6.1.11.2.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**0 < reason < 15** reason='EVENT_ALREADY_REGISTERED'	EVENT_ALREADY_REGISTERED `ToolkitException` is thrown	

6.1.11.2.5 Test Coverage

CRR number	*Test case number*
N1	***1***

6.1.11.3 Method throwIt

6.1.11.3.1 Test Area Reference: API_2_TKE_THITS

6.1.11.3.2 Conformance Requirement:

The method with following header shall compliant to its definition in the API.

```
public static void throwIt(short reason)
                  throws  ToolkitException
```

Normal Execution

CRRN1: Throws the JCRE instance of the ToolkitException class with the specified reason.

CRRN2: extends javacard.framework.CardRuntimeException

Parameters error

No requirements

Context error

No requirements

6.1.11.3.3 Test suite files:

No additional requirements for the GSM personalisation:

- Test Script: API_2_TKE_ THITS.scr
- Test Applet: API_2_TKE_ THITS.java
- Installation parameter: API_2_TKE_ THITS.install (Same as default applet)
- Load Script: API_2_TKE_ THITS.ldr
- Conversion parameter: API_2_TKE_ THITS.cnv

6.1.11.3.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Throws the JCRE instance of ToolkitException with the specified reason**	`Reason (specified)`	can't be checked because the status word's value is not specified
2	**ToolkitException extends javacard.framework.CardRuntimeException**	`Reason (specified)`	

6.1.11.3.5 Test Coverage

CRR number	*Test case number*
N1	***1***
N2	***2***

6.2 Package sim.access

6.2.1 Interface SIMView

Note: The Test applet shall be run on a class that implements this interface.

6.2.1.1 Constants

6.2.1.1.1 Test Area Reference: API_1_SVW_CONST

6.2.1.1.2 Conformance Requirements

This section does not describe the conformance requirements for a method, but rather for the constants of the interface.

Normal Execution

CRRN1: The constants shall have the same name and value that is defined in GSM 03.19.

Parameters error

No requirements

Context error

No requirements

6.2.1.1.3 Test Suite Files

No additional requirements for the GSM personalisation

- Test Script: API_1_SVW_CONST_1.scr
- Test Applet: API_1_SVW_CONST_1.java
- Installation Parameter: API_1_SVW_CONST.install
- Load Script: API_1_SVW_CONST.ldr
- Conversion parameter: API_1_SVW_CONST.cnv

6.2.1.1.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
01	Check constant FID_MF = 0x3F00		
02	Check constant FID_DF_TELECOM = 0x7F10		
03	Check constant FID_DF_GSM = 0x7F20		
04	Check constant FID_DF_DCS_1800 = 0x7F21		
05	Check constant FID_DF_IS_41 = 0x7F22		
06	Check constant FID_DF_FP_CTS = 0x7F23		
07	Check constant FID_DF_Graphics = 0x5F50		
08	Check constant FID_DF_IRIDIUM = 0x5F30		
09	Check constant FID_DF_GLOBALSTAR = 0x5F31		
10	Check constant FID_DF_ICO = 0x5F32		
11	Check constant FID_DF_ACES = 0x5F33		
12	Check constant FID_DF_PCS_1900 = 0x5F40		
13	Check constant FID_DF_CTS = 0x5F60		
14	Check constant FID_DF_SOLSA = 0x5F70		
15	Check constant FID_EF_ICCID = 0x2FE2		

Id	Description	API Expectation	APDU Expectation
16	Check constant FID_EF_ELP = 0x2F05		
17	Check constant FID_EF_ADN = 0x6F3A		
18	Check constant FID_EF_FDN = 0x6F3B		
19	Check constant FID_EF_SMS = 0x6F3C		
20	Check constant FID_EF_CCP = 0x6F3D		
21	Check constant FID_EF_MSISDN = 0x6F40		
22	Check constant FID_EF_SMSP = 0x6F42		
23	Check constant FID_EF_SMSS = 0x6F43		
24	Check constant FID_EF_LND = 0x6F44		
25	Check constant FID_EF_SND = 0x6F49		
26	Check constant FID_EF_EXT1 = 0x6F4A		
27	Check constant FID_EF_EXT2 = 0x6F4B		
28	Check constant FID_EF_EXT3 = 0x6F4C		
29	Check constant FID_EF_BDN = 0x6F4D		
30	Check constant FID_EF_EXT4 = 0x6F4E		
31	Check constant FID_EF_SMSR = 0x6F47		
32	Check constant FID_EF_IMG = 0x4F20		
33	Check constant FID_EF_LP = 0x6F05		
34	Check constant FID_EF_IMSI = 0x6F07		
35	Check constant FID_EF_KC = 0x6F20		
36	Check constant FID_EF_PLMNSEL = 0x6F30		
37	Check constant FID_EF_HPLMN = 0x6F31		
38	Check constant FID_EF_ACMMAX = 0x6F37		
39	Check constant FID_EF_SST = 0x6F38		
40	Check constant FID_EF_ACM = 0x6F39		
41	Check constant FID_EF_GID1 = 0x6F3E		
42	Check constant FID_EF_GID2 = 0x6F3F		
43	Check constant FID_EF_SPN = 0x6F46		
44	Check constant FID_EF_PUCT = 0x6F41		
45	Check constant FID_EF_CBMI = 0x6F45		
46	Check constant FID_EF_BCCH = 0x6F74		
47	Check constant FID_EF_ACC = 0x6F78		
48	Check constant FID_EF_FPLMN = 0x6F7B		
49	Check constant FID_EF_LOCI = 0x6F7E		
50	Check constant FID_EF_AD = 0x6FAD		
51	Check constant FID_EF_PHASE = 0x6FAE		
52	Check constant FID_EF_VGCS = 0x6FB1		
53	Check constant FID_EF_VGCSS = 0x6FB2		
54	Check constant FID_EF_VBS = 0x6FB3		
55	Check constant FID_EF_VBSS = 0x6FB4		
56	Check constant FID_EF_EMLPP = 0x6FB5		
57	Check constant FID_EF_AAEM = 0x6FB6		
58	Check constant FID_EF_CBMID = 0x6F48		
59	Check constant FID_EF_ECC = 0x6FB7		
60	Check constant FID_EF_CBMIR = 0x6F50		
61	Check constant FID_EF_DCK = 0x6F2C		
62	Check constant FID_EF_CNL = 0x6F32		
63	Check constant FID_EF_NIA = 0x6F51		
64	Check constant FID_EF_KCGPRS = 0x6F52		
65	Check constant FID_EF_LOCIGPRS = 0x6F53		
66	Check constant FID_EF_SUME = 0x6F54		
67	Check constant FID_EF_SAI = 0x4F30		
68	Check constant FID_EF_SLL = 0x4F31		
69	Check constant REC_ACC_MODE_MODE_NEXT = 0x02		
70	Check constant REC_ACC_MODE_PREVIOUS = 0x03		
71	Check constant REC_ACC_MODE_ABSOLUTE_CURRENT = 0x04		
72	Check constant SEEK_FROM_BEGINNING_FORWARD = 0x00		
73	Check constant SEEK_FROM_END_BACKWARD = 0x01		
74	Check constant SEEK_FROM_NEXT_FORWARD = 0x02		
75	Check constant SEEK_FROM_PREVIOUS_BACKWARD = 0x03		

6.2.1.1.5 Test Coverage

CR Number	Test Case Number
CRRN1	**1-75**

6.2.1.2 Method select

6.2.1.2.1 Test Area Reference: API_1_SVW_ SLCTS_BSS

6.2.1.2.2 Conformance Requirements

The method with the following header shall be compliant to its definition in the API.

```
public short select(short fid,
                    byte[] fci,
                    short fciOffset,
                    short fciLength)
             throws java.lang.NullPointerException,
                    java.lang.ArrayIndexOutOfBoundsException,
                    SIMViewException
```

Normal Execution

CRRN1: If the desired file is selected, the length of the FCI (File Control Information) which has been written to the array fci is returned.

CRRN2: If the length fciLength is greater than or equal to the length of the response, the whole response is copied into the array fci and the length of the FCI which has been written to the array fci is returned.

CRRN3: If the length fciLength is less than the length of the response, the first part of the response is copied into the array fci and the length of the FCI which has been written to the array fci is returned.

CRRN4: After selecting a DF/MF no EF is selected.

