ETSI TS 151 013 V8.0.0 (2014-07)

TECHNICAL SPECIFICATION

Digital cellular telecommunications system (Phase 2+); Test specification for Subscriber Identity Module (SIM) Application Programming Interface (API) for Java Card (3GPP TS 51.013 version 8.0.0 Release 8)

A GLOBAL INITIATIVE

GLOBAL SYSTEM FOR
MOBILE COMMUNICATIONS

Reference

RTS/TSGC-0651013v800

Keywords

GSM

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**may not**", "**need**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Specification (TS) has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present document covers the minimum characteristics considered necessary in order to provide compliance to 3GPP TS 43.019 [7].

The present document describes the technical characteristics and methods of test for testing the SIM API for Java Card^{TM} (3GPP TS 43.019 [7]) implemented in the Subscriber Identity Modules (SIMs) for GSM. It specifies the following parts:

- test applicability;
- test environment description;
- tests format;
- test area reference;
- conformance requirements;
- test auite files;
- test procedure;
- test coverage; and
- a description of the associated testing tools that shall be used.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] Void.

[2] Void.

[3] 3GPP TS 51.011: "Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface (Release 4)".

[4] 3GPP TS 11.14: "Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface (Release 99)".

[5] 3GPP TS 11.17: "Digital cellular telecommunications system (Phase 2+); Subscriber Identity Module (SIM) test specification (Release 99)".

[6] Void.

[7] 3GPP TS 43.019: "Subscriber Identity Module Application Programming Interface (SIM API) for Java Card™; Stage 2 (Release 5)".

[8] 3GPP TS 23.048: "Security Mechanisms for the (U)SIM application toolkit; Stage 2 (Release 5)".

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

[10] 3GPP TS 42.019: "Subscriber Identity Module Application Programming Interface (SIM API); Stage 1 (Release 5)".

[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".

NOTE: SUN Java Card Specifications can be downloaded at http://java.sun.com/products/javacard.

[14] ETSI TS 101 220: "Smart Cards; ETSI numbering system for telecommunication application providers".

[15] 3GPP TS 51.010-1: "Mobile Station (MS) conformance specification; Part 1: Conformance specification (Release 5)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 51.010-1 [15] and the following apply:

applet: application built up using a number of classes which will run under the control of the Java Card virtual machine

applet installation parameters: default values for applet installation parameters

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

CleanUp Script file: file containing the APDU commands that will restore the Default Initial Conditions on the SIM

Conformance Requirement Reference: description of the expected card behaviour according to 3GPP TS 43.019 [7]

expected state: state in which the SIM is supposed to be after the execution of the test procedure applied on the relevant initial conditions

security parameters: minimum security requirements defined for the applet installation process

test area: set of Test Cases applicable to a specific part (class method, framework behaviour, …) of the 3GPP TS 43.019 [7].

test case: elementary test that checks for compliance with one or more Conformance Requirement References

test Output file: TBD.

test procedure: the sequence of actions/commands to perform all the test cases defined in a test area

test script file: file containing the APDU commands that will execute and verify the test results

Test Toolkit Applet: applet designed to test a specific functionality of the SIM API (3GPP TS 43.019 [7])

3.2 Abbreviations

For the purpose of the present document, the abbreviations given in GSM 01.04 [2] and the following apply:

AC	Application Code
AID	Application Identifier
APDU	Application Protocol Data Unit

API	Application Programming Interface
CAD	Card Acceptance Device
CRR	Conformance Requirements Reference
CRRC	Conformance Requirement Reference Context Error
CRRN	Conformance Requirement Reference Normal
CRRP	Conformance Requirement Reference Parameter Error
FFS	For Further Study
IFD	Interface Device
JCRE	Java Card™ Run Time Environment
JVM	Java Virtual Machine
SE	Sending Entity
SIM	Subscriber Identity Module

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

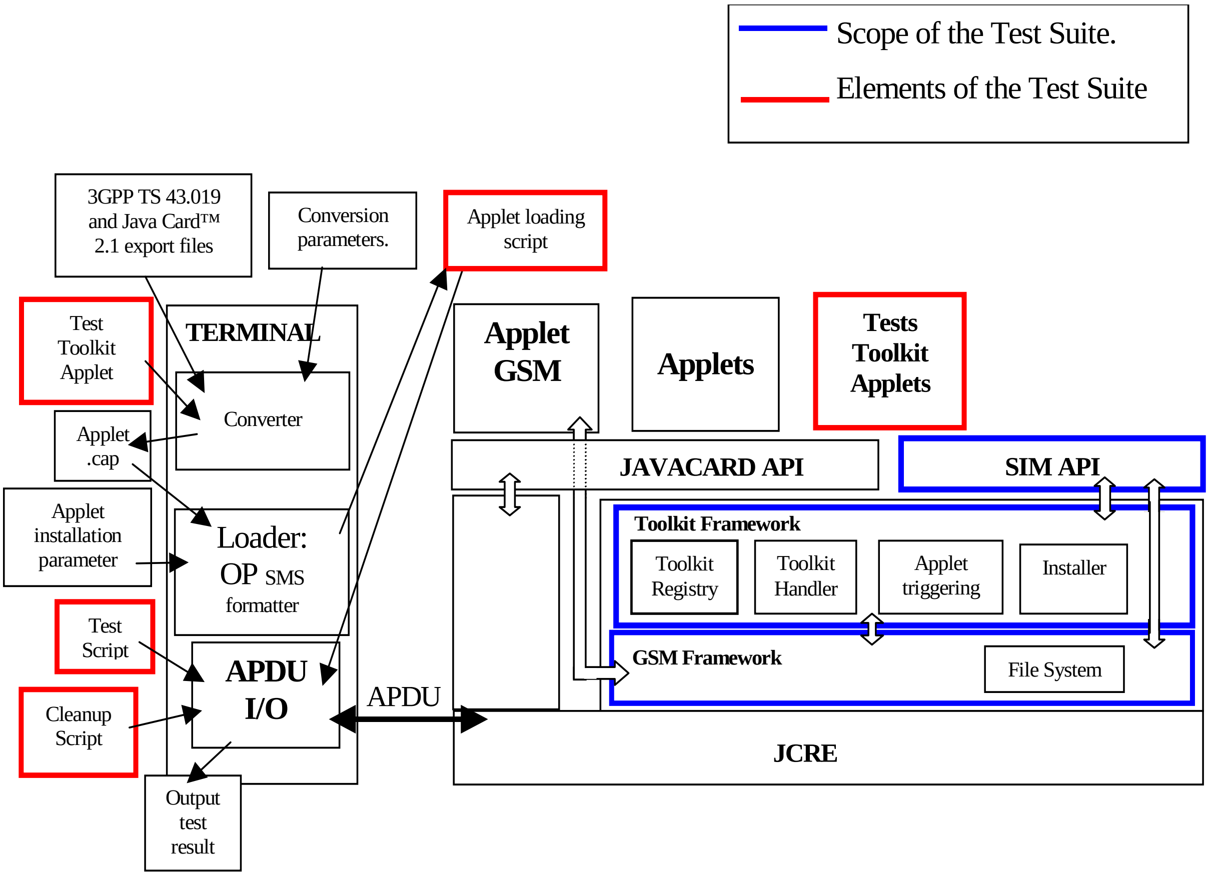
The tests defined in the present document shall be performed taking into account the services supported by the card as specified in the EF_{SST} file.

The test defined in the present document are applicable to cards implementing 3GPP TS 43.019 [7] unless otherwise stated.

The tests defined in the present document require that the card support the concatenation process with 2 concatenated SMS. Therefore the envelope handler shall support 280 bytes of data.

4.2 Test environment description

The general architecture for the test environment is.



NOTE: Figure 4.2 shows the test architecture required to test interoperability at both API and bytcode level. The latter is currently not included in the current specification. The diagram is for information.

Figure 4.2

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

Chapter name:

xxx: 3 letters for each chapter

See annex F for full chapter acronyms list

Subchapter name

yyyy: : 4 letters for each subchapter

See annex F for full subchapter acronyms list

LDR: loader testing

[TBD]

4.3.1.1 Conformance requirements

The conformance requirements are expressed in the following way:

- Method prototype as listed in 3GPP TS 43.019 [7].
- Normal execution:
  - Contains normal execution and correct parameters limit values, each referenced as a Conformance Requirement Reference Normal (CRRN).
- Parameters error:
  - Contains parameter errors and incorrect parameter 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 (CRRC).

4.3.1.2 Test Area 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
- Load Script:	[Test Area Reference]_[Load Script number].ldr
- Cleanup Script:	[Test Area Reference]_[Cleanup Script number].clr
- Parameter File:	[Test Area Reference]_[Parameter File number].par

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

The test script, load script and cleanup script shall share a common syntax and format (see Annex B).

The parameter file has an own syntax (see annex G) and contains parameters to be used for CAP-file conversion and loading/cleanup script generation.

Scripts file shall be run in the following order:

[Test Area Reference]_1.ldr

[Test Area Reference]_1.scr

[Test Area Reference]_1.clr

[Test Area Reference]_2.ldr

[Test Area Reference]_2.scr

[Test Area Reference]_2.clr

….

[Test Area Reference]_n.ldr

[Test Area Reference]_n.scr

[Test Area Reference]_n.clr

In case that one of the files is not needed, it shall be skipped during the tests execution.

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 behaviour.*	*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 Initial Conditions

The Initial Conditions are a set of general prerequisites for the SIM prior to the execution of testing. For each test procedure described in the present document, the following rules apply to the Initial Conditions:

- unless otherwise stated, the file system and the files' content shall fulfil the requirements described in annex C;
- unless otherwise stated, before installing the applet(s) relevant to the current test procedure, all packages specific to other test procedures shall not be present.

When both statements apply, a test procedure is said to be in the "Default Initial Conditions" state.

4.5 Package name

Java packages integrating this Test Suite shall follow this naming convention:

sim.test.access.[Test Area Reference]: Java Card packages containing Test Area References for the 3GPP TS 43.019 [7] sim.access package.

sim.test.framework.[Test Area Reference]: Java Card packages containing Test Area References for the 3GPP TS 43.019 [7] framework.

sim.test.util: for the Test util package defined in this Test Suite.

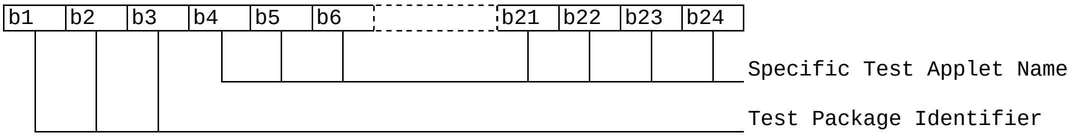
sim.test.toolkit.[Test Area Reference]: Java Card packages containing Test Area References for the 3GPP TS 43.019 [7] sim.toolkit package.

EXAMPLE: The package *../sim.test.access.[Test Area Reference]* creates the following directory structure *../sim/test/access/[Test Area Reference]/API_1_..._[1..n].**, where '*API_1_..._[1..n].**' are the different test applets Java source files used in *[Test Area Reference]*.

4.6 AID Coding

The AID coding for the Test Packages, Applet classes and Applet shall be as specified in TS 101 220 [14]. In addition, the following TAR values are defined for use within the present document:

TAR Coding (3 bytes/ 24 bits):



Test package Identifier(bits b1-b3):

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

001: API

010: Framework

011: Loader

111: sim.test.util

other values are RFU

Application Provider specific data (1 byte):

'00': for Package

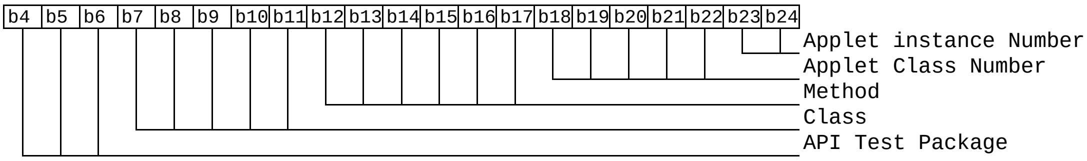
'01': for Applet class

'02': for Applet Instance

EXAMPLE: The AID of Package sim.test.util is 'A0 00 00 00 09 00 02 FF FF FF FF 89 E0 00 00 00'.

4.6.1 Specific Test Applet Name for API

Specific applet test name (bits b4-b24):



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 A for the full list

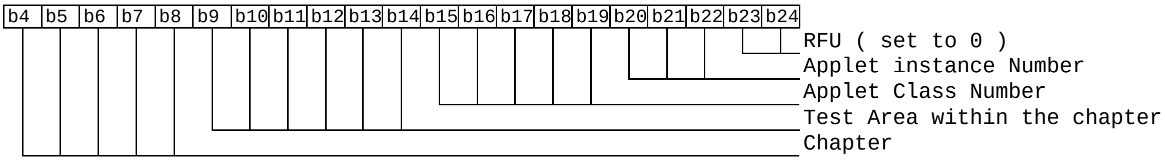
Method (6 bits): need to be assigned specification order see Annex A for the full list

Applet Class Number (5 bits): linked to Test Area, it shall start with 1 for classes and shall be 0 for package.

Applet Instance Number (2 bits) defined in the test procedure it shall start with 01 for applet instance and shall be 00 for package and class.

4.6.2 Specific Test Applet Name for Framework

Specific applet test name (bits b4-b24):



for Chapter (5 bits)

00001 Toolkit Installation Parameters

00010 Minimum Handler Availability

00011 Handler Integrity

00100 Applet Triggering

00101 Proactive Command Sending

00110 Framework Security

00111 Envelope Response Posting

01000 File System Context

01001 Exception Handling

01010 Other parts transferred to framework from API

01011 Concatenation processing

other are RFU

Test Area within the chapter (6 bits): values are defined in Annex F

Applet Class number (5 bits): linked to Test Area, it shall start with 1 for classes and shall be 0 for package.

Applet Instance number (3 bits) defined in the test procedure it shall start with 01 for applet instance and shall be 00 for package and class.

4.7 Test Equipment

These subclauses recommend a minimum specification for each of the items of test equipment referenced in the tests.

4.7.1 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.

4.7.2 Util package

Annex D includes java source code for the sim.test.util package as well as loading , testing and cleaning script examples.

4.7.3 Applet installation parameters

4.7.3.1 Security parameters

Loading scripts shall use the following security parameters as stated in 3GPP TS 23.048 [8] for applet installation:

Parameter	Value in hexadecimal
SPI	0A 00
KIC	00
KID	Value as described in the TS 23.048[8] (recommended value: 15)
TAR	00 00 00
CNTR	00 00 00 00 01
PCNTR	00
Key	Corresponding to KID (recommended value: 01 23 45 67 89 AB CD EF EF CD AB 89 67 45 23 01)

4.7.3.2 Loading components

Cap files in loading scripts shall not include the descriptor component as described in Java Card 2.1 VM Architecture Specification [13].

4.8 Testing methodology

4.8.1 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.access:

6.1.1 Interface SIMView

NOTE: The Test applet shall be run on a class that implements this interface.

6.1.1.1 Constants

Test Area Reference: API_1_SVW_CONST

6.1.1.1.1 Conformance Requirements

This subclause does not describe the conformance requirements for a method, but rather for the constants of the interface.

6.1.1.1.1.1 Normal execution

CRRN1: The constants shall have the same name and value that is defined in 3GPP TS 43.019 [7].

6.1.1.1.2 Test Suite Files

None.

6.1.1.1.3 Test Procedure

The constants in Java are resolved at compilation time, therefore a runtime test is not useful. No test of constants will be performed.