CRRN5: After selecting a linear fixed EF no record is selected.

CRRN6: After selecting a cyclic EF the first record which is the last updated record is selected.

CRRN7: The current files (file context) of any other applets shall not be changed. [03.19 - §5.2]

CRRN8: The information returned by fci shall be formatted as follows:

Fci parameters/data in case of selecting an MF or DF:

Byte(s)	Description	Length
1 – 2	RFU	2
3 – 4	Total amount of free memory of the selected directory	2
5 – 6	File ID	2
7	Type of file: 01 Hex for MF, 02 Hex for DF	1
8 – 12	RFU	5
13	Length of the following data (byte 14 to the end)	1
14	File characteristics: See details No 1 of GSM 11.11 SELECT command	1
15	Number of DFs which are a direct child of the current directory	1
16	Number of EFs which are a direct child of the current directory	1
17	Number of CHVs, UNBLOCK CHVs and administrative codes	1
18	RFU	1
19	CHV1 status: See details No 2 of GSM 11.11 SELECT command	1
20	UNBLOCK CHV1 status: See details No 2 of GSM 11.11 SELECT command	1
21	CHV2 status: See details No 2 of GSM 11.11 SELECT command	1
22	UNBLOCK CHV2 status: See details No 2 of GSM 11.11 SELECT command	1
23	RFU	1
24 - 34	Reserved for the administrative management	0 = lgth = 11

Fci parameters/data in case of selecting an EF:

Byte(s)	Description	Length
1 - 2	RFU	2
3 - 4	File size (for transparent EF: the length of the body part of the EF) (for linear fixed or cyclic EF: record length multiplied by the number of records of the EF)	2
5 - 6	File ID	2
7	Type of file: 04 Hex for EF	1
8	See details No 3 of GSM 11.11 SELECT command	1
9 - 11	Access conditions (see 9.3)	3
12	File-Status: Bit 0 = 0: File is invalidated; Bit 0 = 1:File is rehabilitated; Bit 2 = 0: File can't be read/updated if invalidated. Bit 2 = 1: File can be read/updated if invalidated.	1
13	Length of the following data (byte 14 to the end)	1
14	Structure of EF: 00 Hex is transparent, 01 Hex is linear fixed EF, 03 Hex is cyclic	1
15	Length of a record in bytes: For a transparent EF the value is 0.	1
16 and following	RFU	-

CRRN9: The file with a File-ID that matches fid shall be found according to the following selection rules:

- An immediate child EF or DF of the current MF/DF can be selected,
- A sibling DF of the current DF can be selected,
- The current MF/DF it self can be selected,
- The parent MF/DF of the current DF can be selected,
- The MF can always be selected.

Parameter Errors

CRRP1: If the array fci is null, an instance of NullPointerException shall be thrown.

CRRP2: If the length of the response to be copied into the array fci plus fciOffset is greater than fci.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: If fciOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP4: If fciLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP5: If fciOffset plus fciLength, is greater than the length of the array fci.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

Context Errors

CRRC1: If the file with a File-ID which matches fid could not be found according to the selection rules

- An immediate child EF or DF of the current MF/DF can be selected,
- A sibling DF of the current DF can be selected,
- The current MF/DF it self can be selected,
- The parent MF/DF of the current DF can be selected
- The MF can always be selected

an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.FILE_NOT_FOUND.

CRRC2: If the method call causes a memory problem (e.g. memory access error), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.MEMORY_PROBLEM.

CRRC3: If the method call causes an error to occur that is not expected and thus not handled, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INTERNAL_ERROR.

6.2.1.2.3 Test Suite Files

Additional requirements for the GSM personalisation:

...

- Test Script: API_1_SVW_SLCTS_BSS_1.scr
- Test Applet: API_1_SVW_SLCTS_BSS_1.java
- Installation Parameter: API_1_SVW_SLCTS_BSS.install
- Load Script: API_1_SVW_SLCTS_BSS.ldr
- Conversion parameter: API_1_SVW_SLCTS_BSS.cnv

6.2.1.2.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	**Select EF_{ICCID} in MF (Transparent EF)**
	`fid = SIMView.FID_EF_ICCID`
	`byte[] fci = new byte[34]`
	`fciOffset = 0`
	`fciLength = 20`
	`select()`
		No exception shall be thrown.
		Shall return 14 or more (Transparent EF).
		fci shall contain the entire response.
		<Description of fci>	

2	**Select EF_{ICCID} in MF (Transparent EF)**
	`fid = SIMView.FID_EF_ICCID`
	`fciOffset = 0`
	`fciLength = 13`
	`select()`
		No exception shall be thrown.
		Shall return 13.
		fci shall contain the first 13 bytes of the response.	

Id	Description	API Expectation	APDU Expectation

3	**Select DF_{GSM} in MF**
	`fid = SIMView.FID_DF_GSM`
	`fciOffset = 0`
	`fciLength = 34`
	`select()`
		No exception shall be thrown.
		Shall return 34.
		fci shall contain the entire response.
		<Description of fci>	

4	**Select EF_{ACM} in DF_{GSM} (CyclicEF)**
	`fid = SIMView.FID_EF_ACM`
	`fciOffset = 0`
	`fciLength = 20`
	`select()`
		No exception shall be thrown.
		Shall return 15 or more. (Cyclic EF)
		fci shall contain the first 15 or more bytes of the response.
		Byte 15 shall have the value 3 (length of record).	

5	**Select MF**
	`fid = SIMView.FID_MF`
	`fciOffset = 0`
	`fciLength = 34`
	`select()`
		No exception shall be thrown.
		Shall return 34.
		fci shall contain the entire response.	

6	**Select $DF_{TELECOM}$ in MF**
	`fid = SIMView.FID_DF_TELECOM`
	`fci[0] = fci[1] = 0x05`
	`fciOffset = 2`
	`fciLength = 20`
	`select()`
		No exception shall be thrown.
		Shall return 20.
		fci shall contain the first 20 bytes of the response starting at index 2.
		The first two bytes shall (still) have the value 0x05.	

7	**Select EF_{FDN} in $DF_{TELECOM}$ (Linear FixedEF)**
	`fid = SIMView.FID_EF_FDN`
	`fciOffset = 0`
	`fciLength = 15`
	`select()`
		No exception shall be thrown.
		Shall return 15.
		fci shall contain the first 15 bytes of the response.
		Byte 15 shall have the value 28 (length of record).	

8	**fci is null**
	`fid = SIMView.FID_EF_FDN`
	`byte[] fci = null`
	`fciOffset = 0`
	`fciLength = 15`
	`select()`
		Shall throw java.lang.NullPointerException.	

9	**response + fciOffset > fci.length**
	`fid = SIMView.FID_EF_FDN`
	`byte[] fci = new byte[34]`
	`fciOffset = 20`
	`fciLength = 15`
	`select()`
		Shall throw java.lang.ArrayIndexOutOfBounds Exception.	

10	**fciOffset < 0**
	`fid = SIMView.FID_EF_FDN`
	`fciOffset = -1`
	`fciLength = 15`
	`select()`
		Shall throw java.lang.ArrayIndexOutOfBounds Exception.	

11	**fciLength < 0**
	`fid = SIMView.FID_EF_FDN`
	`fciOffset = 0`
	`fciLength = -1`
	`select()`
		Shall throw java.lang.ArrayIndexOutOfBounds Exception.	

12	**fciOffset + fciLength > fci.length**
	`fid = SIMView.FID_EF_FDN`
	`fciOffset = 20`
	`fciLength = 15`
	`select()`
		Shall throw java.lang.ArrayIndexOutOfBounds Exception.	

13	**Test selection possibilities**
	`1 - fid = SIMView.FID_MF`
	`fciOffset = 20`
	`fciLength = 15`
	`select()`
	`2 - fid = SIMView.FID_MF`
	`select()`
	`3.1 – fid = SIMView.FID_DF_TELECOM`
	`select()`
	`3.2 – fid = SIMView.FID_DF_GSM`
	`select()`
		1 – No exception shall be thrown.
		2 – No exception shall be thrown.
		3 – No exception shall be thrown.
		4 – No exception shall be thrown.
		5 – No exception shall be thrown.	

Id	Description	API Expectation	APDU Expectation
	4 – fid = SIMView.FID_DF_GSM select() 5 - fid = SIMView.FID_MF select()		
14	**Test if EF selected after MF/DF selection** fid = SIMView.FID_MF select() readBinary	Shall throw sim.access.SIMViewException with reason code NO_EF_SELECTED.	
15	**Test of "missing" file** 1 - fid = SIMView.FID_MF select() 2 - fid = SIMView.FID_EF_ACM select()	1 – No exception shall be thrown. 2 – Shall throw sim.access.SIMViewException with reason code FILE_NOT_FOUND.	
16	**Test that no record is selected after selecting linear fixed EF** 1 - fid = SIMView.FID_MF select() 2 - fid = SIMView.FID_DF_TELECOM select() 3 - fid = SIMView.FID_EF_ADN select() 4 – recNumber = 0 mode = REC_ACC_MODE_ABSOLUTE_CURRENT readRecord()	1 – No exception shall be thrown. 2 – No exception shall be thrown. 3 – No exception shall be thrown. 4 – Shall throw sim.access.SIMViewException with reason code RECORD_NUMBER_NOT_AVAILABLE.	
17	**Test record pointer in selected cyclic EF** *<To Be Determined> (CRRN6)*		
18	**Test of lack of change to file context** *<To Be Determined> (CRRN7)*		

6.2.1.2.5 Test Coverage

CRR Number	*Test Case Number*
N1	***1-7***
N2	***1, 3-5***
N3	***2, 6, 7***
N4	***14***
N5	***16***
N6	***17***
N7	***18***
N8	***1, 3***
N9	***1-7, 13***
P1	***8***
P2	***9***
P3	***10***
P4	***11***
P5	***12***
C1	***15***
C2, C3	***Not Tested***

6.2.1.3 Method select

6.2.1.3.1 Test Area Reference: API_1_SVW_SLCTS

6.2.1.3.2 Conformance Requirements

The method with the following header shall be compliant to its definition in the API.

```
public void select(short fid)
```

```
                throws SIMViewException
```

Normal Execution

CRRN1: If the desired file is selected, no exception is thrown.

CRRN2: After selecting a DF/MF no EF is selected.

CRRN3: After selecting a linear fixed EF no record is selected.

CRRN4: After selecting a cyclic EF the first record which is the last updated record is selected.

CRRN5: The current files (file context) of any other applets shall not be changed. [03.19 - §5.2]

CRRN6: The file with a File-ID that matches fid shall be found according to the following selection rules:

- An immediate child EF or DF of the current MF/DF can be selected,
- A sibling DF of the current DF can be selected,
- The current MF/DF it self can be selected,
- The parent MF/DF of the current DF can be selected,
- The MF can always be selected.

Parameter Errors

No requirements

Context Errors

CRRC1: If the file with a File-ID which matches fid could not be found according to the selection rules

An immediate child EF or DF of the current MF/DF can be selected,

A sibling DF of the current DF can be selected,

The current MF/DF it self can be selected,

The parent MF/DF of the current DF can be selected,

The MF can always be selected

an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.FILE_NOT_FOUND.

CRRC2: If the method call causes a memory problem (e.g. memory access error), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.MEMORY_PROBLEM.

CRRC3: If the method call causes an error to occur that is not expected and thus not handled, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INTERNAL_ERROR.

6.2.1.3.3 Test Suite Files

Additional requirements for the GSM personalisation:

...

- Test Script: API_1_SVW_SLCTS_1.scr
- Test Applet: API_1_SVW_SLCTS_1.java
- Installation Parameter: API_1_SVW_SLCTS.install
- Load Script: API_1_SVW_SLCTS.ldr
- Conversion parameter: API_1_SVW_SLCTS.cnv

6.2.1.3.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	**Select EF_{ICCID} in MF (Transparent EF)**
	`fid = SIMView.FID_EF_ICCID`
	`select()`
		No exception shall be thrown.	

2	**Test if EF selected after MF/DF selection**
	`fid = SIMView.FID_DF_TELECOM`
	`select()`
	`readBinary`
		Shall throw sim.access.SIMViewException with reason code NO_EF_SELECTED.	

3	**Test that no record is selected after selecting linear fixed EF**
	`1 - fid = SIMView.FID_MF`
	`select()`
	`2 - fid = SIMView.FID_DF_TELECOM`
	`select()`
	`3 - fid = SIMView.FID_EF_ADN`
	`select()`
	`4 – recNumber = 0`
	`mode = REC_ACC_MODE_ABSOLUTE_CURRENT`
	`readRecord()`
		1 – No exception shall be thrown.
		2 – No exception shall be thrown.
		3 – No exception shall be thrown.
		4 – Shall throw sim.access.SIMViewException with reason code RECORD_NUMBER_NOT_AVAILABLE.	

4	**Test record pointer in selected cyclic EF** *<To Be Determined> (CRRN4)*		
5	**Test of lack of change to file context** *<To Be Determined> (CRRN7)*		

6	**Test selection possibilities**
	`1 - fid = SIMView.FID_MF`
	`select()`
	`2 - fid = SIMView.FID_MF`
	`select()`
	`3.1 – fid = SIMView.FID_DF_TELECOM`
	`select()`
	`3.2 – fid = SIMView.FID_DF_GSM`
	`select()`
	`4 – fid = SIMView.FID_DF_GSM`
	`select()`
	`5 - fid = SIMView.FID_MF`
	`select()`
		1 – No exception shall be thrown.
		2 – No exception shall be thrown.
		3 – No exception shall be thrown.
		4 – No exception shall be thrown.
		5 – No exception shall be thrown.	

7	**Test of "missing" file**
	`1 - fid = SIMView.FID_MF`
	`select()`
	`2 - fid = SIMView.FID_EF_ACM`
	`select()`
		1 – No exception shall be thrown.
		2 – Shall throw sim.access.SIMViewException with reason code FILE_NOT_FOUND.	

6.2.1.3.5 Test Coverage

CR Number	*Test Case Number*
CRRN1	***1***
CRRN2	***2***
CRRN3	***3***
CRRN4	***4***
CRRN5	***5***
CRRN6	***6***
CRRC1	***7***
CRRC2, CRRC3	***Not Tested***

6.2.1.4 Method status

6.2.1.4.1 Test Area Reference: API_1_SVW_STAT_BSS

6.2.1.4.2 Conformance Requirements

The method with the following header shall be compliant to its definition in the API.

```
public short status(byte[] fci,
                    short fciOffset,
                    short fciLength)
             throws java.lang.NullPointerException,
                    java.lang.ArrayIndexOutOfBoundsException,
                    SIMViewException
```

Normal Execution

CRRN1: The FCI (File Control Information) of the current DF (or MF) is returned in the same format as for a SELECT command in case of selecting an MF/DF (see command above).

CRRN2: If the length fciLength is greater than or equal to the length of the response, the whole response is copied into the array fci and the length of the FCI which has been written to the array fci is returned.

CRRN3: If the length fciLength is less than the length of the response, the first part of the response is copied into the array fci and the length of the FCI which has been written to the array fci is returned.

Parameter Errors

CRRP1: If the array fci is null, an instance of NullPointerException shall be thrown.

CRRP2: If fciLength is greater than fci.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: If fciOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP4: If fciLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP5: If fciOffset plus fciLength is greater than the length of the array fci.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

Context Errors

CRRC1: If the method call causes a memory problem (e.g. memory access error), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.MEMORY_PROBLEM.

CRRC2: If the method call causes an error to occur that is not expected and thus not handled, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INTERNAL_ERROR.

6.2.1.4.3 Test Suite Files

Additional requirements for the GSM personalisation:

...

- Test Script: API_1_SVW_STAT_BSS_1.scr
- Test Applet: API_1_SVW_STAT_BSS_1.java
- Installation Parameter: API_1_SVW_STAT_BSS.install
- Load Script: API_1_SVW_STAT_BSS.ldr
- Conversion parameter: API_1_SVW_STAT_BSS.cnv

6.2.1.4.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

Id	Description	API Expectation	APDU Expectation
1	**SIM Initialisation**	Responses ignored.	