6.1.1.2 Method select(short fid, byte[] fci, short fciOffset, short fciLength)

Test Area Reference: API_1_SVW_SLCTS_BSS

6.1.1.2.1 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
```

6.1.1.2.1.1 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 FCI structure, the whole FCI structure 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 FCI structure, the first part of the FCI structure 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. See TS 43.019 [7] - §5.2. This will be tested during the testing of the framework.
- CRRN8: The information returned by fci shall be formatted as described in TS 51.011 [3], §9.2.1.
- CRRN9: The file with a File-ID that matches fid shall be found according to the following selection rules:

  1) An immediate child EF or DF of the current MF/DF can be selected,

  2) A sibling DF of the current DF can be selected,

  3) The current MF/DF it self can be selected,

  4) The parent MF/DF of the current DF can be selected,

  5) The MF can always be selected.

6.1.1.2.1.2 Parameter errors

- CRRP1: If the array fci is null, an instance of NullPointerException shall be thrown.
- CRRP2: If fciOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP3: If fciLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP4: If fciOffset plus fciLength is greater than the length of the array fci.length, or fciOffset equals fci.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

6.1.1.2.1.3 Context errors

- CRRC1: If the file with a File-ID which matches fid could not be found according to the selection rules listed in CRRN9, 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.

6.1.1.2.2 Test Suite Files

Additional requirements for the GSM personalization: None

Test Script:	API_1_SVW_SLCTS_BSS_1.scr
Test Applet:	API_1_SVW_SLCTS_BSS_1.java
Load Script:	API_1_SVW_SLCTS_BSS_1.ldr
Cleanup Script:	API_1_SVW_SLCTS_BSS_1.clr
Parameter File:	API_1_SVW_SLCTS_BSS_1.par

6.1.1.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored.	

1	**Select EFICCID 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 a value not greater than 20.
		
		<Description of fci:
		XX XX
		XX XX
		2F E2
		04
		>	

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 FCI structure.	

3	**Select DF$_{GSM}$ in MF**
	`fid = SIMView.FID_DF_GSM`
	`fciOffset = 0`
	`fciLength = 7`
	`select()`
		No exception shall be thrown.
		Shall return 7.
		fci shall contain the first 7 bytes of the FCI.
		<Description of fci:
		XX XX
		XX XX
		7F 20
		02
		>	

3	**Select DF$_{GSM}$ in MF**
	`fid = SIMView.FID_DF_GSM`
	`fciOffset = 0`
	`fciLength = 7`
	`select()`
		No exception shall be thrown.
		Shall return 7.
		fci shall contain the entire FCI structure.
		<Description of fci:
		XX XX
		XX XX
		7F 20
		02
		>	

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 a value between 15 and 20. (Cyclic EF)
		fci shall contain the first 15 or more bytes of the FCI structure.
		fci[14] 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 a value between 22 and 34.
		fci shall contain the entire FCI structure.	

6	**Select DF$_{TELECOM}$ in MF**
	`fid = SIMView.FID_DF_TELECOM`
	`fci[0] = fci[1] = '05'`
	`fciOffset = 2`
	`fciLength = 20`
	`select()`
		No exception shall be thrown.
		Shall return 20.
		fci shall contain the first 20 bytes of the FCI structure starting at index 2. The first two bytes shall (still) have the value '05'.	

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 FCI structure.
		fci[14] shall have the value 28 (length of record).	

8	**fci is null**
	`fid = SIMView.FID_EF_FDN`
	`byte[] nullBuffer = null`
	`fciOffset = 0`
	`fciLength = 15`
	`select()`
		Shall throw java.lang.NullPointerException.	

Id	Description	API Expectation	APDU Expectation

9	**fciOffset < 0**
	fid = SIMView.FID_EF_FDN
	fciOffset = -1
	fciLength = 15
	select()
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

10	**fciLength < 0**
	fid = SIMView.FID_EF_FDN
	fciOffset = 0
	fciLength = -1
	select()
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

11	**fciOffset + fciLength > fci.length**
	fid = SIMView.FID_EF_FDN
	fciOffset = 20
	fciLength = 15
	select()
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

12	**fciOffset □ fci.length**
	fid = SIMView.FID_EF_FDN
	fciOffset = 34
	fciLength = 1
	select()
		Shall throw java.lang.ArrayIndexOutOfBoundsException	

13	**Selection possibilities**
	1 - fid = SIMView.FID_MF
	fciOffset = 0
	fciLength = 15
	select()
	2 - fid = SIMView.FID_DF_TELECOM
	select()
	3 - fid = SIMView.FID_DF_GRAPHICS
	select()
	4 - fid = SIMView.FID_DF_TELECOM
	select()
	5 - fid = SIMView.FID_DF_GRAPHICS
	select()
	6 - fid = SIMView.FID_MF
	select()
	7 - fid = SIMView.FID_DF_GSM
	select()
	8 - fid = SIMView.FID_DF_TELECOM
	select()
	9 - fid = SIMView.FID_DF_TELECOM
	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.
		6 - No exception shall be thrown.
		7 - No exception shall be thrown.
		8 - No exception shall be thrown.
		9 - No exception shall be thrown.	

14	**EF not selected after MF/DF selection**
	1 - fid = SIMView.FID_MF
	select()
	fid = SIMView.FID_EF_ICCID
	select()
	2 - fid = SIMView.FID_MF
	select()
	readBinary()
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code NO_EF_SELECTED.	

15	**No selection of non-reachable 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	**No record is selected after selecting linear fixed EF**
	1 - fid = SIMView.FID_MF
	select()
	2 - fid = FID_DF_SIMTEST
	select()
	3 - fid = FID_EF_LARU
	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.	

Id	Description	API Expectation	APDU Expectation

17	**Record pointer in selected cyclic EF**
	`1 - fid = SIMView.FID_MF`
	`select()`
	`2 - fid = FID_DF_SIMTEST`
	`select()`
	`3 - fid = FID_EF_CARU`
	`select()`
	`4 - byte[] data1 = { 1,2,3 }`
	`mode = REC_ACC_MODE_PREVIOUS`
	`updateRecord(data1)`
	`5 - fid = FID_EF_CARU`
	`select()`
	`readRecord(data2)`
	`compare data1 to data2`
		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 - The contents of data1 and data2 shall be identical.	

6.1.1.2.4 Test Coverage

CRR Number	Test Case Number
N1	1-7
N2	3, 5
N3	1, 2, 4, 6, 7
N4	14
N5	16
N6	17
N8	1, 3
N9	1-7, 13
P1	8
P2	9
P3	10
P4	11, 12
C1	15
C2, C3	Not Tested

6.1.1.3 Method select (short fid)

Test Area Reference: API_1_SVW_SLCTS

6.1.1.3.1 Conformance Requirements

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

```
public void select(short fid)
           throws SIMViewException
```

6.1.1.3.1.1 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 [TS 43.019 [7] - §5.2]. This will be tested during the testing of the framework.
- CRRN6: The file with a File-ID that matches fid shall be found according to the following selection rules:

  1) An immediate child EF or DF of the current MF/DF can be selected,

  2) A sibling DF of the current DF can be selected,

  3) The current MF/DF it self can be selected,

4) The parent MF/DF of the current DF can be selected,

5) The MF can always be selected.

6.1.1.3.1.2 Parameter errors

No requirements.

6.1.1.3.1.3 Context errors

- CRRC1: If the file with a File-ID which matches fid could not be found according to the selection rules listed in CCRN6, 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.

6.1.1.3.2 Test Suite Files

Additional requirements for the GSM personalization: None

Test Script:	API_1_SVW_SLCTS_1.scr
Test Applet:	API_1_SVW_SLCTS_1.java
Load Script:	API_1_SVW_SLCTS_1.ldr
Cleanup Script:	API_1_SVW_SLCTS_1.clr
Parameter File:	API_1_SVW_SLCTS_1.par

6.1.1.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored.	

1	**Select EF$_{ICCID}$ in MF (Transparent EF)**
	`fid = SIMView.FID_EF_ICCID`
	`select()`
		No exception shall be thrown.	

2	**EF not selected after MF/DF selection**
	`1 - fid = SIMView.FID_MF`
	`select()`
	`fid = SIMView.FID_EF_ICCID`
	`select()`
	`2 - fid = SIMView.FID_MF`
	`select()`
	`readBinary()`
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code NO_EF_SELECTED.	

3	**No record is selected after selecting linear fixed EF**
	`1 - fid = SIMView.FID_MF`
	`select()`
	`2 - fid = FID_DF_SIMTEST`
	`select()`
	`3 - fid =FID_EF_LARU`
	`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.	

Id	Description	API Expectation	APDU Expectation

4	**Record pointer in selected cyclic EF**
	1 - fid = SIMView.FID_MF
	select()
	2 - fid =FID_DF_SIMTEST
	select()
	3 - fid = FID_EF_CARU
	select()
	4 - byte[] data1 = { 1,2,3 }
	updateRecord(data1)
	5 - fid = FID_EF_CARU
	select()
	readRecord(data2)
	compare data1 to data2
		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 - The contents of data1 and data2 shall be identical.	

5	**Selection possibilities** 1 - fid = SIMView.FID_MF select() 2 - fid = SIMView.FID_DF_TELECOM select() 3 - fid = SIMView.FID_DF_GRAPHICS select() 4 - fid = SIMView.FID_DF_TELECOM select() 5 - fid = SIMView.FID_DF_GRAPHICS select() 6 - fid = SIMView.FID_MF select() 7 - fid = SIMView.FID_DF_GSM select() 8 - fid = SIMView.FID_DF_TELECOM select() 9 - fid = SIMView.FID_DF_TELECOM 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. 6 - No exception shall be thrown. 7 - No exception shall be thrown. 8 - No exception shall be thrown. 9 - No exception shall be thrown.	
6	**No selection of unreachable 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.	

6.1.1.3.4 Test Coverage

CRR Number	Test Case Number
N1	1
N2	2
N3	3
N4	4
N6	5
C1	6
C2, C3	Not Tested

6.1.1.4 Method status

Test Area Reference: API_1_SVW_STAT_BSS

6.1.1.4.1 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
```

6.1.1.4.1.1 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 (described in 3GPP TS 43.019 [7], subclause 9.2.1).

- CRRN2: If the length fciLength is greater than or equal to the length of the FCI structure, the whole FCI structure 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 FCI structure, the first part of the FCI structure is copied into the array fci and the length of the FCI which has been written to the array fci is returned.

6.1.1.4.1.2 Parameter errors

- CRRP1: If the array fci is null, an instance of NullPointerException shall be thrown.
- CRRP2: If fciOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP3: If fciLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP4: If fciOffset plus fciLength is greater than the length of the array fci.length, or fciOffset equals fci.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

6.1.1.4.1.3 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.

6.1.1.4.2 Test Suite Files

Additional requirements for the GSM personalization:

Test Script:	API_1_SVW_STAT_BSS_1.scr
Test Applet:	API_1_SVW_STAT_BSS_1.java
Load Script:	API_1_SVW_STAT_BSS_1.ldr
Cleanup Script:	API_1_SVW_STAT_BSS_1.clr
Parameter File:	API_1_SVW_STAT_BSS_1.par

6.1.1.4.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored.	

1	**Status of MF**
	`byte[] fci = new byte[34]`
	`fciOffset = 0`
	`fciLength = 7`
	`status()`
		No exception shall be thrown.
		Shall return 7.
		fci shall contain the entire FCI structure.
		<Description of fci:
		XX XX
		XX XX
		3F 00
		01
		>	

2	**Status after select EF$_{ICCID}$ in MF**
	`1 - fid = SIMView.FID_DF_GSM`
	`fciOffset = 0`
	`fciLength = 34`
	`len = select()`
	`2 - byte[] fci2 = new byte[34]`
	
	`len2 = status()`
	`3 - Compare len and len2`
	`4 - Compare the len bytes of fci and fci2`
		1 - No exception shall be thrown. Shall return a value between 22 and 34.
		2 - No exception shall be thrown. Shall return 22 or more.
		3 - len and len2 shall be identical
		4 - fci and fci2 shall be identical	

Id	Description	API Expectation	APDU Expectation

3	**Status of $\text{DF}_{\text{Telecom}}$**
	1 - fid = SIMView.FID_DF_TELECOM
	select()
	2 - fciOffset = 0
	fciLength = 100
	status()
		1 - No exception shall be thrown. Shall return a value between 22 and 34.
		2 - No exception shall be thrown. Shall return a value between 22 and34.
		fci shall contain the entire FCI structure (check that returned value is equal to 13 plus the "length of following data" - fci[12]).FID of the returned fci (fci[4:5]) is FID_DF_TELECOM.	

4	**Status $\text{DF}_{\text{TELECOM}}$**
	fciOffset = 0
	fciLength = 7
	status()
		No exception shall be thrown.
		Shall return 7.
		fci shall contain the first 7 bytes of the FCI structure starting at index 0.
		FID of the returned fci (fci[4:5]) is FID_DF_TELECOM.	

5	**fci is null**
	byte[] nullBuffer = null
	fciOffset = 0
	fciLength = 34
	status()
		Shall throw java.lang.NullPointerException.	

6	**fciOffset < 0**
	fciOffset = -1
	fciLength = 34
	status()
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

7	**fciLength < 0**
	fciOffset = 0
	fciLength = -1
	status()
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

8	**fciOffset + fciLength > fci.length**
	fciOffset = 20
	fciLength = 15
	status()
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

9	**fciOffset ≥ fci.length**
	fciOffset = 34
	fciLength = 1
	status()
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

6.1.1.4.4 Test Coverage

CRR Number	Test Case Number
N1	1-4
N2	2, 3
N3	1, 4
P1	5
P2	6
P3	7
P4	8, 9
C1, C2	Not Tested

6.1.1.5 Method readBinary

Test Area Reference: API_1_SVW_REDBS_BSS

6.1.1.5.1 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
```

6.1.1.5.1.1 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.

6.1.1.5.1.2 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.

6.1.1.5.1.3 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 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.

6.1.1.5.2 Test Suite Files

Additional requirements for the GSM personalization: none.

Test Script:	API_1_SVW_REDBS_BSS_1.scr
Test Applet:	API_1_SVW_REDBS_BSS_1.java
Load Script:	API_1_SVW_REDBS_BSS_1.ldr

Cleanup Script: API_1_SVW_REDBS_BSS_1.clr

Parameter File: API_1_SVW_REDBS_BSS_1.par

6.1.1.5.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored	

1	**Read from EFICCID in MF (Transparent EF)**
	`1 - fid = SIMView.FID_EF_ICCID`
	`select()`
	`2 - fileOffset = 0`
	`byte[] resp = new byte[20]`
	`resp[0:19] = '55'`
	`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 EFICCID starting at index 10.
		<Description of resp:
		55 55 55 55 55 55 55 55 55 55
		0F FF FF FF FF FF FF FF FF FF
		>	

2	**Read from EFICCID in MF**
	`resp[0:19] = '55'`
	`fileOffset = 5`
	`respOffset = 10`
	`respLength = 5`
	`readBinary()`
		No exception shall be thrown.
		Shall return 15.
		resp shall contain the last 5 bytes of EFICCID starting at index 10.
		<Description of resp:
		55 55 55 55 55 55 55 55 55 55
		FF FF FF FF FF 55 55 55 55 55
		>	

3	**Offset into File out of bounds**
	`fileOffset = -1`
	`respOffset = 0`
	`respLength = 10`
	`readBinary()`
		Shall throw sim.access.SIMViewException with reason code OUT_OF_FILE_BOUNDARIES.	

4	**fileOffset + respLength > EF length**
	`fileOffset = 9`
	`respOffset = 0`
	`respLength = 2`
	`readBinary()`
		Shall throw sim.access.SIMViewException with reason code OUT_OF_FILE_BOUNDARIES.	

5	**resp is null**
	`byte[] nullBuffer = null`
	`fileOffset = 0`
	`respOffset = 0`
	`respLength = 10`
	`readBinary()`
		Shall throw java.lang.NullPointerException.	

6	**respOffset < 0**
	`fileOffset = 0`
	`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.	

9	**EF is not Transparent**
	`1 - fid = FID_DF_SIMTEST`
	`select()`
	`2 - fid = FID_EF_LARU`
	`select()`
	`3 - fileOffset = 0`
	`respOffset = 0`
	`respLength = 1`
	`readBinary()`
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		3 - Shall throw sim.access.SIMViewException with reason code FILE_INCONSISTENT.	

Id	Description	API Expectation	APDU Expectation

10	**Access condition not fulfilled**
	`1 - fid = DFSIMTTEST`
	`select()`
	`2 - fid = EFTNR`
	`select()`
	`3 - fileOffset = 0`
	`respOffset = 0`
	`respLength = 1`
	`readBinary()`
		Shall throw sim.access.SIMViewException with reason code AC_NOT_FULFILLED.	

11	**EF is invalidated**
	`1 - fid = EFTNU`
	`invalidate()`
	`2 - readBinary()`
	`3 - rehabilitate()`
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code INVALIDATION_STATUS_CONTRADICTION.
		3 - No exception shall be thrown.	

12	**No EF selected**
	`1- fid = SIMView.FID_MF`
	`select()`
	`2 readBinary()`
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code NO_EF_SELECTED.	

6.1.1.5.4 Test Coverage

CRR Number	Test Case Number
N1	1-2
P1	3
P2	4
P3	5
P4	6
P5	7
P6	8,
C1	12
C2	9
C3	10
C4	11
C5, C6	Not Tested

6.1.1.6 Method updateBinary

Test Area Reference: API_1_SVW_UPDBS_BSS

6.1.1.6.1 Conformance Requirements

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