2	**Status of MF**
	`byte[] fci = new byte[34]`
	`fciOffset = 0`
	`fciLength = 34`
	`status()`
		No exception shall be thrown.
		Shall return 34.
		fci shall contain the entire response.
		<*Description of fci*>	

3	**Status after select EF_{ICCID} in MF**
	`1 – fid = SIMView.FID_EF_ICCID`
	`select()`
	`2 – byte[] fci2 = new byte[34]`
	`fciOffset = 0`
	`fciLength = 34`
	`status()`
	`3 – Compare the 34 bytes of fci and fci2`
		1 - No exception shall be thrown.
		Shall return 14 or more (Transparent EF).
		2 - No exception shall be thrown.
		Shall return 34.
		fci2 shall contain the entire response.
		<*Description of fci2*>
		fci and fci2 should be identical	

4	**Status of $DF_{Telecom}$**
	`1 – fid = SIMView.FID_DF_TELECOM`
	`select()`
	`2 – fciOffset = 0`
	`fciLength = 35`
	`status()`
		1 - No exception shall be thrown.
		Shall return 34.
		<*Description of fci*>
		2 - No exception shall be thrown.
		Shall return 34.
		fci shall contain the entire response.
		<*Description of fci*>	

5	**Status $DF_{TELECOM}$**
	`fciOffset = 0`
	`fciLength = 20`
	`status()`
		No exception shall be thrown.
		Shall return 20.
		fci shall contain the first 20 bytes of the response starting at index 0.
		<*Description of fci*>	

6	**fci is null**
	`byte[] fci = null`
	`fciOffset = 0`
	`fciLength = 34`
	`status()`
		Shall throw java.lang.NullPointerException.	

7	**response + fciOffset > fci.length**
	`byte[] fci = new byte[34]`
	`fciOffset = 1`
	`fciLength = 34`
	`status()`
		Shall throw java.lang.ArrayIndexOutOfBounds Exception.	

8	**fciOffset < 0**
	`fciOffset = -1`
	`fciLength = 34`
	`status()`
		Shall throw java.lang.ArrayIndexOutOfBounds Exception.	

9	**fciLength < 0**
	`fciOffset = 0`
	`fciLength = -1`
	`status()`
		Shall throw java.lang.ArrayIndexOutOfBounds Exception.	

10	**fciOffset + fciLength > fci.length**
	`fciOffset = 20`
	`fciLength = 15`
	`status()`
		Shall throw java.lang.ArrayIndexOutOfBounds Exception.	

6.2.1.4.5 Test Coverage

CR Number	*Test Case Number*
CRRN1	***1-5***
CRRN2	***1-4***
CRRN3	***5***
CRRP1	***6***
CRRP2	***7***
CRRP3	***8***
CRRP4	***9***
CRRP5	***10***

CR Number	Test Case Number
CRRC1, CRRC2	Not Tested

6.2.1.5 Method readBinary

6.2.1.5.1 Test Area Reference: API_1_SVW_REDBS_BSS

6.2.1.5.2 Conformance Requirements

The method with the following header shall be compliant to its definition in the API.

```
public short readBinary(short fileOffset,
                        byte[] resp,
                        short respOffset,
                        short respLength)
                 throws java.lang.NullPointerException,
                        java.lang.ArrayIndexOutOfBoundsException,
                        SIMViewException
```

Normal Execution

CRRN1: If data can be accessed at the specified offset, the value respOffset plus respLength are returned and the data bytes of the currently selected transparent file are returned in resp.

Parameter Errors

CRRP1: If fileOffset is less than 0, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.OUT_OF_FILE_BOUNDARIES.

CRRP2: If fileOffset plus respLength exceeds the length of the file, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.OUT_OF_FILE_BOUNDARIES.

CRRP3: If the array resp is null, an instance of NullPointerException shall be thrown.

CRRP4: If respOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP5: If respLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP6: If respOffset plus respLength is greater than the length of the array resp.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

Context Errors

CRRC1: If the calling applet has currently no EF selected, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.NO_EF_SELECTED.

CRRC2: If the currently selected EF is not transparent, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.FILE_INCONSISTENT.

CRRC3: If the calling applet does not fulfil the access condition, READ, to perform this function, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.AC_NOT_FULFILLED.

CRRC4: If the currently selected EF is invalidated and the file status of the EF does not allow for the reading or updating of an invalidated file, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALIDATION_STATUS_CONTRADICTION.

CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.MEMORY_PROBLEM.

CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INTERNAL_ERROR.

6.2.1.5.3 Test Suite Files

Additional requirements for the GSM personalisation:

The file $EF_{ToBeDetermined1}$ shall have no READ access, so the CRR3 can be tested.

The file $EF_{ToBeDetermined2}$ shall have no READ or UPDATE access, but shall have INVALIDATE access.

- Test Script: API_1_SVW_REDBS_BSS_1.scr
- Test Applet: API_1_SVW_REDBS_BSS_1.java
- Installation Parameter: API_1_SVW_REDBS_BSS.install
- Load Script: API_1_SVW_REDBS_BSS.ldr
- Conversion parameter: API_1_SVW_REDBS_BSS.cnv

6.2.1.5.4 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	**Read from EF_{ICCID} in MF (Transparent EF)**
	`1 – fid = SIMView.FID_EF_ICCID`
	`select()`
	`2 – fileOffset = 0`
	`byte[] resp = new byte[20]`
	`respOffset = 10`
	`respLength = 10`
	`readBinary()`
		1 - No exception shall be thrown.
		2 - No exception shall be thrown. Shall return 20.
		resp shall contain the entire contents of EF_{ICCID} starting at index 10.
		<Description of resp>	

2	**Read from EF_{ICCID} in MF**
	`fileOffset = 5`
	`respOffset = 10`
	`respLength = 5`
	`readBinary()`
		No exception shall be thrown. Shall return 15.
		resp shall contain the last 5 bytes of EF_{ICCID} starting at index 10.
		<Description of resp>	

3	**File out of bounds**
	`fileOffset = -1`
	`respOffset = 0`
	`respLength = 10`
	`readBinary()`
		Shall throw sim.access.SIMViewException with reason code OUT_OF_FILE_BOUNDARDIES.	

4	**fileOffset + respLength > EF length**
	`fileOffset = 9`
	`respOffset = 0`
	`respLength = 2`
	`readBinary()`
		Shall throw sim.access.SIMViewException with reason code OUT_OF_FILE_BOUNDARDIES.	

5	**resp is null**
	`byte[] resp = null`
	`fileOffset = 0`
	`respOffset = 0`
	`respLength = 10`
	`readBinary()`
		Shall throw java.lang.NullPointerException.	

6	**respOffset < 0**
	`fileOffset = 0`
	`resp = new byte[20]`
	`respOffset = -1`
	`respLength = 10`
	`readBinary()`
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

7	**respLength < 0**
	`fileOffset = 0`
	`respOffset = 0`
	`respLength = -1`
	`readBinary()`
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

8	**respOffset + respLength > resp.length**
	`fileOffset = 0`
	`respOffset = 10`
	`respLength = 11`
	`readBinary()`
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

Id	Description	API Expectation	APDU Expectation
9	*<To Be Determined>*		
10	*<To Be Determined>*		
11	*<To Be Determined>*		
12	**No EF selected** 1- fid = SIMView.FID_MF select() 2 – fileOffset = 0 respOffset = 0 respLength = 10 readBinary()	1 - No exception shall be thrown. 2 - Shall throw sim.access.SIMViewException with reason code NO_EF_SELECTED.	

6.2.1.5.5 Test Coverage

CRR Number	*Test Case Number*
CRRN1	***1-2***
CRRP1	***3***
CRRP2	***4***
CRRP3	***5***
CRRP4	***6***
CRRP5	***7***
CRRP6	***8***
CRRC1	***12***
CRRC2	***9***
CRRC3	***10***
CRRC4	***11***
CRRC5, CRRC6	***Not Tested***

6.2.1.6 Method updateBinary

6.2.1.6.1 Test Area Reference: API_1_SVW_UPDBS_BSS

6.2.1.6.2 Conformance Requirements

The method with the following header shall be compliant to its definition in the API.

```
public short updateBinary(short fileOffset,
                          byte[] data,
                          short dataOffset,
                          short dataLength)
                   throws java.lang.NullPointerException,
                          java.lang.ArrayIndexOutOfBoundsException,
                          SIMViewException
```

Normal Execution

CRRN1: The currently selected transparent file is updated starting at fileOffset, with the string of dataLength bytes in the array data starting at dataOffset.

Parameter Errors

CRRP1: If fileOffset is less than 0, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.OUT_OF_FILE_BOUNDARIES.

CRRP2: If fileOffset plus dataLength exceeds the length of the file, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.OUT_OF_FILE_BOUNDARIES.

CRRP3: If the array data is null, an instance of NullPointerException shall be thrown.

CRRP4: If dataOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP5: If dataLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP6: If dataOffset plus dataLength greater than the length of the array data.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

Context Errors

CRRC1: If the calling applet has currently no EF selected, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.NO_EF_SELECTED.

CRRC2: If the currently selected EF is not transparent, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.FILE_INCONSISTENT.

CRRC3: If the calling applet does not fulfil the access condition, UPDATE, to perform this function, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.AC_NOT_FULFILLED.

CRRC4: If the currently selected EF is invalidated and the file status of the EF does not allow for reading or updating of an invalidated file, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALIDATION_STATUS_CONTRADICTION.

CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.MEMORY_PROBLEM.

CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INTERNAL_ERROR.

6.2.1.6.3 Test Suite Files

Additional requirements for the GSM personalisation:

...

- Test Script: API_1_SVW_UPDBS_BSS_1.scr
- Test Applet: API_1_SVW_UPDBS_BSS_1.java
- Installation Parameter: API_1_SVW_UPDBS_BSS.install
- Load Script: API_1_SVW_UPDBS_BSS.ldr
- Conversion parameter: API_1_SVW_UPDBS_BSS.cnv

6.2.1.6.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			
2			
3			
4			
5			

6.2.1.6.5 Test Coverage

CR Number	*Test Case Number*
CRRN1	
CRRP1	
CRRP2	
CRRP3	
CRRP4	
CRRP5	
CRRP6	
CRRC1	
CRRC2	
CRRC3	
CRRC4	

CR Number	Test Case Number
CRRC5, CRRC6	Not Tested

6.2.1.7 Method readRecord

6.2.1.7.1 Test Area Reference: API_1_SVW_REDRSBS_BSS

6.2.1.7.2 Conformance Requirements

The method with the following header shall be compliant to its definition in the API.

```
public short readRecord(short recNumber,
                        byte mode,
                        short recOffset,
                        byte[] resp,
                        short respOffset,
                        short respLength)
                 throws java.lang.NullPointerException,
                        java.lang.ArrayIndexOutOfBoundsException,
                        SIMViewException
```

Normal Execution

CRRN1: The data bytes from the record, specified by mode and recNumber of the currently selected linear fixed or cyclic EF, is read at recOffset. A total of respLength bytes of this data is copied to the array resp at respOffset.

CRRN2: If the access mode is REC_ACC_MODE_ABSOLUTE_CURRENT:

- if recNumber is not 0, the record addressed by recNumber will be read;
- if recNumber is 0 the current selected record will be read; and
- the current record pointer shall not change.

CRRN3: If the access mode is REC_ACC_MODE_NEXT:

- the next record relative to the current selected record will be selected and read;
- if no current record is selected, the first record will be selected and read;
- if the current record pointer is set to the last record for a cyclic EF the record pointer is set to the first record and the record is read;
- the current record pointer of any other applet shall not be changed.

CRRN4: If the access mode is REC_ACC_MODE_PREVIOUS:

- the previous record relative to the current selected record will be selected and read;
- if no current record is selected, the last record will be selected and read;
- if the current record pointer is set to the first record, for a linear fixed EF the method responses with an error exception and for a cyclic EF the record pointer is set to the last record and the record is read;
- the current record pointer of any other applet shall not be changed.

Parameter Errors

CRRP1: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_ABSOLUTE_CURRENT and recNumber is less than 0 or greater than records available, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.RECORD_NUMBER_NOT_AVAILABLE.

CRRP2: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_ABSOLUTE_CURRENT, recNumber is 0 and there is no current record selected, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.RECORD_NUMBER_NOT_AVAILABLE.

CRRP3: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_NEXT and the current record pointer is set to the last record, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.RECORD_NUMBER_NOT_AVAILABLE.

CRRP4: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_PREVIOUS and the current record pointer is set to the first record, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.RECORD_NUMBER_NOT_AVAILABLE.

CRRP5: If the specified offset into the selected record recOffset is less than 0, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.OUT_OF_RECORD_BOUNDARIES.

CRRP6: If recOffset plus+ respLength is greater than the record lengh, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.OUT_OF_RECORD_BOUNDARIES.

CRRP7: If the access mode is not between 2 and 4 inclusive (2 = REC_ACC_MODE_NEXT, etc.), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALID_MODE.

CRRP8: If the access mode is other than REC_ACC_MODE_ABSOLUTE_CURRENT and recNumber is not 0 an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALID_MODE.

CRRP9: If the array resp is null, an instance of NullPointerException shall be thrown.

CRRP10: If respOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP11: If respLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP12: If respOffset plus respLength is greater than the length of the array resp.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

Context Errors

CRRC1: If the calling applet has currently no EF selected, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.NO_EF_SELECTED.

CRRC2: If the currently selected EF is neither linear fixed nor cyclic, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.FILE_INCONSISTENT.

CRRC3: If the calling applet does not fulfil the access condition, READ, to perform this function, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.AC_NOT_FULFILLED.

CRRC4: If the currently selected EF is invalidated and the file status of the EF does not allow for reading / updating an invalidated file, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALIDATION_STATUS_CONTRADICTION.

CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.MEMORY_PROBLEM.

CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INTERNAL_ERROR.

6.2.1.7.3 Test Suite Files

Additional requirements for the GSM personalisation:

...

- Test Script: API_1_SVW_REDRSBS_BSS_1.scr
- Test Applet: API_1_SVW_REDRSBS_BSS_1.java
- Installation Parameter: API_1_SVW_REDRSBS_BSS.install
- Load Script: API_1_SVW_REDRSBS_BSS.ldr
- Conversion parameter: API_1_SVW_REDRSBS_BSS.cnv

6.2.1.7.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			
2			
3			
4			
5			

6.2.1.7.5 Test Coverage

CR Number	*Test Case Number*
CRRN1	
CRRN2	
CRRN3	
CRRN4	
CRRP1	
CRRP2	
CRRP3	
CRRP4	
CRRP5	
CRRP6	
CRRP7	
CRRP8	
CRRP9	
CRRP10	
CRRP11	
CRRP12	
CRRC1	
CRRC2	
CRRC3	
CRRC4	
CRRC5, CRRC6	***Not Tested***

6.2.1.8 Method updateRecord

6.2.1.8.1 Test Area Reference: API_1_SVW_UPDRSBS_BSS

6.2.1.8.2 Conformance Requirements

The method with the following header shall be compliant to its definition in the API.

```
public short updateRecord(short recNumber,
                          byte mode,
                          short recOffset,
                          byte[] data,
                          short dataOffset,
                          short dataLength)
                   throws java.lang.NullPointerException,
                          java.lang.ArrayIndexOutOfBoundsException,
                          SIMViewException
```

Normal Execution

CRRN1: dataLength bytes of the record specified by mode and recNumber of the current selected linear fixed or cyclic EF are updated at recOffset, by using the string of bytes in the array data starting at dataOffset.

CRRN2: If the access mode is REC_ACC_MODE_ABSOLUTE_CURRENT and the file is a linear fixed EF:

- the record addressed by recNumber will be updated;

- if recNumber is 0 the current selected record will be updated; and
- the current record pointer shall not change.

CRRN3: If the access mode is REC_ACC_MODE_NEXT and the file is a linear fixed EF:

- the next record relative to the current selected record will be selected and updated;
- if no current record is selected, the first record will be selected and updated;
- the current record pointer of any other applet shall not be changed.

CRRN4: If the access mode is REC_ACC_MODE_PREVIOUS:

- the previous record relative to the current selected record will be selected and updated;
- if no current record is selected, the last record will be selected and updated;
- if a cyclic EF is updated, the oldest record will be updated independent of the current record pointer and this record becomes record number 1 and the current record;
- the current record pointer of any other applet shall not be changed in case of a linear fixed EF.

Parameter Errors

CRRP1: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_ABSOLUTE_CURRENT and recNumber is less than 0 or greater than records available, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.RECORD_NUMBER_NOT_AVAILABLE.

CRRP2: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_ABSOLUTE_CURRENT, recNumber is 0 and there is no current record selected, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.RECORD_NUMBER_NOT_AVAILABLE.