```
public void updateBinary(short fileOffset,
                         byte[] data,
                         short dataOffset,
                         short dataLength)
                  throws java.lang.NullPointerException,
                         java.lang.ArrayIndexOutOfBoundsException,
                         SIMViewException
```

6.1.1.6.1.1 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.

6.1.1.6.1.2 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.

6.1.1.6.1.3 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 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.

6.1.1.6.2 Test Suite Files

Additional requirements for the GSM personalization: None

Test Script:	API_1_SVW_UPDBS_BSS_1.scr
Test Applet:	API_1_SVW_UPDBS_BSS_1.java
Load Script:	API_1_SVW_UPDBS_BSS_1.ldr
Cleanup Script:	API_1_SVW_UPDBS_BSS_1.clr
Parameter File:	API_1_SVW_UPDBS_BSS_1.par

6.1.1.6.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored.	

1	**No EF selected**
	`fileOffset = 0`
	`byte[] data = new byte[20]`
	`data[0] = '55'`
	`dataOffset = 0`
	`dataLength = 10`
	`updateBinary()`
		Shall throw sim.access.SIMViewException with reason code NO_EF_SELECTED.	

Id	Description	API Expectation	APDU Expectation

2	**Update Transparent EF**
	`1 - fid = DFSIMTEST`
	`select()`
	`2 - fid = EFTARU`
	`select()`
	`3 - fileOffset = 3`
	`data[0] = '55'`
	`dataOffset = 0`
	`dataLength = 1`
	`updateBinary()`
	`4 - fileOffset = 3`
	`respOffset = 0`
	`respLength = 1`
	`readBinary()`
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		3 - No exception shall be thrown.
		4 - No exception shall be thrown.
		Data in resp[0] shall be '55'.	

3	**1 - fileOffset = 254**
	`data[0] = '55'`
	`data[1] = 'AA'`
	`data[2] = '66'`
	`dataOffset = 0`
	`dataLength = 3`
	`updateBinary()`
	`2 - fileOffset = 254`
	`respOffset = 0`
	`respLength = 3`
	`readBinary()`
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		Data in resp shall be
		resp[0] = '55'
		resp[1] = 'AA'
		resp[2] = '66'	

4	**Offset into File out of bounds**
	`fileOffset = -1`
	`dataOffset = 0`
	`dataLength = 10`
	`updateBinary()`
		Shall throw sim.access.SIMViewException with reason code OUT_OF_FILE_BOUNDARIES.	

5	**fileOffset + dataLength > EF length**
	`fileOffset = 259`
	`dataOffset = 0`
	`dataLength = 2`
	`updateBinary()`
		Shall throw sim.access.SIMViewException with reason code OUT_OF_FILE_BOUNDARIES.	

6	**data is null**
	`byte[] nullBuffer = null`
	`fileOffset = 0`
	`dataOffset = 0`
	`dataLength = 10`
	`updateBinary()`
		Shall throw java.lang.NullPointerException.	

7	**dataOffset < 0**
	`fileOffset = 0`
	
	`dataOffset = -1`
	`dataLength = 10`
	`updateBinary()`
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

8	**dataLength < 0**
	`fileOffset = 0`
	`dataOffset = 0`
	`dataLength = -1`
	`updateBinary()`
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

9	**dataOffset + dataLength > data.length**
	`fileOffset = 0`
	`dataOffset = 10`
	`dataLength = 11`
	`updateBinary()`
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

10	**EF is not Transparent**
	`1 - fid = FID_DF_SIMTEST`
	`select()`
	`2 - fid = FID_EF_LARU`
	`select()`
	`3 - fileOffset = 0`
	`data[0] = '55'`
	`dataOffset = 0`
	`dataLength = 1`
	`updateBinary()`
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		3 - Shall throw sim.access.SIMViewException with reason code FILE_INCONSISTENT.	

Id	Description	API Expectation	APDU Expectation

11	**Access condition not fulfilled**
	1 - fid = DFSIMTEST
	select()
	fid = EFTNU
	select()
	2 - fileOffset = 0
	data[0] = '55'
	dataOffset = 0
	dataLength = 1
	updateBinary()
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code AC_NOT_FULFILLED.	

12	**EF is invalidated**
	1 - fid = EFTNR
	invalidate()
	2 - fileOffset = 0
	data[0] = '55'
	dataOffset = 0
	dataLength = 1
	updateBinary()
	3 - rehabilitate()
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code INVALIDATION_STATUS_CONTRADICTION.
		3 - No exception shall be thrown.	

6.1.1.6.4 Test Coverage

CRR Number	Test Case Number
N1	2, 3
P1	4
P2	5
P3	6
P4	7
P5	8
P6	9
C1	1
C2	10
C3	11
C4	12
C5, C6	Not Tested

6.1.1.7 Method readRecord

Test Area Reference: API_1_SVW_REDRSBS_BSS

6.1.1.7.1 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
```

6.1.1.7.1.1 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.

6.1.1.7.1.2 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 length, 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 array resp is null, an instance of NullPointerException shall be thrown.
- CRRP9: If respOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP10: If respLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP11: If respOffset plus respLength is greater than the length of the array resp.length, or respOffset equals resp.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

6.1.1.7.1.3 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 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.

6.1.1.7.2 Test Suite Files

Additional requirements for the GSM personalization: None

Test Script:	API_1_SVW_REDRSBS_BSS_1.scr
Test Applet:	API_1_SVW_REDRSBS_BSS_1.java
Load Script:	API_1_SVW_REDRSBS_BSS_1.ldr
Cleanup Script:	API_1_SVW_REDRSBS_BSS_1.clr
Parameter File:	API_1_SVW_REDRSBS_BSS_1.par

6.1.1.7.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored.	

1	**No EF selected**
	`recNumber = 1`
	`mode = REC_ACC_MODE_ABSOLUTE_CURRENT`
	`recOffset = 0`
	`byte[] resp = new byte[20]`
	`respOffset = 0`
	`respLength = 10`
	`readRecord()`
		Shall throw sim.access.SIMViewException with reason code NO_EF_SELECTED.	

Id	Description	API Expectation	APDU Expectation

2	**Read Absolute and Current from Linear Fixed EF**
	`1 - fid = DFSIMTEST`
	`select()`
	`2 - fid = EFLARU`
	`select()`
	`// Record pointer not set.`
	`3 - recNumber = 0`
	`mode = REC_ACC_MODE_NEXT`
	`recOffset = 0`
	`respOffset = 0`
	`respLength = 4`
	`readRecord()`
	`4 - recNumber = 2`
	`mode = REC_ACC_MODE_ABSOLUTE_CURRENT`
	`readRecord()`
	`5 - recNumber = 1`
	`readRecord()`
	`6 - recNumber = 0`
	`resp[0] = resp[1] = resp[2] = resp[3] = '00'`
	`readRecord()`
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		3 - No exception shall be thrown.
		resp shall be:
		resp[0] = '55'
		resp[1] = '55'
		resp[2] = '55'
		resp[3] = '55'
		4 - No exception shall be thrown.
		resp shall be:
		resp[0] = 'AA'
		resp[1] = 'AA'
		resp[2] = 'AA'
		resp[3] = 'AA'
		5 - No exception shall be thrown.
		resp shall be:
		resp[0] = '55'
		resp[1] = '55'
		resp[2] = '55'
		resp[3] = '55'
		6 - No exception shall be thrown.
		resp shall be:
		resp[0] = '55'
		resp[1] = '55'
		resp[2] = '55'
		resp[3] = '55'	

3	**Read Next from Linear Fixed EF**
	`recNumber = 0`
	`mode = REC_ACC_MODE_NEXT`
	`recOffset = 0`
	`respOffset = 0`
	`respLength = 4`
	`readRecord()`
		No exception shall be thrown.
		resp shall be:
		resp[0] = 'AA'
		resp[1] = 'AA'
		resp[2] = 'AA'
		resp[3] = 'AA'	

4	**Read Next from Linear Fixed EF**
	`recNumber = 0`
	`mode = REC_ACC_MODE_NEXT`
	`recOffset = 0`
	`respOffset = 0`
	`respLength = 4`
	`readRecord()`
		Shall throw sim.access.SIMViewException with reason code RECORD_NUMBER_NOT_AVAILABLE.	

5	**Read Previous from Linear Fixed EF**
	`recNumber = 0`
	`mode = REC_ACC_MODE_PREVIOUS`
	`recOffset = 0`
	`respOffset = 0`
	`respLength = 4`
	`readRecord()`
		No exception shall be thrown.
		resp shall be:
		resp[0] = '55'
		resp[1] = '55'
		resp[2] = '55'
		resp[3] = '55'	

6	**Read Previous from Linear Fixed EF**
	`recNumber = 0`
	`mode = REC_ACC_MODE_PREVIOUS`
	`recOffset = 0`
	`respOffset = 0`
	`respLength = 4`
	`readRecord()`
		Shall throw sim.access.SIMViewException with reason code RECORD_NUMBER_NOT_AVAILABLE.	

Id	Description	API Expectation	APDU Expectation

7	**Read Absolute and Current from Cyclic EF**
	1 - fid = EFCARU
	select()
	2 - recNumber = 2
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
	3 - recNumber = 1
	readRecord()
	4 - recNumber = 0
	resp[0] = resp[1] = resp[2] = '00
	readRecord()
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		resp shall be:
		resp[0] = ' AA'
		resp[1] = ' AA'
		resp[2] = ' AA'
		3 - No exception shall be thrown.
		resp shall be:
		resp[0] = '55'
		resp[1] = '55'
		resp[2] = '55'
		4 - No exception shall be thrown.
		resp shall be:
		resp[0] = '55'
		resp[1] = '55'
		resp[2] = '55'	

8	**Read Next from Cyclic EF**
	recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
		No exception shall be thrown.
		resp shall be:
		resp[0] = 'AA'
		resp[1] = 'AA'
		resp[2] = 'AA'	

9	**Read Next from Cyclic EF**
	recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
		No exception shall be thrown.
		resp shall be:
		resp[0] = '55'
		resp[1] = '55'
		resp[2] = '55'	

10	**Read Previous from Cyclic EF**
	recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
		No exception shall be thrown.
		resp shall be:
		resp[0] = 'AA'
		resp[1] = 'AA'
		resp[2] = 'AA'	

11	**Read Previous from Cyclic EF**
	recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
		No exception shall be thrown.
		resp shall be:
		resp[0] = '55'
		resp[1] = '55'
		resp[2] = '55'	

12	**Read Absolute from Linear Fixed EF beyond** Records
	1 - fid = EFLARU
	select()
	2 - recNumber = -1
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = 0
	respOffset = 0
	respLength = 4
	readRecord()
	3 - recNumber = 3
	readRecord()
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code RECORD_NUMBER_NOT_AVAILABLE.
		3 - Shall throw sim.access.SIMViewException with reason code RECORD_NUMBER_NOT_AVAILABLE.	

13	**No current record in linear fixed EF, read** current
	1 - fid = EFLARU
	select() // No curr rec
	2 - recNumber = 0 // curr rec
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = 0
	respOffset = 0
	respLength = 4
	readRecord()
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code RECORD_NUMBER_NOT_AVAILABLE.	

Id	Description	API Expectation	APDU Expectation

14	**recOffset < 0**
	1 - fid = EFLARU
	select()
	2 - recNumber = 1 // rec 1
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = -1
	respOffset = 0
	respLength = 4
	readRecord()
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code OUT_OF_RECORD_BOUNDARIES.	

15	**recOffset + respLength > Record Length**
	1 - fid = EFLARU
	select()
	2 - recNumber = 1
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = 2
	respOffset = 0
	respLength = 4
	readRecord()
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code OUT_OF_RECORD_BOUNDARIES.	

16	**Reading with invalid mode**
	1 - fid = EFLARU
	select()
	2 - recNumber = 0
	mode = 1
	recOffset = 0
	respOffset = 0
	respLength = 4
	readRecord()
	3 - mode = 5
	readRecord()
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code INVALID_MODE.
		3 - Shall throw sim.access.SIMViewException with reason code INVALID_MODE.	

17	**resp is null**
	byte[] nullBuffer = null
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	respOffset = 0
	respLength = 10
	readRecord()
		Shall throw java.lang.NullPointerException.	

18	**respOffset < 0**
	respOffset = -1
	respLength = 10
	readRecord ()
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

19	**respLength < 0**
	respOffset = 0
	respLength = -1
	readRecord ()
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

20	**respOffset + respLength > resp.length**
	respOffset = 10
	respLength = 11
	readRecord ()
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

21	**EF is neither Cyclic nor Linear Fixed**
	1 - fid = DFSIMTEST
	select()
	2 - fid = EFTNU
	select()
	3 - respOffset = 0
	respLength = 4
	readRecord()
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		3 - Shall throw sim.access.SIMViewException with reason code FILE_INCONSISTENT.	

22	**Access condition not fulfilled** 1 - fid = EFCNR select() 2 - respLength = 3 readRecord()	1 - No exception shall be thrown. 2 - Shall throw sim.access.SIMViewException with reason code AC_NOT_FULFILLED.	
23	**EF is invalidated** 1 - fid = EFCNU invalidate() 2 - readRecord() 3 - rehabilitate()	1 - No exception shall be thrown. 2 - Shall throw sim.access.SIMViewException with reason code INVALIDATION_STATUS_CONTRADICTION. 3 - No exception shall be thrown.	

6.1.1.7.4 Test Coverage

CRR Number	Test Case Number
N1	2-5, 7-11
N2	2, 7
N3	3, 8, 9
N4	5, 10, 11
P1	12
P2	13
P3	4
P4	6
P5	14
P6	15
P7	16
P8	17
P9	18
P10	19
P11	20
C1	1
C2	21
C3	22
C4	23
C5, C6	Not Tested

6.1.1.8 Method updateRecord

Test Area Reference: API_1_SVW_UPDRSBS_BSS

6.1.1.8.1 Conformance Requirements

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

```
public void updateRecord(short recNumber,
                         byte mode,
                         short recOffset,
                         byte[] data,
                         short dataOffset,
                         short dataLength)
                  throws java.lang.NullPointerException,
                         java.lang.ArrayIndexOutOfBoundsException,
                         SIMViewException
```

6.1.1.8.1.1 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.

6.1.1.8.1.2 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 length, 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 array data is null, an instance of NullPointerException shall be thrown.
- CRRP10: If dataOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP11: If dataLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP12: If dataOffset plus dataLength, is greater than the length of the array data.length, or dataOffset equals data.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

6.1.1.8.1.3 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 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.

6.1.1.8.2 Test Suite Files

Additional requirements for the GSM personalization: This test is based on the assumption that the contents of the EFs in $DF_{SIMTEST}$ are identical to those defined in the default pre-personalization and the current record pointers have not been altered.

Test Script:	API_1_SVW_UPDRSBS_BSS_1.scr
Test Applet:	API_1_SVW_UPDRSBS_BSS_1.java
Load Script:	API_1_SVW_UPDRSBS_BSS_1.ldr
Cleanup Script:	API_1_SVW_UPDRSBS_BSS_1.clr
Parameter File:	API_1_SVW_UPDRSBS_BSS_1.par

6.1.1.8.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored.	

1	**No EF selected**
	`recNumber = 1`
	`mode = REC_ACC_MODE_ABSOLUTE_CURRENT`
	`recOffset = 0`
	`byte[] data = new byte[20]`
	`dataOffset = 0`
	`dataLength = 10`
	`updateRecord()`
		Shall throw sim.access.SIMViewException with reason code NO_EF_SELECTED.	