CRRP3: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_NEXT and the current record pointer is set to the last record, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.RECORD_NUMBER_NOT_AVAILABLE.

CRRP4: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_PREVIOUS and the current record pointer is set to the first record; an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.RECORD_NUMBER_NOT_AVAILABLE.

CRRP5: If the specified offset into the selected record recOffset is less than 0, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.OUT_OF_RECORD_BOUNDARIES.

CRRP6: If recOffset plus dataLength is greater than the record lengh, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.OUT_OF_RECORD_BOUNDARIES.

CRRP7: If the access mode is not between 2 and 4 inclusive (2 = REC_ACC_MODE_NEXT, etc.), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALID_MODE.

CRRP8: If the currently selected EF is cyclic and the mode of record access mode is not REC_ACC_MODE_PREVIOUS, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALID_MODE.

CRRP9: If the access mode is not REC_ACC_MODE_ABSOLUTE_CURRENT and recNumber is not 0 an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALID_MODE.

CRRP10: If the array data is null, an instance of NullPointerException shall be thrown.

CRRP11: If dataOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP12: If dataLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP13: If dataOffset plus dataLength, is greater than the length of the array data.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

Context Errors

CRRC1: If the calling applet has currently no EF selected, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.NO_EF_SELECTED.

CRRC2: If the currently selected EF is neither linear fixed nor cyclic, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.FILE_INCONSISTENT.

CRRC3: If the calling applet does not fulfil the access condition, UPDATE, to perform this function, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.AC_NOT_FULFILLED.

CRRC4: If the currently selected EF is invalidated and the file status of the EF does not allow for reading / updating an invalidated file, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALIDATION_STATUS_CONTRADICTION.

CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.MEMORY_PROBLEM.

CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INTERNAL_ERROR.

6.2.1.8.3 Test Suite Files

Additional requirements for the GSM personalisation:

...

- Test Script: API_1_SVW_UPDRSBS_BSS_1.scr
- Test Applet: API_1_SVW_UPDRSBS_BSS_1.java
- Installation Parameter: API_1_SVW_UPDRSBS_BSS.install
- Load Script: API_1_SVW_UPDRSBS_BSS.ldr
- Conversion parameter: API_1_SVW_UPDRSBS_BSS.cnv

6.2.1.8.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			
2			
3			
4			
5			

6.2.1.8.5 Test Coverage

CR Number	*Test Case Number*
CRRN1	
CRRN2	
CRRN3	
CRRN4	
CRRP1	
CRRP2	
CRRP3	
CRRP4	
CRRP5	
CRRP6	
CRRP7	
CRRP8	
CRRP9	

CR Number	Test Case Number
CRRP10	
CRRP11	
CRRP12	
CRRP13	
CRRC1	
CRRC2	
CRRC3	
CRRC4	
CRRC5, CRRC6	**Not Tested**

6.2.1.9 Method seek

6.2.1.9.1 Test Area Reference: API_1_SVW_SEEKB_BSS

6.2.1.9.2 Conformance Requirements

The method with the following header shall be compliant to its definition in the API.

```
public short seek(byte mode,
                  byte[] patt,
                  short pattOffset,
                  short pattLength)
           throws java.lang.NullPointerException,
                  java.lang.ArrayIndexOutOfBoundsException,
                  SIMViewException
```

Normal Execution

CRRN1: If the pattern in patt with the length pattLength at offset pattOffset is found in the record being specified by mode, the current record pointer is set to that record and the record number is returned. The record pointer of any other applet is not changed.

CRRN2: If mode is SEEK_FROM_BEGINNING_FORWARD, the search starts with the first record forward towards the end of the file.

CRRN3: If mode is SEEK_FROM_END_BACKWARD, the search starts with the last record backward towards the beginning of the file.

CRRN4: If mode is SEEK_FROM_NEXT_FORWARD, the search starts from the next record after the current record pointer forward towards the end of file. If no current record pointer is selected, the search starts with the first record.

CRRN5: If mode is SEEK_FROM_PREVIOUS_BACKWARD, the search starts from the previous record before the current record pointer backward towards the beginning of the file. If no current record pointer is selected the search starts with the last record.

CRRN6: If pattern in patt is not found in record, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.PATTERN_NOT_FOUND.

CRRN7: If mode is SEEK_FROM_NEXT_FORWARD and the record pointer is at the last record, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.PATTERN_NOT_FOUND.

CRRN8: If mode is SEEK_FROM_PREVIOUS_BACKWARD and the record pointer is at the first record, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.PATTERN_NOT_FOUND.

Parameter Errors

CRRP1: If mode is not between 0 and 3 inclusive (0 = SEEK_FROM_BEGINNING_FORWARD, etc.), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALID_MODE.

CRRP2: If the pattern array patt is null, an instance of NullPointerException shall be thrown.

CRRP3: If pattOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP4: If pattLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP5: If pattLength is not between 1 and 16 inclusive, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.OUT_OF_RECORD_BOUNDARIES

CRRP6: If pattLength is greater than the size of the record of the currently selected EF, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.OUT_OF_RECORD_BOUNDARIES.

CRRP7: If pattLength plus pattLength, is greater than the length of the pattern array patt.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

Context Errors

CRRC1: If the calling applet has currently no EF selected, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.NO_EF_SELECTED.

CRRC2: If the currently selected EF is not linear fixed, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.FILE_INCONSISTENT.

CRRC3: If the calling applet does not fulfil the access condition, READ, to perform this function, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.AC_NOT_FULFILLED.

CRRC4: If the currently selected EF is invalidated and the file status of the EF does not allow for reading / updating an invalidated file, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALIDATION_STATUS_CONTRADICTION.

CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.MEMORY_PROBLEM.

CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INTERNAL_ERROR.

6.2.1.9.3 Test Suite Files

Additional requirements for the GSM personalisation:

...

- Test Script: API_1_SVW_SEEKB_BSS_1.scr
- Test Applet: API_1_SVW_SEEKB_BSS_1.java
- Installation Parameter: API_1_SVW_SEEKB_BSS.install
- Load Script: API_1_SVW_SEEKB_BSS.ldr
- Conversion parameter: API_1_SVW_SEEKB_BSS.cnv

6.2.1.9.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			
2			
3			
4			
5			

6.2.1.9.5 Test Coverage

CR Number	*Test Case Number*
CRRN1	
CRRN2	
CRRN3	

CR Number	Test Case Number
CRRN4	
CRRN5	
CRRN6	
CRRN7	
CRRN8	
CRRP1	
CRRP2	
CRRP3	
CRRP4	
CRRP5	
CRRP6	
CRRP7	
CRRC1	
CRRC2	
CRRC3	
CRRC4	
CRRC5, CRRC6	***Not Tested***

6.2.1.10 Method increase

6.2.1.10.1 Test Area Reference: API_1_SVW_INCR_BS_BS

6.2.1.10.2 Conformance Requirements

The method with the following header shall be compliant to its definition in the API.

```
public short increase(byte[] incr,
                      short incrOffset,
                      byte[] resp,
                      short respOffset)
               throws java.lang.NullPointerException,
                      java.lang.ArrayIndexOutOfBoundsException,
                      SIMViewException
```

Normal Execution

CRRN1: The value in the array incr is added to the value of the last increased / updated record in the currently selected cyclic EF. The result is stored in the oldest record and returned in the array resp. The updated record becomes record number 1 and is selected as current record.

Parameter Errors

CRRP2: If the array incr is null, an instance of NullPointerException shall be thrown.

CRRP3: If incrOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP4: If incrOffset plus the value 3, is greater than the length of the array incr.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP5: If the result of the addition is greater than the maximum value of the record (represented by all bytes set to 0xFF), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.MAX_VALUE_REACHED.

CRRP6: If the array resp is null, an instance of NullPointerException shall be thrown.

CRRP7: If respOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

CRRP8: If the remaining length of the array resp at the offset respOffset is less than the length of the record, an instance of ArrayIndexOutOfBoundsException shall be thrown.

Context Errors

CRRC1: If the calling applet has currently no EF selected, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.NO_EF_SELECTED.

CRRC2: If the currently selected EF is not cyclic, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.FILE_INCONSISTENT.

CRRC3: If increase is not allowed as indicated by the FCI byte 8 (GSM 11.11: FCI structure of an EF returned by the SELECT command), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.FILE_INCONSISTENT.

CRRC4: If the calling applet does not fulfil the access condition, INCREASE, to perform this function, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.AC_NOT_FULFILLED.

CRRC5: If the currently selected EF is invalidated, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALIDATION_STATUS_CONTRADICTION.

CRRC6: If the method call causes a memory problem (e.g. memory access error), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.MEMORY_PROBLEM.

CRRC7: If the method call causes an error to occur that is not expected and thus not handled, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INTERNAL_ERROR.

6.2.1.10.3 Test Suite Files

Additional requirements for the GSM personalisation:

...

- Test Script: API_1_SVW_INCR_BS_BS_1.scr
- Test Applet: API_1_SVW_INCR_BS_BS_1.java
- Installation Parameter: API_1_SVW_INCR_BS_BS.install
- Load Script: API_1_SVW_INCR_BS_BS.ldr
- Conversion parameter: API_1_SVW_INCR_BS_BS.cnv

6.2.1.10.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			
2			
3			
4			
5			

6.2.1.10.5 Test Coverage

CR Number	*Test Case Number*
CRRN1	
CRRP1	
CRRP2	
CRRP3	
CRRP4	
CRRP5	
CRRP6	
CRRP7	
CRRP8	
CRRC1	
CRRC2	

CR Number	Test Case Number
CRRC3	
CRRC4	
CRRC5	
CRRC6, CRRC7	***Not Tested***

6.2.1.11 Method invalidate

6.2.1.11.1 Test Area Reference: API_1_SVW_INVL

6.2.1.11.2 Conformance Requirements

The method with the following header shall be compliant to its definition in the API.

```
public void invalidate()
                throws SIMViewException
```

Normal Execution

CRRN1: The currently selected EF of the calling applet shall be invalidated, i.e. the flag in the EF file status shall be changed accordingly.

Parameter Errors

No requirements

Context Errors

CRRC1: If the calling applet has currently no EF selected, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.NO_EF_SELECTED.

CRRC2: If the calling applet does not fulfil the access condition, INVALIDATE, to perform this function, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.AC_NOT_FULFILLED.

CRRC3: If the currently selected EF is already invalidated, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALIDATION_STATUS_CONTRADICTION.

CRRC4: If the method call causes a memory problem (e.g. memory access error), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.MEMORY_PROBLEM.

CRRC5: If the method call causes an error to occur that is not expected and thus not handled, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INTERNAL_ERROR.

6.2.1.11.3 Test Suite Files

Additional requirements for the GSM personalisation:

...

- Test Script: API_1_SVW_INVL_1.scr
- Test Applet: API_1_SVW_INVL_1.java
- Installation Parameter: API_1_SVW_INVL.install
- Load Script: API_1_SVW_INVL.ldr
- Conversion parameter: API_1_SVW_INVL.cnv

6.2.1.11.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			
2			

Id	Description	API Expectation	APDU Expectation
3			
4			
5			

6.2.1.11.5 Test Coverage

CR Number	*Test Case Number*
CRRN1	
CRRC1	
CRRC2	
CRRC3	
CRRC4, CRRC5	***Not Tested***

6.2.1.12 Method rehabilitate

6.2.1.12.1 Test Area Reference: API_1_SVW_REHA

6.2.1.12.2 Conformance Requirements

The method with the following header shall be compliant to its definition in the API.

```
public void rehabilitate()
                  throws SIMViewException
```

Normal Execution

CRRN1: The currently selected EF of the calling applet shall be rehabilitated, i.e. the flag in the EF file status shall be changed accordingly.

Parameter Errors

No requirements

Context Errors

CRRC1: If the calling applet has currently no EF selected, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.NO_EF_SELECTED.

CRRC2: If the calling applet does not fulfil the access condition, REHABILITATE, to perform this function, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.AC_NOT_FULFILLED.

CRRC3: If the currently selected EF is not invalidated, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALIDATION_STATUS_CONTRADICTION.

CRRC4: If the applet is not allowed to rehabilitate the EF, because the EF is the EF_{IMSI} or EF_{LOCI} and BDN is enabled, and the PROFILE DOWNLOAD procedure indicating that the ME supports the “Call control by SIM” facility is not yet performed, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INVALIDATION_STATUS_CONTRADICTION.

CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.MEMORY_PROBLEM.

CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.INTERNAL_ERROR.

6.2.1.12.3 Test Suite Files

Additional requirements for the GSM personalisation:

...

- Test Script: API_1_SVW_REHA_0.scr

- Test Applet: API_1_SVW_REHA_0.java
- Installation Parameter: API_1_SVW_REHA.install
- Load Script: API_1_SVW_REHA.ldr
- Conversion parameter: API_1_SVW_REHA.cnv

6.2.1.12.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1			
2			
3			
4			
5			

6.2.1.12.5 Test Coverage

CR Number	*Test Case Number*
CRRN1	
CRRC1	
CRRC2	
CRRC3	
CRRC4	
CRRC5, CRRC6	***Not Tested***

6.2.1.13 Test of File System

6.2.1.13.1 Test Area Reference: API_1_SVW_FS

6.2.1.13.2 Conformance Requirements

This test area tests the existence and correct layout of the mandatory File System as described in GSM 11.11 [].

Does there need to be an area for this?

Should this area also test the default pre-personalisation in some way?

6.2.2 Class SIMSystem

6.2.2.1 Method getTheSIMView

6.2.2.1.1 Test Area Reference: API_1_SSY_GETS

6.2.2.1.2 Conformance Requirement:

The method with following header shall compliant to its definition in the API.

```
public static SIMView getTheSIMView()
```

Normal Execution

CRRN1: returns a reference to class which implements the SIMView interface

Parameters error

No requirements

Context error

No requirements

6.2.2.1.3 Test suite files:

No additional requirements for the GSM personalisation:

- Test Script: API_1_SSY_GETS.scr
- Test Applet: API_1_SSY_GETS.java
- Installation parameter: API_1_SSY_GETS.install (Same as default applet)
- Load Script: API_1_SSY_GETS.ldr
- Conversion parameter: API_1_SSY_GETS.cnv

6.2.2.1.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Return a reference to the GSM interface**		

6.2.2.1.5 Test Coverage

CR number	***Test case number***
N1	***1***

6.2.3 Class SIMViewException

6.2.3.1 Method throwIt

6.2.3.1.1 Test Area Reference: API_1_SVE_THIT_S

6.2.3.1.2 Conformance Requirement:

The method with following header shall compliant to its definition in the API.

```
public static void throwIt(short reason)
                throws  SIMViewException
```

Normal Execution

CRRN1: Throws the JCRE instance of SIMViewException with the specified reason

CRRN2: extends javacard.framework.CardRuntimeException

Parameters error

No requirements

Context error

No requirements

6.2.3.1.3 Test suite files:

No additional requirements for the GSM personalisation

- Test Script: API_1_SVE_THIT_S.scr
- Test Applet: API_1_SVE_THIT_S .java
- Installation parameter: API_1_SVE_THIT_S.install (Same as default applet)
- Load Script: API_1_SVE_THIT_S.ldr
- Conversion parameter: API_1_SVE_THIT_S.cnv

6.2.3.1.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Throws the JCRE instance of SIMViewException with the specified reason**	`Reason (specified)`	
2	**SIMViewException extends javacard.framework.CardRuntimeException**	`Reason (specified)`	

6.2.3.1.5 Test Coverage

CRR number	***Test case number***
N1	***1***
N2	***2***

6.2.3.2 Constructor

6.2.3.2.1 Test Area Reference: API_1_SVE_COOR_S

6.2.3.2.2 Conformance Requirement:

The method with following header shall compliant to its definition in the API.

```
public SIMViewException(short reason)
                throws  SIMViewException
```

Normal Execution

CRRN1: Construct a SIMViewException with the specified reason

Parameters error

No requirements

Context error

No requirements

6.2.3.2.3 Test suite files:

No additional requirements for the GSM personalisation

- Test Script: API_1_SVE_COOR_S.scr
- Test Applet: API_1_SVE_COOR_S.java
- Installation parameter: API_1_SVE_COOR_S.install (Same as default applet)
- Load Script: API_1_SVE_COOR_S.ldr
- Conversion parameter: API_1_SVE_COOR_S.cnv

6.2.3.2.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**SIMViewException with the specified reason** (The reason shall set with setReason and compare the Exception with getReason)	Reason (specified)	

6.2.3.2.5 Test Coverage

CRR number	*Test case number*
N1	***1***

6.2.3.3 Reason Codes

6.2.3.3.1 Test Area Reference: API_1_SVE_CONS

6.2.3.3.2 Conformance Requirement:

There is no API, only constants. This constants shall compliant to its definition in the API.