2	**Update Absolute and Current from Linear Fixed EF**
	`1 - fid = DFSIMTEST`
	`select()`
	`2 - fid = EFLARU`
	`select()`
	`// Record pointer not set.`
	
	`3 - recNumber = 2`
	`mode = REC_ACC_MODE_ABSOLUTE_CURRENT`
	`data[0:3] = '11'`
	`recOffset = 0`
	`dataOffset = 0`
	`dataLength = 4`
	`updateRecord()`
	`respOffset = 0`
	`respLength = 0`
	`readRecord()`
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		3 - No exception shall be thrown.
		Resp shall be:
		Resp[0] = '11'
		Resp[1] = '11'
		Resp[2] = '11'
		Resp[3] = '11'
			= 4

Id	Description	API Expectation	APDU Expectation

3	**Update Current from Linear Fixed EF**
	1 - fid = DFSIMTEST
	select()
	2 - fid = EFLARU
	select()
	// Set record pointer with mode "next".
	3 - recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	data[0:3] = '00'
	dataOffset = 0
	dataLength = 4
	updateRecord()
	// write data with mode "current"
	4 - recNumber = 0
	data[0:3] = '22'
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	updateRecord()
	// read result with mode "absolute"
	respOffset = 0
	respLength = 4
	recNumber = 1
	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 - No exception shall be thrown.
		resp shall be:
		resp[0] = '22'
		resp[1] = '22'
		resp[2] = '22'
		resp[3] = '22'	

4	**Update Next from Linear Fixed EF, no record pointer set**
	1 - fid = FID_DF_SIMTEST
	select()
	2 - fid = FID_EF_LARU
	select
	3 - recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	data[0:3] = '33'
	dataOffset = respOffset = 0
	dataLength = respLength = 4
	updateRecord()
	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.
		Resp shall be:
		Resp[0] = '33'
		Resp[1] = '33'
		Resp[2] = '33'
		Resp[3] = '33'	

5	**Update Next from Linear Fixed EF, record pointer set**
	1 - recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	data[0:3] = '44'
	dataOffset = 0
	dataLength = 4
	updateRecord()
	2 - mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	readRecord()
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		resp shall be:
		resp[0] = '44'
		resp[1] = '44'
		resp[2] = '44'
		resp[3] = '44'	

6	**Update Next from Linear Fixed EF, no more records**
	recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	data[0:3] = '55'
	dataOffset = 0
	dataLength = 4
	updateRecord()
		Shall throw sim.access.SIMViewException with reason code RECORD_NUMBER_NOT_AVAILABLE.	

7	**Update Previous from Linear Fixed EF, no record pointer set**
	1 - fid = DFSIMTEST
	select()
	2 - fid = EFLARU
	select()
	3 - recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	data[0:3] = '66'
	dataOffset = respOffset = 0
	dataLength = respLength = 4
	updateRecord()
	4 - 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 - No exception shall be thrown.
		resp shall be:
		resp[0] = '66'
		resp[1] = '66'
		resp[2] = '66'
		resp[3] = '66'	

Id	Description	API Expectation	APDU Expectation

8	**Update Previous from Linear Fixed EF, record pointer set**
	1 - recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	data[0:3] = '77'
	dataOffset = respOffset = 0
	dataLength = respLength = 4
	updateRecord()
	readRecord()
	2 - mode = REC_ACC_MODE_ABSOLUTE_CURRENT
		1 - No exception shall be thrown
		2 - No exception shall be thrown.
		Resp shall be:
		Resp[0] = '7744'
		Resp[1] = '7744'
		Resp[2] = '7744'
		Resp[3] = '7744'	

9	**Update Previous from Linear Fixed EF , no more records**
	recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	data[0:3] = '88'
	dataOffset = respOffset = 0
	dataLength = respLength = 4
	updateRecord()
		Shall throw sim.access.SIMViewException with reason code RECORD_NUMBER_NOT_AVAILABLE.	

10	**Update Previous from Cyclic EF**
	1 - fid = FID_DF_SIMTEST
	select()
	2 - fid = FID_EF_CARU
	select()
	3 - recNumber = 2
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
	4 - recNumber = 2
	mode = REC_ACC_MODE_PREVIOUS
	data[0:2] = resp[0:2] ^ 'FF'
	dataOffset = 0
	dataLength = 3
	updateRecord()
	5 - recNumber = 0
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	respOffset = 0
	respLength = 3
	readRecord()
		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.
		resp shall be:
		resp[0] = data[0]
		resp[1] = data[1]
		resp[2] = data[2]	

11	**Update Absolute from Linear Fixed EF beyond Records**
	1 - fid = EFLARU
	select()
	2 -recNumber = -1
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = 0
	dataOffset = 0
	dataLength = 4
	updateRecord()
	2 - recNumber = 3
	updateRecord()
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code RECORD_NUMBER_NOT_AVAILABLE.
		3 - Shall throw sim.access.SIMViewException with reason code RECORD_NUMBER_NOT_AVAILABLE.	

12	**No current record in linear fixed EF, update current**
	1 - fid = EFLARU
	select() // No curr rec
	2 - recNumber = 0 // curr rec
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = 0
	dataOffset = 0
	dataLength = 4
	updateRecord()
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code RECORD_NUMBER_NOT_AVAILABLE.	

13	**recOffset < 0**
	1 - fid = EFLARU
	select()
	2 - recNumber = 1 // rec 1
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = -1
	dataOffset = 0
	dataLength = 4
	updateRecord()
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code OUT_OF_RECORD_BOUNDARIES.	

Id	Description	API Expectation	APDU Expectation

14	**recOffset + dataLength > Record Length**
	`1 - fid = EFLARU`
	`select()`
	`2 - recNumber = 1`
	`mode = REC_ACC_MODE_ABSOLUTE_CURRENT`
	`recOffset = 2`
	`dataOffset = 0`
	`dataLength = 4`
	`updateRecord()`
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code OUT_OF_RECORD_BOUNDARIES.	

15	**Updating with invalid mode**
	`1 - fid = EFLARU`
	`select()`
	`2 - recNumber = 0`
	`mode = 1`
	`recOffset = 0`
	`dataOffset = 0`
	`dataLength = 4`
	`updateRecord()`
	`3 - mode = 5`
	`updateRecord()`
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code INVALID_MODE.
		3 - Shall throw sim.access.SIMViewException with reason code INVALID_MODE.	

16	**Updating Cyclic EF with invalid mode**
	`1 - fid = DFSIMTEST`
	`select()`
	`2 - fid = EFCARU`
	`select()`
	`3 - recNumber = 0`
	`mode = REC_ACC_MODE_NEXT`
	`recOffset = 0`
	`data[0:2] = '00'`
	`dataOffset = 0`
	`dataLength = 3`
	`updateRecord()`
	`4 - recNumber = 0`
	`mode = REC_ACC_MODE_ABSOLUTE_CURRENT`
	`updateRecord()`
	`5 - recNumber = 2`
	`mode = REC_ACC_MODE_ABSOLUTE_CURRENT`
	`updateRecord()`
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		3 - Shall throw sim.access.SIMViewException with reason code INVALID_MODE.
		4 - Shall throw sim.access.SIMViewException with reason code INVALID_MODE.
		5 - Shall throw sim.access.SIMViewException with reason code INVALID_MODE.	

17	**data is null**
	`byte[] nullBuffer = null`
	`dataOffset = 0`
	`dataLength = 10`
	`updateRecord()`
		Shall throw java.lang.NullPointerException.	

18	**dataOffset < 0**
	`dataOffset = -1`
	`dataLength = 10`
	`updateRecord()`
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

19	**dataLength < 0**
	`dataOffset = 0`
	`dataLength = -1`
	`updateRecord()`
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

20	**dataOffset + dataLength > data.length**
	`dataOffset = 10`
	`dataLength = 11`
	`updateRecord()`
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

21	**EF is neither Cyclic nor Linear Fixed**
	`1 - fid = DFSIMTEST`
	`select()`
	`2 - fid = EFTNR`
	`select()`
	`3 - dataOffset = 0`
	`dataLength = 4`
	`updateRecord()`
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		3 - Shall throw sim.access.SIMViewException with reason code FILE_INCONSISTENT.	

Id	Description	API Expectation	APDU Expectation

22	**Access condition not fulfilled**
	`1 - fid = EFCNU`
	`select()`
	`2 - recOffset = 0`
	`dataOffset = 0`
	`dataLength = 1`
	`mode = REC_ACC_MODE_PREVIOUS`
	`updateRecord()`
	`3 - fid = EFLNU`
	`select()`
	`4 - recNumber = 1`
	`mode = REC_ACC_MODE_ABSOLUTE_CURRENT`
	`recOffset = 0`
	`dataOffset = 0`
	`dataLength = 1`
	`updateRecord()`
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code AC_NOT_FULFILLED.
		3 - No exception shall be thrown.
		4 - Shall throw sim.access.SIMViewException with reason code AC_NOT_FULFILLED.	

23	**EF is invalidated**
	`1 - fid = EFCNR`
	`mode = REC_ACC_MODE_PREVIOUS`
	`invalidate()`
	`2 - updateRecord()`
	`3 - rehabilitate()`
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code INVALIDATION_STATUS_CONTRADICTION.
		3 - No exception shall be thrown.	

6.1.1.8.4 Test Coverage

CRR Number	Test Case Number
N1	2, 3,4, 5, 7, 8, 10
N2	2, 3
N3	5, 6
N4	7, 8, 9, 10
P1	11
P2	12
P3	6
P4	9
P5	13
P6	14
P7	15
P8	16
P9	17
P10	18
P11	19
P12	20
C1	1
C2	21
C3	22
C4	23
C5, C6	Not Tested

6.1.1.9 Method seek

Test Area Reference: API_1_SVW_SEEKB_BSS

6.1.1.9.1 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
```

6.1.1.9.1.1 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. This will be tested during the testing of the framework.

- 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, 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.

6.1.1.9.1.2 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 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.

- CRRP6: If pattOffset plus pattLength is greater than the length of the pattern array patt.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

6.1.1.9.1.3 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 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.

6.1.1.9.2 Test Suite Files

Additional requirements for the GSM personalization: None

Test Script:	API_1_SVW_SEEKB_BSS_1.scr
Test Applet:	API_1_SVW_SEEKB_BSS_1.java
Load Script:	API_1_SVW_SEEKB_BSS_1.ldr
Cleanup Script:	API_1_SVW_SEEKB_BSS_1.ldr
Parameter File:	API_1_SVW_SEEKB_BSS_1.par

6.1.1.9.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored.	

1	**No EF selected**
	`Byte[] patt = new byte[20]`
	`pattOffset = 0`
	`pattLength = 10`
	`mode = SEEK_FROM_BEGINNING_FORWARD`
	`seek()`
		Shall throw sim.access.SIMViewException with reason code NO_EF_SELECTED.	

2	**Pattern not Found**
	`1 - fid = DFSIMTEST`
	`select()`
	`2 - fid = EFLARU`
	`select()`
	`3 - patt[0] = 'DA'`
	`pattOffset = 0`
	`pattLength = 1`
	`mode = SEEK_FROM_BEGINNING_FORWARD`
	`seek()`
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		3 - Shall throw sim.access.SIMViewException with reason code PATTERN_NOT_FOUND.	

3	**Seek from Beginning Forward**
	`patt[0:2] = '55'`
	`pattOffset = 0`
	`pattLength = 3`
	`mode = SEEK_FROM_BEGINNING_FORWARD`
	`seek()`
		No exception shall be thrown. Shall return 1	

4	**Seek from End Backward**
	`patt[0:2] = '55'`
	`pattOffset = 0`
	`pattLength = 3`
	`mode = SEEK_FROM_END_BACKWARD`
	`seek()`
		No exception shall be thrown. Shall return 1	

5	**Seek from Next Forward**
	`patt[0:2] = 'AA'`
	`pattOffset = 0`
	`pattLength = 3`
	`mode = SEEK_FROM_NEXT_FORWARD`
	`seek()`
		No exception shall be thrown. Shall return 2	

6	**Last Record, Seek from Next Forward**
	`mode = SEEK_FROM_NEXT_FORWARD`
	`seek()`
		Shall throw sim.access.SIMViewException with reason code PATTERN_NOT_FOUND.	

7	**Seek from Previous Backward**
	`patt[0:2] = '55'`
	`pattOffset = 0`
	`pattLength = 3`
	`mode = SEEK_FROM_PREVIOUS_BACKWARD`
	`seek()`
		No exception shall be thrown. Shall return 1	

Id	Description	API Expectation	APDU Expectation
8	**First Record, Seek from Previous Backward** SEEK_FROM_PREVIOUS_BACKWARD seek()	Shall throw sim.access.SIMViewException with reason code PATTERN_NOT_FOUND.	

9	**Pattern not Found (out of reach)**
	patt[0:2] = '55'
	pattOffset = 0
	pattLength = 3
	mode = SEEK_FROM_NEXT_FORWARD
	seek()
		Shall throw sim.access.SIMViewException with reason code PATTERN_NOT_FOUND.	

10	**Invalid mode** 1 - mode = 4 seek() 2 - mode = -1 seek()	1 - Shall throw sim.access.SIMViewException with reason code INVALID_MODE 2 - Shall throw sim.access.SIMViewException with reason code INVALID_MODE	
11	**patt is null** byte[] nullBuffer = null mode = SEEK_FROM_BEGINNING_FORWARD seek ()	Shall throw java.lang.NullPointerException.	

12	**pattOffset < 0**
	patt[0:2] = '55'
	pattOffset = -1
	pattLength = 3
	mode = SEEK_FROM_BEGINNING_FORWARD
	seek()
		Shall throw java.lang.ArrayIndexOutOfBoundsException	

13	**pattLength < 0**
	patt[0:2] = '55'
	pattOffset = 0
	pattLength = -1
	mode = SEEK_FROM_BEGINNING_FORWARD
	seek()
		Shall throw java.lang.ArrayIndexOutOfBoundsException	

14	**pattLength > size of record**
	patt[0:4] = '55'
	pattOffset = 0
	pattLength = 4
	mode = SEEK_FROM_BEGINNING_FORWARD
	seek()
		Shall throw sim.access.SIMViewException with reason code OUT_OF_RECORD_BOUNDARIES	

15	**pattOffset + pattLength > patt.length**
	patt[0:2] = '55'
	pattOffset = 1
	pattLength = 3
	mode = SEEK_FROM_BEGINNING_FORWARD
	seek()
		Shall throw java.lang.ArrayIndexOutOfBoundsException	

16	**EF is not Linear Fixed**
	1 - fid = EFTNU
	select()
	2 - pattOffset = 0
	pattLength = 3
	mode = SEEK_FROM_BEGINNING_FORWARD
	seek()
	3 - fid = EFCNU
	select()
	seek()
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code FILE_INCONSISTENT
		3 - Shall throw sim.access.SIMViewException with reason code FILE_INCONSISTENT	

17	**Access condition not fulfilled**
	1 - fid = EFLNR
	select()
	2 - patt[0] = '55'
	pattOffset = 0
	pattLength = 1
	mode = SEEK_FROM_BEGINNING_FORWARD
	seek()
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code AC_NOT_FULFILLED.	

18	**EF is invalidated**
	1 - fid = EFLARU
	select()
	2 - invalidate()
	3 - patt[0] = '55
	pattOffset = 0
	pattLength = 1
	mode = SEEK_FROM_BEGINNING_FORWARD
	seek()
	4 - rehabilitate()
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		3 - Shall throw sim.access.SIMViewException with reason code INVALIDATION_STATUS_CONTRADICTION.
		4 - No exception shall be thrown.	

6.1.1.9.4 Test Coverage

CRR Number	Test Case Number
N1	2, 3 - 6, 7
N2	3
N3	4
N4	5
N5	7
N6	2, 6, 8, 9
N7	6
N8	8
P1	10
P2	11
P3	12
P4	13
P5	14
P6	15
C1	1
C2	16
C3	17
C4	18
C5, C6	Not Tested

6.1.1.10 Method increase

Test Area Reference: API_1_SVW_INCR_BS_BS

6.1.1.10.1 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
```

6.1.1.10.1.1 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. The number of bytes of valid data in resp is returned.

6.1.1.10.1.2 Parameter errors

- CRRP1: If the array incr is null, an instance of NullPointerException shall be thrown.
- CRRP2: If incrOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP3: If incrOffset plus the value 3, is greater than the length of the array incr.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP4: If the result of the addition is greater than the maximum value of the record (represented by all bytes set to 'FF'), an instance of SIMViewException shall be thrown. The reason code shall be SIMViewException.MAX_VALUE_REACHED.
- CRRP5: If the array resp is null, an instance of NullPointerException shall be thrown.
- CRRP6: If respOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP7: 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.

6.1.1.10.1.3 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 (TS 51.011: 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.

6.1.1.10.2 Test Suite Files

Additional requirements for the GSM personalization: None

Test Script:	API_1_SVW_INCR_BS_BS_1.scr
Test Applet:	API_1_SVW_INCR_BS_BS_1.java
Load Script:	API_1_SVW_INCR_BS_BS_1.ldr
Cleanup Script:	API_1_SVW_INCR_BS_BS_1.clr
Parameter File:	API_1_SVW_INCR_BS_BS_1.par

6.1.1.10.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored.	

1	**No EF selected**
	`byte[] incr = new byte[4]`
	`byte[] resp = new byte[4]`
	`incrOffset = 0`
	`respOffset = 0`
	`increase()`
		Shall throw sim.access.SIMViewException with reason code NO_EF_SELECTED.	