Normal Execution

CRRN1: The Constants of the class SIMViewException shall all have the same name and value defined in the GSM03.19

CRRN2: Constructs SIMViewException a Exception with the specified reason

Parameters error

None

Context error

None

6.2.4 Class SIMViewException

6.3 SIM Toolkit Framework:

System Handlers management

- minimum handler availability / status / content (+ response handling)

Applet Triggering :

- for each event test separately:
  - Registration / de registration, and dynamic registration
  - triggering
  - multiple applet triggering
  - limitations
  - busy state
- Exception hiding to the mobile.
- Events combination.

Proactive commands sending

Envelope response sending

Toolkit applets installation

Access control

File system state

7 Annex A: Acronyms

7.1 Annex A.1: Classes Acronyms

Package sim.toolkit

Class	
EditHandler	EDH
EnvelopeHandler	ENH
EnvelopeResponseHandler	ERH
MEProfile	MEP
ProactiveHandler	PAH
ProactiveResponseHandler	PRH
ToolkitRegistery	TKR
ViewHandler	VWH

Exception	
Toolkit Exception	TKE

Interface	
Toolkit Constant	TKC
Toolkit Interface	TKI

Package Sim Access

Interface	SIMView	SVW
Class	SIMViewException	SVE
Class	SIMSystem	SSY

7.2 Annex A.2: Methods Acronyms

7.2.1 ProactiveHandler methods

Method Name	Acronym
GetTheHandler()	GTHD
Init(byte type, byte qualifier, byte dstDevice)	INITBBB
InitDisplayText(byte qualifier, byte dcs, byte[] buffer, short offset, short length)	INDTBB_BSS
InitGetInkey(byte qualifier, byte dcs, byte[] buffer, short offset, short length)	INGKBB_BSS
InitGetInput(byte qualifier, byte dcs, byte[] buffer, short offset, short length, short minRespLength, short maxRespLength)	INGPBB_BSSSS
Byte send()	SEND

7.2.2 ProactiveResponseHandler methods

Method Name	Acronym
Short CopyAdditionalInformation(byte[] dstBuffer, short dstOffset, short dstLength)	CPAI_BSS
Short copyTextString(byte[] dstBuffer, short dstOffset)	CPT_BS
Short getAdditionalInformationLength()	GTIL
Byte getGeneralResult()	GTGR
Byte getItemIdentifier()	GTII
Byte getTextStringCodingScheme()	GTCS
Short getTextStringLength()	GTTL
GetTheHandler()	GTHD

7.2.3 ToolkitRegistry methods

Method Name	Acronym
byte allocateTimer()	ATIM
void changeMenuEntry(byte id, byte [] menuEntry, short offset, short length, byte nextAction, boolean helpSupported, byte iconQualifier, short iconIdentifier)	CMETB_BSSBZBS
void clearEvent(byte event)	CEVTB
void disableMenuEntry(byte id)	DMETB

void enableMenuEntry(byte id)	EMETB
static ToolkitRegistry getEntry()	GETY
short getPollInterval()	GPOL
byte initMenuEntry(byte [] menuEntry, short offset, short length, byte nextAction, boolean helpSupported, byte iconQualifier, short iconIdentifier)	IMET_BSSBZBS
boolean isEventSet(byte event)	IEVSB
void releaseTimer(byte timerIdentifier)	RTIM
void requestPollInterval(short duration)	RPOL
void setEvent(byte id)	SEVTB
void setEventList(byte [] eventList, short offset, short length)	SEVL_BSS

7.2.4 EditHandler methods

Method Name	Acronym
Void appendArray(byte[] buffer, short offset, short length, short dstLength)	APDA
Void appendTLV(byte tag, byte value)	APTLVBB
Void appendTLV(byte tag, byte[] value, short valueOffset, short valueLength)	APTLVB_BSS
Void appendTLV(byte tag, byte value1, byte value2)	APTLVBBB
Void appendTLV(byte tag, byte value1, byte[] value2, short value2Offset, short value2Length)	APTLVBB_BSS
Void clear()	CLR

7.2.5 ViewHandler methods

Method Name	Acronym
byte compareValue(short valueOffset,byte[] compareBuffer,short compareOffset, short compareLength)	CPRVS_BSS

Short copy(byte[] dstBuffer,short dstOffset,short dstLength)	COPY_BSS
Short copyValue(short valueOffset, byte[] dstBuffer,short dstOffset,short dstLength)	CPYVS_BSS
Byte findAndCompareValue(byte tag,byte[] compareBuffer,short compareOffset)	FACRB_BS
Byte findAndCompareValue(byte tag,byte occurence, short valueOffset,byte[] compareBuffer,short compareOffset,short compareLength)	FACRBBS_BSS
Short FindAndCopyValue(byte tag,byte occurence,short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	FACYBBS_BSS
Short findAndCopyValue(byte tag,byte[] dstBuffer,short dstOffset)	FACYBS_BSS
Byte findTLV(byte tag,byte occurrence)	FINDBB
Short getLength()	GLEN
Byte GetValueByte(short valueOffset)	GVBYTS
Short GetValueLength()	GVLEN

7.2.6 sim.access.SIMView Methods / Fields

Method Name	Acronym
static final Constants	CONST
select(short fid, byte[] fci, short fciOffset, short fciLength)	SLCTS_BSS
void select(short fid)	SLCTS
short status(byte[] fci, short fciOffset, short fciLength)	STAT_BSS
short readBinary(short fileOffset, byte[] resp, short respOffset, short respLength)	REDBS_BSS
short updateBinary(short fileOffset, byte[] data, short dataOffset, short dataLength)	UPDBS_BSS
short readRecord(short recNumber, byte mode, short recOffset, byte[] resp, short respOffset, short	REDRSBS_BSS

respLength)	
short updateRecord(short recNumber, byte mode, short recOffset, byte[] data, short dataOffset, short dataLength)	UPDRSBS_BSS
short seek(byte mode, byte[] patt, short pattOffset, short pattLength)	SEEKB_BSS
short increase(byte[] incr, short incrOffset, byte[] resp, short respOffset)	INCR_BS_BS
void invalidate()	INVL
void rehabilitate()	REHA

7.2.7 EnvelopeHandler methods

Method Name	Acronym
byte getEnvelopeTag()	GENT
byte getItemIdentifier()	GIID
short getSecuredDataLength()	GSDL
short getSecuredDataOffset()	GSDO
EnvelopeHandler getTheHandler()	GTHD
short getTPUDLOffset()	GTPO

7.2.8 EnvelopeResponseHandler methods

Method Name	Acronym
EnvelopeResponseHandler getTheHandler()	GTHD
Void post(byte statusType)	POST_B
Void postAsBERTLV(byte statusType, byte tag)	POST_BB

8 Annex B: Script file syntax description

```
CMD : Command to follow
RST : Resets and powers on the card
OFF : Powers off the card
REM : Uses for comments
'\n': Empty lines are accepted
' ', '\t' : Can be used as separator
'\\' : Continues on next line
XXX : each line beginning with 3 characters indicates other tool command.

     XX not to be checked
     [] data to be checked, need to be present for an outgoing command
     () status to be checked
```

```
REM this is an example

RST
REM Case 1 example
CMD A0 C2 00 00 00 (91 33 , 69 XX)

REM Case 2 example
CMD A0 C2 00 00 B0 [XX XX XX 55 55 XX 55](91 33 , 69 XX)
CMD A0 C2 00 00 B0 [] (91 33 , 69 XX)
CMD A0 C2 00 00 B0  \
         [XX XX XX 55  \
         55 XX 55]

REM Case 3 example
CMD A0 C2 00 00 B0 D1 81 AE ... 33 (91 33)

OFF
```

9 Annex C (Informative): Change history

Document history		
0.2.0	Nov 2000	Draft presented at T3#16
1.0.0	Dec 2000	Presented to TSG-T #10 for information