2	**Increase , verify response**
	`1 - fid = DFSIMTEST`
	`select()`
	`2 - fid = EFCARU`
	`select()`
	`3 - //Set both records to 00 00 00`
	`mode = REC_ACC_MODE_PREVIOUS`
	`data[0:3] = 0`
	`dataOffset = 0`
	`dataLength = 3`
	`updateRecord()`
	`updateRecord()`
	`4 - incrOffset = 0`
	`incr[2] = 1`
	`respOffset = 0`
	`increase()`
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		3 - No exception shall be thrown.
		4 - No exception shall be thrown.
		resp[] shall contain {0,0,1,0}.	

Id	Description	API Expectation	APDU Expectation

3	**Increase, verify file**
	`1 - incrOffset = 1`
	`incr[2] = 0, incr[3] = 2`
	`respOffset = 1`
	`increase()`
	`2 - resp[3] = 0`
	`recNumber = 0`
	`mode = REC_ACC_MODE_ABSOLUTE_CURRENT`
	`recOffset = 0`
	`respOffset = 0`
	`respLength = 0`
	`readRecord()`
		1 - No exception shall be thrown.
		resp[] shall contain {0,0,0,3}.
		2 - No exception shall be thrown.
		resp[] shall contain {0,0,3,0}.	

4	**incr is null**
	`byte[] nullBuffer = null`
	`incrOffset = 0`
	`respOffset = 0`
	`increase()`
		Shall throw java.lang.NullPointerException.	

5	**incrOffset < 0**
	`incrOffset = -1`
	`respOffset = 0`
	`increase()`
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

6	**incrOffset + 3 > incr.length**
	`incrOffset = 2`
	`respOffset = 0`
	`increase()`
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

7	**Reach Maximum Value**
	`incr[0] = incr[1] = incr[2] = 'FF'`
	`incrOffset = 0`
	`respOffset = 0`
	`increase()`
		Shall throw sim.access.SIMViewException with reason code MAX_VALUE_REACHED.	

8	**resp is null**
	`incr[0] = incr[1] = 0x00'`
	`incr[2] = '02'`
	`incrOffset = 0`
	`byte[] respNull = null`
	`respOffset = 0`
	`increase()`
		Shall throw java.lang.NullPointerException.	

9	**respOffset < 0**
	`incrOffset = 0`
	`respOffset = -1`
	`increase()`
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

10	**respOffset + recordLength > resp.length**
	`incrOffset = 0`
	`respOffset = 2`
	`increase()`
		Shall throw java.lang. ArrayIndexOutOfBoundsException.	

11	**EF is not Cyclic**
	`1 - fid = EFTARU`
	`select()`
	`2 - incrOffset = 0`
	`respOffset = 0`
	`increase()`
	`3 - fid = EFLARU`
	`select()`
	`4 - incrOffset = 0`
	`respOffset = 0`
	`increase()`
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code FILE_INCONSISTENT.
		3 - No exception shall be thrown.
		4 - Shall throw sim.access.SIMViewException with reason code FILE_INCONSISTENT.	

12	**Access condition not fulfilled**
	`1 - fid = EFCNIC`
	`select()`
	`2 - incrOffset = 0`
	`respOffset = 0`
	`increase()`
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code AC_NOT_FULFILLED.	

13	**EF is invalidated**
	`1 - fid = EFCARU`
	`select()`
	`2 - invalidate()`
	`3 - incrOffset = 0`
	`respOffset = 0`
	`increase()`
	`4 - rehabilitate()`
		1 - No exception shall be thrown.
		2 - No exception shall be thrown.
		3 - Shall throw sim.access.SIMViewException with reason code INVALIDATION_STATUS_CONTRADICTION.
		4 - No exception shall be thrown.	

Id	Description	API Expectation	APDU Expectation

14	**Check increase not allowed from FCI**
	`1 - fciOffset = 0`
	`fciLength = 8`
	`select (FID_EF_CINA, fci...)`
	`Verify FCI byte 8 (fci[7])`
	`2 - incrOffset = 0`
	`respOffset = 0`
	`increase()`
		1 - No exception shall be thrown. Bit 7 of resp[7] shall not be set (0), indicating that increase is not allowed.
		2 - Shall throw sim.access.SIMViewException with reason code FILE_INCONSISTENT	

6.1.1.10.4 Test Coverage

CRR Number	Test Case Number
N1	2, 3
P1	4
P2	5
P3	6
P4	7
P5	8
P6	9
P7	10
C1	1
C2	11
C3	14
C4	12
C5	13
C6, C7	Not Tested

6.1.1.11 Method invalidate

Test Area Reference: API_1_SVW_INVL

6.1.1.11.1 Conformance Requirements

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

```
public void invalidate()
                throws SIMViewException
```

6.1.1.11.1.1 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.

6.1.1.11.1.2 Parameter errors

No requirements.

6.1.1.11.1.3 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.

6.1.1.11.2 Test Suite Files

Additional requirements for the GSM personalization: None

Test Script:	API_1_SVW_INVL_1.scr
Test Applet:	API_1_SVW_INVL_1.java
Load Script:	API_1_SVW_INVL_1.ldr
Cleanup Script:	API_1_SVW_INVL_1.clr
Parameter File:	API_1_SVW_INVL_1.par

6.1.1.11.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored.	
1	**No EF is selected** 1 - invalidate()	1 - Shall throw sim.access.SIMViewException with reason code NO_EF_SELECTED.	
2	**Invalidate EF** 1 - fid = DFSIMTEST select() 2 - fid = EFTNR select() 3 - invalidate() 4 - rehabilitate()	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - No exception shall be thrown. 4 - No exception shall be thrown.	
3	**Access condition not fulfilled** 1 - fid = EFCNIV select() 2 - invalidate()	1 - No exception shall be thrown. 2 - Shall throw sim.access.SIMViewException with reason code AC_NOT_FULFILLED.	
4	**EF is already invalidated** 1 - fid = EFTNR select() 2 - invalidate() 3 - invalidate()	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - Shall throw sim.access.SIMViewException with reason code INVALIDATION_STATUS_CONTRADICTION.	

6.1.1.11.4 Test Coverage

CRR number	Test Case Number
N1	2
C1	1
C2	3
C3	4
C4, C5	Not Tested

6.1.1.12 Method rehabilitate

Test Area Reference: API_1_SVW_REHA

6.1.1.12.1 Conformance Requirements

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

```
public void rehabilitate()
                  throws SIMViewException
```

6.1.1.12.1.1 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.

6.1.1.12.1.2 Parameter errors

No requirements.

6.1.1.12.1.3 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 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.

6.1.1.12.2 Test Suite Files

Additional requirements for the GSM personalization: None

Test Script:	API_1_SVW_REHA_1.scr
Test Applet:	API_1_SVW_REHA_1.java
Load Script:	API_1_SVW_REHA_1.ldr
Cleanup Script:	API_1_SVW_REHA_1.clr
Parameter File:	API_1_SVW_REHA_1.par

6.1.1.12.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored.	
1	**No EF is selected** `1 - rehabilitate()`	1 - Shall throw sim.access.SIMViewException with reason code NO_EF_SELECTED.	

Id	Description	API Expectation	APDU Expectation

2	**Rehabilitate invalidated File**
	`1 - fid = DFSIMTEST`
	`select()`
	`2 - fid = EFCNR`
	`select()`
	`3 - invalidate()`
	`4 - rehabilitate()`
	`5 - byte[] incr = new byte[3] = {0,0,1}`
	`incrOffset = 0`
	`byte[] resp = new byte[1] = 1`
	`respOffset = 0`
	`increase()`
		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.
		resp[] shall contain {0,0,1}.	

3	**Access condition not fulfilled**
	`1 - fid = EFCNRH`
	`select()`
	`2 - rehabilitate()`
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code AC_NOT_FULFILLED..	

4	**Rehabilitate validated File**
	`1 - fid = EFCNR`
	`select()`
	`2 - rehabilitate()`
		1 - No exception shall be thrown.
		2 - Shall throw sim.access.SIMViewException with reason code INVALIDATION_STATUS_CONTRADICTION.	

6.1.1.12.4 Test Coverage

CRR number	Test Case Number
N1	2
C1	1
C2	3
C3	4
C4, C5	Not Tested

6.1.2 Class SIMSystem

6.1.2.1 Method getTheSIMView

Test Area Reference: API_1_SSY_GETS

6.1.2.1.1 Conformance Requirement:

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

```
public static SIMView getTheSIMView()
```

6.1.2.1.1.1 Normal execution

- CRRN1: returns a reference to class which implements the SIMView interface.

6.1.2.1.1.2 Parameters error

No requirements.

6.1.2.1.1.3 Context errors

No requirements.

6.1.2.1.2 Test suite files

No additional requirements for the GSM personalization:

Test Script:	API_1_SSY_GETS_1.scr
Test Applet:	API_1_SSY_GETS_1.java
Load Script:	API_1_SSY_GETS_1.ldr
Cleanup Script:	API_1_SSY_GETS_1.clr
Parameter File:	API_1_SSY_GETS_1.par

6.1.2.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**reference not equal null after execute**	The returned reference shall be not null after execute	
2	**reference to the GSM interface**	Returned a reference to the GSM interface	

6.1.2.1.4 Test Coverage

CRR number	Test case number
N1	1,2

6.1.3 Class SIMViewException

6.1.3.1 Method throwIt

Test Area Reference: API_1_SVE_THITS

6.1.3.1.1 Conformance Requirement:

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

```
public static void throwIt(short reason)
                    throws SIMViewException
```

6.1.3.1.1.1 Normal execution

- CRRN1: Throws the JCRE instance of SIMViewException with the specified reason.
- CRRN2: Extends javacard.framework.CardRuntimeException.

6.1.3.1.1.2 Parameter errors

No requirements.

6.1.3.1.1.3 Context errors

No requirements.

6.1.3.1.2 Test Suite Files

No additional requirements for the GSM personalization

Test Script:	API_1_SVE_THITS_1.scr
Test Applet:	API_1_SVE_THITS_1.java

Load Script:	API_1_SVE_THITS_1.ldr
Cleanup Script:	API_1_SVE_THITS_1.clr
Parameter File:	API_1_SVE_THITS_1.par

6.1.3.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Throws the JCRE instance of SIMViewException with the specified reason**	Reason = 0	
2	**Throws the JCRE instance of SIMViewException with the specified reason**	Reason = 1	
3	**Throws the JCRE instance of SIMViewException with the specified reason**	Reason = 15	
4	**SIMViewException extends javacard.framework.CardRuntimeException**	Reason = 0	
5	**SIMViewException extends javacard.framework.CardRuntimeException**	Reason = 1	
6	**SIMViewException extends javacard.framework.CardRuntimeException**	Reason = 15	

6.1.3.1.4 Test Coverage

CRR number	Test case number
N1	1,2,3
N2	4,5,6

6.1.3.2 Constructor

Test Area Reference: API_1_SVE_COORS

6.1.3.2.1 Conformance Requirement:

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

```
public SIMViewException(short reason)
                 throws SIMViewException
```

6.1.3.2.1.1 Normal execution

- CRRN1: Construct a SIMViewException with the specified reason.

6.1.3.2.1.2 Parameters error

No requirements.

6.1.3.2.1.3 Context errors

No requirements.

6.1.3.2.2 Test suite files

No additional requirements for the GSM personalization

Test Script:	API_1_SVE_COORS_1.scr
Test Applet:	API_1_SVE_COORS_1.java
Load Script:	API_1_SVE_COORS.ldr
Cleanup Script:	API_1_SVE_COORS.clr

Parameter File: API_1_SVE_COORS.par

6.1.3.2.3 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)	

6.1.3.2.4 Test Coverage

CRR number	Test case number
N1	1

6.1.3.3 Reason Codes

Test Area Reference: API_1_SVE_CONS

6.1.3.3.1 Conformance Requirement:

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

6.1.3.3.1.1 Normal execution

- CRRN1: The Constants of the class SIMViewException shall all have the same name and value defined in the 3GPP TS 43.019 [7].
- CRRN2: Constructs SIMViewException a Exception with the specified reason.

6.1.3.3.1.2 Parameters error

No requirements.

6.1.3.3.1.3 Context errors

No requirements.

6.1.3.3.2 Test suite files

None.

6.1.3.3.3 Test Procedure

The constants in Java are resolved at compilation time, therefore a runtime test is not useful. No test of constants will be performed

6.2 Package sim.toolkit

6.2.1 Interface ToolkitConstants

6.2.1.1 Constants

Test Area Reference: API_2_TKC_CONS

6.2.1.1.1 Conformance Requirement

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

6.2.1.1.1.1 Normal execution

- CRRN1: The Toolkit Constants shall all have the same name and value as defined in 3GPP TS 43.019 [7].

6.2.1.1.1.2 Parameters error

No requirements.

6.2.1.1.1.3 Context errors

No requirements.

6.2.1.1.2 Test suite files

None.

6.2.1.1.3 Test Procedure

The constants in Java are resolved at compilation time, therefore a runtime test is not useful. No test of constants will be performed.

6.2.2 Interface ToolkitInterface

6.2.2.1 Method processToolkit

Test Area Reference: API_2_TKI_PRTKB

6.2.2.1.1 Conformance Requirement:

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

```
public void processToolkit(byte event)
                    throws ToolkitException
```

6.2.2.1.1.1 Normal execution

- CRRN1: This interface must be implemented by a Toolkit applet (which extends the javacard.framework.Applet class) so that it can be triggered by the Toolkit Handler according to the registration information.
- CRRN2: The Toolkit applet will have to implement the processToolkit shared method so that the following events can be notified:

Event	Description
EVENT_PROFILE_DOWNLOAD	Terminal Profile command reception
EVENT_FORMATTED_SMS_PP_ENV	Formatted envelope SMS-PP Data Download reception
EVENT_FORMATTED_SMS_PP_UPD	Formatted Update Record EF SMS
EVENT_FORMATTED_SMS_CB	Formatted envelope Cell Broadcast Data Download command reception
EVENT_UNFORMATTED_SMS_PP_ENV	Unformatted Envelope SMS-PP Data Download reception
EVENT_UNFORMATTED_SMS_PP_UPD	Unformatted Update Record EF SMS
EVENT_UNFORMATTED_SMS_CB	Unformatted Cell Broadcast Data Download command reception
EVENT_MENU_SELECTION	Envelope Menu Selection command reception
EVENT_MENU_SELECTION_HELP_REQUEST	Envelope Menu Selection Help Request command reception
EVENT_CALL_CONTROL_BY_SIM	Envelope Call Control by SIM command reception

Event	Description
`EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM`	Envelope MO Short Message Control by SIM command reception
`EVENT_TIMER_EXPIRATION`	Envelope Timer Expiration
`EVENT_EVENT_DOWNLOAD_MT_CALL`	Envelope Event Download - MT call
`EVENT_EVENT_DOWNLOAD_CALL_CONNECTED`	Envelope Event Download - Call connected
`EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED`	Event Download - Call disconnected
`EVENT_EVENT_DOWNLOAD_LOCATION_STATUS`	Envelope Event Download - Location status
`EVENT_EVENT_DOWNLOAD_USER_ACTIVITY`	Envelope Event Download - User activity
`EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE`	Envelope Event Download - Idle screen available
`EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS`	Envelope Event Download - Card Reader Status
`EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION`	Envelope Event Download - Language Selection
`EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION`	Envelope Event Download - Browser Termination
`EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE`	Envelope Event Download - Data Available
`EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS`	Envelope Event Download - Channel Status
`EVENT_FIRST_COMMAND_AFTER_SELECT`	First command performed after select GSM application or ATR
`EVENT_STATUS_COMMAND`	Status APDU command event
`EVENT_UNRECOGNIZED_ENVELOPE`	Unrecognized Envelope command reception

6.2.2.1.1.2 Parameters error

No requirements.

6.2.2.1.1.3 Context errors

No requirements.

6.2.2.1.2 Test suite files

The method is tested in the Framework.

6.2.2.1.3 Test Coverage

CRR number	Test case number
N1	Tested in Framework
N2	Tested in Framework

6.2.3 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.

6.2.4 Class EnvelopeHandler

6.2.4.1 Method getEnvelopeTag

Test Area Reference: API_2_ENH_GENT

6.2.4.1.1 Conformance Requirement:

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

```
public byte getEnvelopeTag()
```

6.2.4.1.1.1 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 available.

6.2.4.1.1.2 Parameters error

No requirements.

6.2.4.1.1.3 Context errors

No requirements.

6.2.4.1.2 Test suite files

Test Script:	API_2_ENH_GENT_1.scr
Test Applet:	API_2_ENH_GENT_1.java
Load Script:	API_2_ENH_GENT_1.ldr
Cleanup Script:	API_2_ENH_GENT_1.clr
Parameter File:	API_2_ENH_GENT_1.par

6.2.4.1.3 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	

6.2.4.1.4 Test Coverage

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

6.2.4.2 Method getItemIdentifier

Test Area Reference: API_2_ENH_GIID

6.2.4.2.1 Conformance Requirement:

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

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

6.2.4.2.1.1 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.

6.2.4.2.1.2 Parameters error

No requirements.

6.2.4.2.1.3 Context errors

- CRRC1: The method shall throw ToolkitException (UNAVAILABLE_ELEMENT) if the item identifier TLV is not present.
- CRRC2: The method shall throw ToolkitException (OUT_OF_TLV_BOUNDARIES) if the item identifier byte is missing in the Item Identifier Simple TLV.

6.2.4.2.2 Test suite files

Test Script:	API_2_ENH_GIID_1.scr
Test Applet:	API_2_ENH_GIID_1.java
Load Script:	API_2_ENH_GIID_1.ldr
Cleanup Script:	API_2_ENH_GIID_1.clr
Parameter File:	API_2_ENH_GIID_1.par

6.2.4.2.3 Test Procedure

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

6.2.4.2.4 Test Coverage

CRR number	Test case number
N1	1, 2, 3
N2	2, 3
N3	4
N4	6
C1	5
C2	7

6.2.4.3 Method getSecuredDataLength

Test Area Reference: API_2_ENH_GSDL

6.2.4.3.1 Conformance Requirement:

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

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

6.2.4.3.1.1 Normal execution

- CRRN1: The method shall return the length of the Secured Data from the Command Packet in the SMS TPDU (simple or concatenated) or Cell Broadcast Page Simple TLV contained in the Envelope handler.
- CRRN2: The length is from the first SMS TPDU TLV or Cell Broadcast Page Simple 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 3GPP TS 23.048 [8].
- 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 3GPP TS 23.048 [8].
- CRRN6: The method can be used if the event is EVENT_FORMATTED_SMS_CB and if the Cell Broadcast Page is formatted according to 3GPP TS 23.048 [8].
- CRRN7: If the method is successful and if the event is EVENT_FORMATTED_SMS_PP_ENV, the selected TLV should be the SMS TPDU TLV.
- CRRN8: If the method is successful and if the event is EVENT_FORMATTED_SMS_PP_UPD, the selected TLV should be the SMS TPDU TLV.
- CRRN9: If the method is successful and if the event is EVENT_FORMATTED_SMS_CB, the selected TLV should be the Cell Broadcast Page TLV.

6.2.4.3.1.2 Parameters error

No requirements.

6.2.4.3.1.3 Context errors

- CRRC1: The method shall thrown ToolkitException (UNAVAILABLE_ELEMENT) in case of unavailable SMS TPDU TLV element or Cell Broadcast Page Simple TLV.
- CRRC2: The method shall thrown ToolkitException (UNAVAILABLE_ELEMENT) in case of wrong data format.

6.2.4.3.2 Test suite files

Specific triggering:

- FORMATTED SMS CB.
- UNFORMATTED SMS CB.
- FORMATTED SMS PP UPD.
- UNFORMATED SMS PP ENV.

```
-   For Formatted triggering if CC/RC/DS is used, the security parameters are the one used for
    downloading applications.
```

Test Script:	API_2_ENH_GSDL_1.scr
Test Applet:	API_2_ENH_GSDL_1.java
Load Script:	API_2_ENH_GSDL_1.ldr
Cleanup Script:	API_2_ENH_GSDL_1.clr

Parameter File: API_2_ENH_GSDL_1.par

6.2.4.3.3 Test procedure

Id	Description	API Expectation	APDU Expectation
	FORMATTED SMS PP ENV Triggering		

Id	Description	API Expectation	APDU Expectation
1	Test with FORMATTED_SMS_PP_ENV and TP-OA length of 2	Returns 0x002A	
2	Test with TP-OA length of 6	Returns 0x002A	
3	Test with TP-OA length of 12	Returns 0x002A	
4	Test with RC/CC/DS length of 0	Returns 0x0010	
5	Test with RC/CC/DS length of 8	Returns 0x0010	
6	Test with PCNTR = 0	Returns 0x0010	
7	Test with PCNTR = 7	Returns 0x0005	
8	Test with Secured Data Length = 00	Returns 0x0000	
9	Test with Secured Data Length = 0x33	Returns 0x0033	
10	Test with Secured Data Length = 0x6C (UDL = 0x7F)	Returns 0x006C	
11	Test with Secured Data Length = 0x6D (UDL = 0x80)	Returns 0x006D	
12	Test with Secured Data Length = maximum length for one envelope : 0x79 (UDL = 0x8C)	Returns 0x0079	
13	Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV and inside two different secured data lengths: 5 and 10	Returns 0x0005	
14	Test with secured data length = 0x7F (2 concatenated envelopes are needed)	Returns 0x007F	
15	Test with secured data length = 0x80 (2 concatenated envelopes are needed)	Returns 0x0080	
16	Test with secured data length = maximum length for 2 concatenated envelopes : 0xFA	Returns 0x00FA	
17	Test with FORMATTED_SMS_PP_ENV Verify after call of the method the current TLV is the TPDU TLV: findTLV device identities, getSecuredDataLength and then getValueByte to verify that the current TLV is the TPDU TLV	getValueByte returns 0x0040	
	FORMATTED SMS PP UPD Triggering		
18	Same test as 1 but with FORMATTED_SMS_PP_UPD	Returns 0x002A	
19	Same test as 2 but with FORMATTED_SMS_PP_UPD	Returns 0x002A	
20	Same test as 3 but with FORMATTED_SMS_PP_UPD	Returns 0x002A	
21	Same test as 4 but with FORMATTED_SMS_PP_UPD	Returns 0x0010	
22	Same test as 5 but with FORMATTED_SMS_PP_UPD	Returns 0x0010	
23	Same test as 6 but with FORMATTED_SMS_PP_UPD	Returns 0x0010	
24	Same test as 7 but with FORMATTED_SMS_PP_UPD	Returns 0x0005	
25	Same test as 8 but with FORMATTED_SMS_PP_UPD	Returns 0x0000	
26	Same test as 9 but with FORMATTED_SMS_PP_UPD	Returns 0x0033	
27	Same test as 10 but with FORMATTED_SMS_PP_UPD	Returns 0x006C	
28	Same test as 11 but with FORMATTED_SMS_PP_UPD	Returns 0x006D	
29	Same test as 12 but with FORMATTED_SMS_PP_UPD	Returns 0x0079	
30	Same test as 13 but with FORMATTED_SMS_PP_UPD	Returns 0x0005	
31	Test with secured data length = 0x7F (2 concatenated envelopes are needed)	Returns 0x007F	
32	Test with secured data length = 0x80 (2 concatenated envelopes are needed)	Returns 0x0080	
33	Test with secured data length = maximum length for 2 concatenated envelopes : 0xFA	Returns 0x00FA	
34	Test with FORMATTED_SMS_PP_UPD Verify after call of the method the current TLV is	getValueByte returns 0x0040	

Id	Description	API Expectation	APDU Expectation
	the TPDU TLV: findTLV device identities, getSecuredDataLength and then getValueByte to verify that the current TLV is the TPDU TLV		
	FORMATTED SMS CB Triggering		
35	Same test as 4 but with FORMATTED_SMS_CB	Returns 0x0010	
36	Same test as 5 but with FORMATTED_SMS_CB	Returns 0x0010	
37	Same test as 6 but with FORMATTED_SMS_CB	Returns 0x0010	
38	Same test as 7 but with FORMATTED_SMS_CB	Returns 0x0005	
39	Same test as 8 but with FORMATTED_SMS_CB	Returns 0x0000	
40	Same test as 9 but with FORMATTED_SMS_CB	Returns 0x0033	
41	Same test as 12 but with maximum secured data length: 0x42, and FORMATTED_SMS_CB	Returns 0x0042	
42	Test with FORMATTED_SMS_CB Verify after call of the method the current TLV is the Cell Broadcast Page TLV: findTLV device identities, getSecuredDataLength and then getValueByte to verify that the current TLV is the Cell Broadcast Page TLV	getValueByte returns 0x00	
	Error tests		
43	Send an envelope SMS CB, getSecuredDataLength	ToolkitException UNAVAILABLE_ELEMENT	
44	Send an envelope SMS PP unformatted	ToolkitException UNAVAILABLE_ELEMENT	

6.2.4.3.4 Test Coverage

CRR number	Test case number
N1	1 to 42
N2	13, 30
N3	6, 7, 23, 24, 37, 38
N4	1 to 17
N5	18 to 34
N6	35 to 42
N7	17
N8	34
N9	42
C1	43
C2	44

6.2.4.4 Method getSecuredDataOffset

Test Area Reference: API_2_ENH_GSDO

6.2.4.4.1 Conformance Requirement:

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

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

6.2.4.4.1.1 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 3GPP TS 23.048 [8].
- 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 3GPP TS 23.048 [8].
- CRRN5: The method can be used if the event is EVENT_FORMATTED_SMS_CB and if the Cell Broadcast Page is formatted according to 3GPP TS 23.048 [8].
- CRRN6: If the method is successful and if the event is EVENT_FORMATTED_SMS_PP_ENV, the selected TLV should be the SMS TPDU TLV.
- CRRN7: If the method is successful and if the event is EVENT_FORMATTED_SMS_PP_UPD, the selected TLV should be the SMS TPDU TLV.
- CRRN8: If the method is successful and if the event is EVENT_FORMATTED_SMS_CB, the selected TLV should be the Cell Broadcast Page TLV.
- CRNN9: If the Secured Data length is zero the value returned shall be the offset of the first byte following the 3GPP TS 23.048 [8] Command Packet structure.

6.2.4.4.1.2 Parameters error

No requirements.

6.2.4.4.1.3 Context errors

- 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 wrong data format.

6.2.4.4.2 Test suite files

Specific triggering:

- FORMATTED SMS CB.
- UNFORMATTED SMS CB.
- FORMATTED SMS PP UPD.
- UNFORMATED SMS PP ENV.
- For Formatted triggering if CC/RC/DS is used, the security parameters are the one used for downloading applications.

Test Script:	API_2_ENH_GSDO_1.scr
Test Applet:	API_2_ENH_GSDO_1.java
Load Script:	API_2_ENH_GSDO_1.ldr
Cleanup Script:	API_2_ENH_GSDO_1.clr
Parameter File:	API_2_ENH_GSDO_1.par

6.2.4.4.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
	FORMATTED SMS PP ENV triggering		

Id	Description	API Expectation	APDU Expectation
1	Test with TP-OA length of 2 and RC/CC/DS length is 0	Returns 0x21	
2	Test with TP-OA length of 6 and RC/CC/DS length is 0	Returns 0x23	
3	Test with TP-OA length of 12 and RC/CC/DS length is 0	Returns 0x26	
4	Test with RC/CC/DS length of 0 and TP-OA length is 2	Returns 0x21	
5	Test with RC/CC/DS length of 8 and TP-OA length is 2	Returns 0x29	
6	Send a SMS PP with 2 TPDU TLV and inside two different secured data offsets	Returns 0x24 (the first offset)	
7	Same test as 1 but without any secured data	Returns 0x21	
8	Test with FORMATTED_SMS_PP ENV Verify after call of the method the current TLV is the TPDU TLV: findTLV device identities, getSecuredDataOffset and then getValueByte to verify that the current TLV is the TPDU TLV	Returns 0x40	
9	Same test as 1, but with a concatenated SMS (2 Short Messages and maximum Secured Data Length = 0x00FA)	Returns 0x21	
	FORMATTED SMS PP UPR triggering		
10	Same test as 1 but with FORMATTED_SMS_PP_UPD	Returns 0x21	
11	Same test as 2 but with FORMATTED_SMS_PP_UPD	Returns 0x23	
12	Same test as 3 but with FORMATTED_SMS_PP_UPD	Returns 0x26	
13	Same test as 4 but with FORMATTED_SMS_PP_UPD	Returns 0x21	
14	Same test as 5 but with FORMATTED_SMS_PP_UPD	Returns 0x29	
15	Same test as 6 but with FORMATTED_SMS_PP_UPD	Returns 0x24 (the first offset)	
16	Same test as 7 but with FORMATTED_SMS_PP_UPD	Returns 0x21	
17	Test with FORMATTED_SMS_PP UPD Verify after call of the method the current TLV is the TPDU TLV: findTLV device identities, getSecuredDataOffset and then getValueByte to verify that the current TLV is the TPDU TLV	Returns 0x40	
18	Same test as 10, but with a concatenated SMS (2 Short Messages and maximum Secured Data Length = 0x00FA)	Returns 0x21	
	FORMATTED SMS CB triggering		
19	Same test as 4 but with FORMATTED_SMS_CB	Returns 0x16	
20	Same test as 5 but with FORMATTED_SMS_CB	Returns 0x1E	
21	Same test as 7 but with FORMATTED_SMS_CB	Returns 0x16	
22	Test with FORMATTED_SMS_CB Verify after call of the method the current TLV is the Cell Broadcast Page TLV: findTLV device identities, getSecuredDataOffset and then getValueByte to verify that the current TLV is the Cell Broadcast Page TLV	Returns 0x00	
	UNFORMATTED Triggering		
23	Send an UNFORMATTED SMS CB envelope, getSecuredDataOffset	ToolkitException UNAVAILABLE_ELEMENT	
24	Send an UNFORMATTED SMS PP envelope, getSecuredDataOffset	ToolkitException UNAVAILABLE_ELEMENT	

6.2.4.4.4 Test Coverage

CRR number	Test case number
N1	1 to 22.
N2	6, 15.
N3	1 to 9.
N4	10 to 18.
N5	19, 20, 21, 22
N6	8
N7	17
N8	22
N9	7, 16, 21.
C1	23
C2	24

6.2.4.5 Method getTheHandler

Test Area Reference: API_2_ENH_GTHD

6.2.4.5.1 Conformance Requirements

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

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

6.2.4.5.1.1 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 Javacard 2.1 Runtime Environment (JCRE) Specification [12])

6.2.4.5.1.2 Parameters error

No requirements.

6.2.4.5.1.3 Context errors

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

6.2.4.5.2 Test suite files

Test Script:	API_2_ENH_GTHD_1.scr
Test Applet:	API_2_ENH_GTHD_1.java
Load Script:	API_2_ENH_GTHD_1.ldr
Cleanup Script:	API_2_ENH_GTHD_1.clr
Parameter File:	API_2_ENH_GTHD_1.par

6.2.4.5.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	getTheHandler twice	The returned objects shall be the same	
2	Verify that getTheHandler returns an EnvelopeHandler GetTheHandler	The reference returned shall be an EnvelopeHandler (check cast)	
3	Verify the returned value is not null GetTheHandler	The reference returned shall not be null.	

6.2.4.5.4 Test Coverage

CRR number	Test case number
N1	1, 2, 3
N2	Checked in Framework tests: FWK_API_HEPO (test case 1)
C1	Checked in Framework tests: FWK_MHA_ENHD

6.2.4.6 Method getTPUDLOffset

Test Area Reference: API_2_ENH_GTPO

6.2.4.6.1 Conformance Requirement:

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

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

6.2.4.6.1.1 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.

6.2.4.6.1.2 Parameters error

No requirements.

6.2.4.6.1.3 Context errors

- 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.

6.2.4.6.2 Test suite files

Specific triggering:

- FORMATTED SMS PP UPD.
- UNFORMATTED SMS PP UPD.
- UNFORMATTED SMS PP ENV.
- UNFORMATTED SMS CB.

Test Script:	API_2_ENH_GTPO_1.scr
Test Applet:	API_2_ENH_GTPO_1.java

Load Script: API_2_ENH_GTPO_1.ldr

Cleanup Script: API_2_ENH_GTPO_1.clr

Parameter File: API_2_ENH_GTPO_1.par

6.2.4.6.3 Test procedure

Id	Description	API Expectation	APDU Expectation
	FORMATTED SMS PP ENV triggering		
1	Test with TP-OA length of 2	Returns 0x0D	
2	Test with TP-OA length of 6	Returns 0x0F	
3	Test with TP-OA length of 12	Returns 0x12	
4	Send a SMS PP with 2 TPDU TLV and inside two different UDL offsets	Returns 0x10 (the first offset)	
5	Same test as 1, but with a concatenated SMS (2 Short Messages and maximum Secured Data Length = 0x00FA)	Returns 0x0D	
6	Verify after call of the method the current TLV is the TPDU TLV: findTLV device identities, getTPUDLOffset and then getValueByte to verify that the current TLV is the TPDU TLV	Returns 0x40	
	FORMATTED SMS PP UPD triggering		
7	Same test as 1 but with FORMATTED_SMS_PP_UPD	Returns 0x0D	
8	Same test as 2 but with FORMATTED_SMS_PP_UPD	Returns 0x0F	
9	Same test as 3 but with FORMATTED_SMS_PP_UPD	Returns 0x12	
10	Same test as 4 but with FORMATTED_SMS_PP_UPD	Returns 0x10 (the first offset)	
11	Same test as 7, but with a concatenated SMS (2 Short Messages and maximum Secured Data Length = 0x00FA)	Returns 0x0D	
	UNFORMATTED SMS PP UPD triggering		
12	Same test as 1 but with UNFORMATTED_SMS_PP_UPD	Returns 0x0D	
13	Same test as 2 but with UNFORMATTED_SMS_PP_UPD	Returns 0x0F	
14	Same test as 3 but with UNFORMATTED_SMS_PP_UPD	Returns 0x12	
15	Same test as 4 but with UNFORMATTED_SMS_PP_UPD	Returns 0x12 (the first offset)	
16	Same test as 12, but with a concatenated SMS (2 Short Messages and maximum User Data Length = 0x010C)	Returns 0x0D	
	UNFORMATTED SMS PP ENV triggering		
17	Same test as 1 but with UNFORMATTED_SMS_PP_ENV	Returns 0x0D	
18	Same test as 2 but with UNFORMATTED_SMS_PP_ENV	Returns 0x0F	
19	Same test as 3 but with UNFORMATTED_SMS_PP_ENV	Returns 0x12	
20	Same test as 4 but with UNFORMATTED_SMS_PP_ENV	Returns 0x10 (the first offset)	
21	Same test as 17, but with a concatenated SMS (2 Short Messages and maximum User Data Length = 0x010C)	Returns 0x0D	
	SMS CB triggering		
22	Send an envelope SMS CB, getTPUDLOffset	ToolkitException UNAVAILABLE_ELEMENT	

6.2.4.6.4 Test Coverage

CRR number	Test case number
N1	1 to 21.
N2	4, 10, 15, 20.
N3	1, 2, 3, 4, 5, 6
N4	7, 8, 9, 10, 11
N5	12, 13, 14, 15, 16
N6	17, 18, 19, 20, 21
N7	6
C1	22
C2	Not applicable

6.2.4.7 Method getLength

Test Area Reference: API_2_ENH_GLEN

6.2.4.7.1 Conformance Requirement

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

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

6.2.4.7.1.1 Normal execution

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

6.2.4.7.1.2 Parameter Error

No requirements.

6.2.4.7.1.3 Context errors

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

6.2.4.7.2 Test Suite files

Specific triggering: None

Test Script:	API_2_ENH_GLEN_1.scr
Test Applet:	API_2_ENH_GLEN_1.java
Load Script:	API_2_ENH_GLEN_1.ldr
Cleanup Script:	API_2_ENH_GLEN_1.clr
Parameter File:	API_2_ENH_GLEN_1.par

6.2.4.7.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Send an envelope SMS PP with BER length of 0x31	Result of getLength() is 0x0031	
2	Send an envelope SMS PP with BER length of 0x7F	Result of getLength() is 0x007Fh	
3	Send an envelope SMS PP with BER length of 81 80	Result of getLength() is 0x0080h	
4	Send an envelope SMS PP with BER length of 81 FC (maximum length for a single SMS)	Result of getLength() is 0x00FCh	
5	Send formatted SMS with BER length of 0x00FF, using 2 concatenated SMS	Result of getLength() is 0x00FFh	

6	Send formatted SMS with BER length of 0x0100, using 2 concatenated SMS	Result of getLength() is 0x0100h	
7	Send formatted SMS with maximum user data length (0x10D) (BER length:0x012F), using 2 concatenated SMS	Result of getLength() is 0x012Fh	

6.2.4.7.4 Test Coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5, 6, 7
C1	Does not apply for EnvelopeHandler

6.2.4.8 Method copy

Test Area Reference: API_2_ENH_COPY_BSS

6.2.4.8.1 Conformance Requirement

The method with following header 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
```

6.2.4.8.1.1 Normal execution

- CRRN1: copies the simple TLV list contained in the handler to the destination byte array.
- CRRN2: returns dstOffset + dstLength.

6.2.4.8.1.2 Parameter errors

- 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.

6.2.4.8.1.3 Context errors

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

6.2.4.8.2 Test Suite files

Specific triggering: None

Test Script:	API_2_ENH_COPY_BSS_1.scr
Test Applet:	API_2_ENH_COPY_BSS_1.java
Load Script:	API_2_ENH_COPY_BSS_1.ldr
Cleanup Script:	API_2_ENH_COPY_BSS_1.clr
Parameter File:	API_2_ENH_COPY_BSS_1.par

6.2.4.8.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**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`
	`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 = 48`
	`DstOffset = 0`
	`DstLength = 48`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

8	**Successful call, dstBuffer is the whole buffer**
	`DstBuffer.length = 47`
	`DstOffset = 0`
	`DstLength = 47`
		Result of copy() is 0X0047	

9	**Compare the buffer**	Result of arrayCompare() is 0	

10	**Successful call, dstBuffer is part of a buffer**
	`DstBuffer.length = 50`
	`dstOffset = 3`
	`dstLength = 47`
		Result of copy() is 0X0032	

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 0X0009	

13	**Compare the whole buffer**	Result of arrayCompare() is 0	

14	**Successful call, dstBuffer is part of a buffer**
	`dstBuffer.length = 260`
	`dstOffset = 257`
	`dstLength = 3`
		Result of copy() is 0X0104	

15	**Compare the whole buffer**	Result of arrayCompare() is 0	

16	**Successful call, copy with length =0**
	`dstBuffer.length = 260`
	`dstOffset = 260`
	`dstLength = 0`
		Result of copy() is 0x104	

	Send a Formatted SMS PP with the maximum user data length = 0x010D, using 2 concatenated envelopes		

17	**Successful call, copy with length =299**
	`dstBuffer.length = 299`
	`dstOffset = 0`
	dstLength = 299
		Result of copy() is 0x12B	

6.2.4.8.4 Test Coverage

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

6.2.4.9 Method findTLV

Test Area Reference: API_2_ENH_FINDBB

6.2.4.9.1 Conformance Requirement

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

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

6.2.4.9.1.1 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, the current TLV is no longer defined and TLV_NOT_FOUND is returned.
- CRRN5: The search method is comprehension required flag independent.

6.2.4.9.1.2 Parameter errors

- 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.

6.2.4.9.1.3 Context errors

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

6.2.4.9.2 Test Suite files

Specific triggering: None

Test Script:	API_2_ENH_FINDBB_1.scr
Test Applet:	API_2_ENH_FINDBB_1.java
Load Script:	API_2_ENH_FINDBB_1.ldr
Cleanup Script:	API_2_ENH_FINDBB_1.clr
Parameter File:	API_2_ENH_FINDBB_1.par

6.2.4.9.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
	Trig the applet with SMS PP including one more tag 02 and one TAG 04		
1	**Invalid input parameter** `Occurrence = 0`	ToolkitException.BAD_INPUT_PARAMETER is thrown	
2	**Search 1st TLV** `Tag = 02h`	Result is TLV_FOUND_CR_SET	

Id	Description	API Expectation	APDU Expectation
	`Occurrence = 1`		
3	**Call the getValueLength() method**	Result is 0x02	

4	**Search 2^{nd} TLV**
	`Tag = 06h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_SET	

5	**Call the getValueLength() method**	Result is 0x05h	
6	**Select a TLV (tag 02h)**		

	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 = 02h`
	`Occurrence = 3`
		Result is TLV_NOT_FOUND	

9	**Call the getValueLength() method**	ToolkitException.UNAVAILABLE_ ELEMENT is thrown.	

10	**Search the TLV**
	`Tag = 02h`
	`Occurrence = 2`
		Result is TLV_FOUND_CR_NOT_SET	

11	**Search the TLV**
	`Tag = 04h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_NOT_SET	

12	**Search tag 86h**
	`Tag = 86h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_SET	

13	**Search tag 84h**
	`Tag = 84h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_NOT_SET	

6.2.4.9.4 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 EnvelopeHandler

6.2.4.10 Method getValueLength

Test Area Reference: API_2_ENH_GVLE

6.2.4.10.1 Conformance Requirement

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

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

6.2.4.10.1.1 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.

6.2.4.10.1.2 Parameter errors

No requirements.

6.2.4.10.1.3 Context errors

- 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.

6.2.4.10.2 Test Suite files

Specific triggering: None

Test Script:	API_2_ENH_GVLE_1.scr
Test Applet:	API_2_ENH_GVLE_1.java
Load Script:	API_2_ENH_GVLE_1.ldr
Cleanup Script:	API_2_ENH_GVLE_1.clr
Parameter File:	API_2_ENH_GVLE_1.par

6.2.4.10.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
	Fill the SMS PP with TLV: Tag 33, Length C8		
1	**getValueLength()**	ToolkitException.UNAVAILABLE_ ELEMENT is thrown	
2	**Search TLV 02h**		
	`getValueLength()`	Result is 0X0002	
3	**Search TLV 0Bh**		
	`getValueLength()`	Result is 0X0024	
4	**Search TLV 33h**		
	`getValueLength()`	Result is 0X00C8	
	Send Formatted SMS PP with the maximum user data length = 0x010D, using 2 concatenated envelopes		
5	`Search SMS TPDU TAG`		
	`getValueLength()`	Result is 0X0120	

6.2.4.10.4 Test Coverage

CRR number	Test case number
N1	2, 3, 4, 5
C1	Does not apply for EnvelopeHandler
C2	1

6.2.4.11 Method getValueByte

Test Area Reference: API_2_ENH_GVBYS

6.2.4.11.1 Conformance Requirement

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

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

6.2.4.11.1.1 Normal execution

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

6.2.4.11.1.2 Parameter errors

- 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.

6.2.4.11.1.3 Context errors

- 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.

6.2.4.11.2 Test Suite files

Specific triggering: None

Test Script:	API_2_ENH_GVBYS_1.scr
Test Applet:	API_2_ENH_GVBYS_1.java
Load Script:	API_2_ENH_GVBYS_l.dr
Cleanup Script:	API_2_ENH_GVBYS_1.clr
Parameter File:	API_2_ENH_GVBYS_1.par

6.2.4.11.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
	Fill the SMS PP with TLV: Tag 33, Length C8 Value 01 02 …		
1	getValueByte(0)	ToolkitException.UNAVAILABLE_ ELEMENT is thrown	
2	**Search TLV 02h**		
	`getValueByte(2)`	ToolkitException.OUT_OF_TLV_ BOUNDARIES is thrown	
3	**Search TLV 02h**		
	`getValueByte(1)`	Result is 0x81	
4	**Search TLV 02h (Device Identities TLV)**		
	`getValueByte(0)`	Result is 83h (Source)	
5	**Search TLV 33h**		
	`getValueByte(7E)`	Result is 0x7F	
6	**Search TLV 33h**		
	`getValueByte(80)`	Result is 0x81	
7	`getValueByte(7F)`	Result is 0x80	
8	**Search TLV B3h**		
	`getValueByte(C7)`	Result is 0xC8	
	Send Formatted SMS PP with the maximum user data length = 0x010D, using 2 concatenated envelopes		
9	`Search SMS TPDU TAG`		
	`getValueByte(0x011F)`	Result is 0xFA	

6.2.4.11.4 Test Coverage

CRR number	Test case number
N1	3, 4, 5, 6, 7, 8, 9
P1	2
C1	Does not apply for EnvelopeHandler
C2	1

6.2.4.12 Method copyValue

Test Area Reference: API_2_ENH_CPYVS_BSS

6.2.4.12.1 Conformance Requirement

The method with following header 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
```

6.2.4.12.1.1 Normal execution

- CRRN1: copies a part of the last TLV element which has been found, into a destination. buffer.
- CRRN2: returns dstOffset + dstLength.

6.2.4.12.1.2 Parameter errors

- 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.

6.2.4.12.1.3 Context errors

- 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.

6.2.4.12.2 Test Suite files

Specific triggering: None

Test Script:	API_2_ENH_CPYVS_BSS_1.scr
Test Applet:	API_2_ENH_CPYVS_BSS_1.java
Load Script:	API_2_ENH_CPYVS_BSS_1.ldr
Cleanup Script:	API_2_ENH_CPYVS_BSS_1.clr
Parameter File:	API_2_ENH_CPYVS_BSS_1.par

6.2.4.12.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Search TLV 02h**		
	`copyValue() with a null dstBuffer`	NullPointerException is thrown	
2	**Search TLV 0Bh**		

	dstOffset ≥ dstBuffer.length
	`dstBuffer.length = 5`
	`dstOffset = 5`
	`dstLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

3	**dstOffset < 0**	ArrayIndexOutOfBoundsExceptio	

Id	Description	API Expectation	APDU Expectation

	dstBuffer.length = 5
	dstOffset = -1
	dstLength = 1
		n 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	**Search TLV 06h**		

	valueOffset ≥ TLV 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 > TLV length**
	valueOffset = 0
	dstBuffer.length = 15
	dstOffset = 0
	dstLength = 7
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

10	**valueOffset + dstLength > TLV length**
	valueOffset = 2
	dstBuffer.length = 15
	dstOffset = 0
	dstLength = 5
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

11	**Search TLV 01h**		
	copyValue()	ToolkitException.UNAVAILABLE_ELEMENT is thrown on the copyValue() method	
12	**Search TLV 06h**		

	Successful call
	valueOffset = 0
	dstBuffer.length = 6
	dstOffset = 0
	dstLength = 6
		Result of copyValue() is 0x0006	

13	**Compare buffer** buffer = 81 11 22 33 44 F5	Result is 00h	
14	**initialise dstBuffer** dstBuffer = 55 55 … 55		

	Successful call
	valueOffset = 1
	dstBuffer.length = 20
	dstOffset = 3
	dstLength = 4
		Result of copyValue() is 0x0007	

15	**Compare buffer** buffer = 55 55 55 11 22 33 44 55 55 55 55 55 55 55 55 55 55 55 55 55	Result is 00h	

16	**Successful call, copy with length =0**
	dstBuffer.length = 20
	dstOffset = 20
	dstLength = 0
		Result of copyValue() is 20	

	Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes		

Id	Description	API Expectation	APDU Expectation
17	**Search SMS TPDU TAG**		

	Successful call
	`valueOffset = 0x11`
	`dstBuffer.length = 0x010D`
	`dstOffset = 0`
	`dstLength = 0x010D`
		Result of copyValue() is 0x010D	

18	**Compare buffer** `buffer = 0348 header and secured data (01 … FA)`	Result is 00h	
19	**Initialise dstBuffer** `dstBuffer = 55 55 … 55`		

	Successful call
	`valueOffset = 0x0111`
	`dstBuffer.length = 0x010D`
	`dstOffset = 0x0100`
	`dstLength = 0x000D`
		Result of copyValue() is 0x010D	

20	**Compare buffer**
	`buffer =`
	`55 55 55 55 55 55 55 55`
	`…`
	`55 55 EE EF F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA`
		Result is 00h	

6.2.4.12.4 Test Coverage

CRR number	Test case number
N1	13, 15, 18, 20
N2	12, 14, 16, 17, 19
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for EnvelopeHandler
C2	11

6.2.4.13 Method compareValue

Test Area Reference: API_2_ENH_CPRVS_BSS

6.2.4.13.1 Conformance requirement

The method with following header 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
```

6.2.4.13.1.1 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.

6.2.4.13.1.2 Parameter errors

- 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.

6.2.4.13.1.3 Context errors

- 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.

6.2.4.13.2 Test Suite files

Specific triggering: None

Test Script:	API_2_ENH_CPRVS_BSS_1.scr
Test Applet:	API_2_ENH_CPRVS_BSS_1.java
Load Script:	API_2_ENH_CPRVS_BSS_1.ldr
Cleanup Script:	API_2_ENH_CPRVS_BSS_1.clr
Parameter File:	API_2_ENH_CPRVS_BSS_1.par

6.2.4.13.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Search TLV 02h**		
	compareValue() with a null compareBuffer	NullPointerException is thrown	
2	**Search TLV 0Bh**		

	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	**Search TLV 06h**		

	valueOffset ≥ TLV 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 > TLV length** valueOffset = 0	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

Id	Description	API Expectation	APDU Expectation

	compareBuffer.length = 15
	compareOffset = 0
	compareLength = 7		

10	**valueOffset + compareLength > TLV length**
	valueOffset = 2
	compareBuffer.length = 15
	compareOffset = 0
	compareLength = 5
		ToolkitException.OUT_OF_TLV_ BOUNDARIES is thrown	

11	**Search TLV 01h**	Result is TLV_NOT_FOUND	
	compareValue()	ToolkitException.UNAVAILABLE_ ELEMENT is thrown	
12	**Search TLV 06h**		
	Initialise compareBuffer compareBuffer = 81 11 22 33 44 F5		

	Compare buffers
	valueOffset = 0
	compareOffset = 0
	compareLength = 6
		Result is 00h	

13	**Initialise compareBuffer** compareBuffer = 7F 11 22 33 44 F5		
	Compare buffers with same parameters	Result is -1	
14	**Initialise compareBuffer** compareBuffer = 83 11 22 33 44 F5		
	Compare buffers with same parameters	Result is -1	
15	**Initialise compareBuffer** compareBuffer = 55 55 55 81 11 22 33 44 F5 55 55 55 55 55		

	Compare buffers
	valueOffset = 1
	compareOffset = 4
	compareLength = 5
		Result is 00h	

16	**Initialise compareBuffer** compareBuffer = 55 55 55 81 10 22 33 44 F5 55 55 55 55 55		
	Compare buffers with same parameters	Result is +1	
17	**Initialise compareBuffer** compareBuffer = 55 55 55 81 12 22 33 44 F5 55 55 55 55 55		
	Compare buffers with same parameters	Result is -1	

18	**Successful call, compareValue with length** =0
	CompareBuffer.length = 15
	CompareOffset = 15
	CompareLength = 0
		Result of compareValue() is 0	

	Send Formatted SMS PP with the maximum user data length = 0x010D, using 2 concatenated envelopes		
	Search SMS TPDU TAG		
	Initialise compareBuffer compareBuffer = 0348 header and formatted data(01 02 … FA)		

19	**Compare buffers**
	valueOffset = 0x11
	compareOffset = 0
	compareLength = 0x010D
	compareBufferLength = 0x010D
		Result is 00h	

20	**Compare buffers**
	valueOffset = 0x0111
	compareOffset = 0x0100
	compareLength = 0x000D
	compareBufferLength = 0x010D
		Result is 00h	

6.2.4.13.4 Test Coverage

CRR number	Test case number
N1	12, 15, 19, 20
N2	13, 16, 18
N3	14, 17
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for EnvelopeHandler
C2	11

6.2.4.14 Method findAndCopyValue(byte tag, byte[] dstBuffer, short dstOffset)

Test Area Reference: API_2_ENH_FACYB_BS

6.2.4.14.1 Conformance requirement

The method with following header 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
```

6.2.4.14.1.1 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.

6.2.4.14.1.2 Parameter errors

- CRRP1: if dstBuffer is null NullPointerException shall be thrown.
- CRRP2: if dstOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

6.2.4.14.1.3 Context errors

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

6.2.4.14.2 Test Suite files

Specific triggering: None

Test Script:	API_2_ENH_FACYB_BS_1.scr
Test Applet:	API_2_ENH_FACYB_BS_1.java
Load Script:	API_2_ENH_FACYB_BS_1.ldr
Cleanup Script:	API_2_ENH_FACYB_BS_1.clr
Parameter File:	API_2_ENH_FACYB_BS_1.par

6.2.4.14.3 Test procedure

Id	Description	API Expectation	APDU Expectation
	Fill the SMS PP with TLV: Tag 02 Value 22 44 Tag 33, Length C4 Value 01 02 …		
1	**FindAndCopyValue() with a null dstBuffer**	NullPointerException is thrown	

2	**dstOffset ≥ dstBuffer.length**
	tag = 06h
	dstBuffer.length = 06
	dstOffset = 06
		ArrayIndexOutOfBoundsException is thrown	

3	**dstOffset < 0**
	dstBuffer.length = 06
	dstOffset = -1
		ArrayIndexOutOfBoundsException is thrown	

4	**length > dstBuffer.length**
	dstBuffer.length = 05
	dstOffset = 0
		ArrayIndexOutOfBoundsException is thrown	

5	**DstOffset + length >dstBuffer.length**
	DstBuffer.length = 06
	DstOffset = 1
		ArrayIndexOutOfBoundsException is thrown	

6	**Select a TLV (tag 02h)**		

	findAndCopyValue()
	tag = 03h
		ToolkitException.UNAVAILABLE_ELEMENT is thrown	

7	**Call the getValueLength() method**	ToolkitException.UNAVAILABLE_ELEMENT is thrown.	

8	**Successful call**
	Tag = 06h
	DstBuffer.length = 06
	DstOffset = 0
		Result of findAndCopyValue () is 0006	

9	**Compare buffer** buffer = 81 11 22 33 44 F5	Result is 00h	
10	**initialise dstBuffer** dstBuffer = 55 55 … 55		

	Successful call
	dstBuffer.length = 12
	dstOffset = 2
		Result of findAndCopyValue () is 0008	

11	**Compare buffer** buffer = 55 55 81 11 22 33 44 F5 55 55 55 55	Result is 00h	

12	**Successful call**
	tag = 02h
	dstBuffer.length = 2
	dstOffset = 0
		Result of findAndCopyValue () is 0002	

13	**Compare buffer** buffer = 83 81	Result is 00h	

14	**Successful call (with tag 82h)**
	tag = 82h
	dstBuffer.length = 02
	dstOffset = 0
		Result of findAndCopyValue () is 0002	

15	**Compare buffer** buffer = 83 81	Result is 00h	

16	**Successful call (with tag B3h)**
	tag = B3h
	dstBuffer.length = C4
	dstOffset = 0
		Result of findAndCopyValue () is 00C4	

Id	Description	API Expectation	APDU Expectation
17	**Compare buffer** `buffer = 01 02 … C4`	Result is 00h	
	Send Formatted SMS PP with the maximum user data length = 0x010D, using 2 concatenated envelopes		

18	**Successful call (with SMS TPDU TAG)**
	`tag = 0Bh`
	`dstBuffer.length = 0x011E`
	dstOffset = 0
		Result of findAndCopyValue () is 0x011E	

19	**Compare buffer** `buffer = 0348 Header + secured data (01 02 … FA)`	Result is 00h	

20	**Successful call (with SMS TPDU TAG)**
	`tag = 0Bh`
	`dstBuffer.length = 0x0220`
	dstOffset = 0x0100
		Result of findAndCopyValue () is 0x021E	

21	**Compare buffer** `buffer = 0348 Header + secured data (01 02 … FA)`	Result is 00h	

6.2.4.14.4 Test Coverage

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

6.2.4.15 Method findAndCopyValue(byte tag, byte occurrence, short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)

Test Area Reference: API_2_ENH_FACYBS_BSS

6.2.4.15.1 Conformance requirement

The method with following header 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
```

6.2.4.15.1.1 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.

6.2.4.15.1.2 Parameter errors

- 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.

6.2.4.15.1.3 Context errors

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

6.2.4.15.2 Test Suite files

Test Script:	API_2_ENH_FACYBS_BSS_1.scr
Test Applet:	API_2_ENH_FACYBS_BSS_1.java
Load Script:	API_2_ENH_FACYBS_BSS_1.ldr
Cleanup Script:	API_2_ENH_FACYBS_BSS_1.clr
Parameter File:	API_2_ENH_FACYBS_BSS_1.par

6.2.4.15.3 Test procedure

Id	Description	API Expectation	APDU Expectation
	Fill the SMS PP with TLV: Tag 02 Value 22 44 Tag 33, Length C4 Value 01 02 …		
1	**findAndCopyValue() with a null dstBuffer**	NullPointerException is thrown	

2	**dstOffset ≥ dstBuffer.length**
	`tag = 06h, 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	**valueOffset ≥ Value Length**
	`tag = 06h, 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 > Value length** `valueOffset = 0`	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

Id	Description	API Expectation	APDU Expectation

	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	**Select a TLV (tag 02h)**		

	findAndCopyValue()
	tag = 06h
	occurrence = 2
		ToolkitException.UNAVAILABLE_ ELEMENT is thrown	

12	**Call the getValueLength() method**	ToolkitException.UNAVAILABLE_ ELEMENT is thrown.	

13	**Successful call**
	tag = 06h, occurrence = 1
	valueOffset = 0
	dstBuffer.length = 06
	dstOffset = 0
	dstLength = 06
		Result of findAndCopyValue() is 6	

14	**Compare buffer** buffer = 81 11 22 33 44 F5	Result is 00h	
15	**initialise dstBuffer** dstBuffer = 55 55 … 55		

	Successful call
	tag = 06h, occurrence = 1
	valueOffset = 2
	dstBuffer.length = 12
	dstOffset = 3
	dstLength = 04
		Result of findAndCopyValue () is 0007	

16	**Compare buffer** buffer = 55 55 55 22 33 44 F5 55 55 55 55 55	Result is 00h	

17	**Successful call**
	tag = 02h, occurrence = 1
	valueOffset = 0
	dstBuffer.length = 12
	dstOffset = 0
	dstLength = 2
		Result of findAndCopyValue() is 0002	

18	**Compare buffer** buffer = 83 81 55 … 55	Result is 00h	

19	**Successful call**
	tag = 02h, occurrence = 2
	valueOffset = 0
	dstBuffer.length = 12
	dstOffset = 0
	dstLength = 2
		Result of findAndCopyValue() is 0002	

20	**Compare buffer** buffer = 22 44 55 … 55	Result is 00h	

21	**Successful call (with tag 82h)**
	tag = 82h
	occurrence = 1
	valueOffset = 0
	dstBuffer.length = 12
	dstOffset = 0
	dstLength = 02
		Result of findAndCopyValue () is 0002	

22	**Compare buffer** buffer = 83 81 55 … 55	Result is 00h	

23	**Successful call (with tag 82h)**
	tag = 82h
	occurrence = 2
	valueOffset = 0
	dstBuffer.length = 12
	dstOffset = 0
	dstLength = 02
		Result of findAndCopyValue () is 0002	

24	**Compare buffer** Buffer = 22 44 55 … 55	Result is 00h	

25	**Successful call, findAndCopyValue with length =0**
	DstBuffer.length = 12
	dstOffset = 12
	dstLength = 0
		Result of findAndCopyValue () is 12	

	Send Formatted SMS PP with the maximum		

Id	Description	API Expectation	APDU Expectation
	user data length = 0x010D, using 2 concatenated envelopes		

26	**Successful call**
	`tag = 0Bh, occurrence = 1`
	`valueOffset = 0x11`
	`dstBuffer.length = 0x010D`
	`dstOffset = 0`
	`dstLength = 0x010D`
		Result of findAndCopyValue() is 0x010D	

27	**Compare buffer** `buffer = 0348 Header + secured data (01 02 … FA)`	Result is 00h	
28	**initialise dstBuffer** `dstBuffer = 55 55 … 55`		

	Successful call
	`tag = 0Bh, occurrence = 1`
	`valueOffset = 0x0111`
	`dstBuffer.length = 0x010D`
	`dstOffset = 0x0100`
	`dstLength = 0x0D`
		Result of findAndCopyValue () is 0x010D	

29	**Compare buffer**
	`buffer =`
	`55 55 … 55 55 EE EF F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA`
		Result is 00h	

6.2.4.15.4 Test Coverage

CRR number	Test case number
N1	14, 15, 17, 19, 20
N2	11, 12
N3	13, 15, 17, 19, 25
N4	21, 22, 23, 24, 26, 27, 28,29
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for EnvelopeHandler

6.2.4.16 Method findAndCompareValue(byte tag, byte[] compareBuffer, short compareOffset)

Test Area Reference: API_2_ENH_FACRB_BS

6.2.4.16.1 Conformance requirement

The method with following header 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
```

6.2.4.16.1.1 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.

6.2.4.16.1.2 Parameter errors

- CRRP1: if compareBuffer is null NullPointerException shall be thrown.
- CRRP2: if compareOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

6.2.4.16.1.3 Context errors

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

6.2.4.16.2 Test Suite files

Test Script:	API_2_ENH_FACRB_BS_1.scr
Test Applet:	API_2_ENH_FACRB_BS_1.java
Load Script:	API_2_ENH_FACRB_BS_1.ldr
Cleanup Script:	API_2_ENH_FACRB_BS_1.clr
Parameter File:	API_2_ENH_FACRB_BS_1.par

6.2.4.16.3 Test procedure

Id	Description	API Expectation	APDU Expectation
	Fill the SMS PP with TLV: Tag 02 Value 22 44 Tag 33, Length C4 Value 01 02 …		
1	**findAndCompareValue() with a null dstBuffer**	NullPointerException is thrown	

2	**compareOffset ≥ compareBuffer.length**
	`tag = 06h`
	`compareBuffer.length = 12`
	`compareOffset = 12`
		ArrayIndexOutOfBoundsException is thrown	

3	**compareOffset < 0**
	`compareBuffer.length = 12`
	`compareOffset = -1`
		ArrayIndexOutOfBoundsException is thrown	

4	**length > compareBuffer.length**
	`compareBuffer.length = 05`
	`compareOffset = 0`
		ArrayIndexOutOfBoundsException is thrown	

5	**compareOffset + length > compareBuffer.length**
	`compareBuffer.length = 12`
	`compareOffset = 7`
		ArrayIndexOutOfBoundsException is thrown	

6	**Select a TLV (tag 02h)**		

	findAndCompareValue()
	`tag = 03h`
		ToolkitException.UNAVAILABLE_ ELEMENT is thrown	

7	**Call the getValueLength() method**	ToolkitException.UNAVAILABLE_ ELEMENT is thrown.	
8	**Initialise compareBuffer** `compareBuffer = 81 11 22 33 44 F5`		

	Compare buffers
	`tag = 06h`
	`compareOffset = 0`
		Result is 00h	

9	**Verify current TLV** `getValueLength()`	Result is 06	
10	**Initialise compareBuffer** `compareBuffer = 81 11 22 33 44 F4`		
	Compare buffers with same parameters	Result is +1	
11	**Initialise compareBuffer** `compareBuffer = 81 11 22 33 44 F6`		
	Compare buffers with same parameters	Result is -1	
12	**Initialise compareBuffer**		

	compareBuffer = 55 55 81 11 22 33 44 F5 55 55 55 55		
	Compare buffers compareOffset = 2	Result is 00h	
13	**Initialise compareBuffer** compareBuffer = 55 55 83 81 55 55 55 55 55 55 55 55		
	Compare buffers compareOffset = 2	Result is 00h	
14	**Initialise compareBuffer** compareBuffer = 55 55 83 80 55 55 55 55 55 55 55 55		
	Compare buffers compareOffset = 2	Result is +1	
15	**Initialise compareBuffer** compareBuffer = 55 55 83 82 55 55 55 55 55 55 55 55		
	Compare buffers compareOffset = 2	Result is -1	
16	**Initialise compareBuffer** compareBuffer = 83 81 55 55 55 55 55 55 55 55 55 55		

	Successful call (with tag 02h)
	tag = 02h
	compareBuffer.length = 12
	compareOffset = 0
		Result is 00h	

17	**Initialise compareBuffer** CompareBuffer = 01 02 … C4		

	Successful call (with tag B3h)
	Tag = B3h
	CompareBuffer.length = C4
	CompareOffset = 0
		Result is 00h	

	Send Unformatted SMS PP with the maximum user data length = 0x010C, using 2 concatenated envelopes		
	Initialise compareBuffer CompareBuffer = 0340 Header + user data (00 01 02 … FF 01 … 0C)		

18	**Successful call (with SMS TPDU TAG)**
	Tag = 0Bh
	CompareBuffer.length = 0x011E
	CompareOffset = 0
		Result is 00h	

	Initialise compareBuffer CompareBuffer = 55 55 … 55 CompareBuffer from offset 0x0100= 0340 Header + user data (00 01 02 … FF 01 … 0C)		

19	**Successful call (with SMS TPDU TAG)**
	Tag = 0Bh
	CompareBuffer.length = 0x220
	CompareOffset = 0x0100
		Result is 00h	

6.2.4.16.4 Test Coverage

CRR number	Test case number
N1	6,7
N2	9
N3	8, 12, 13, 18, 19
N4	10, 14
N5	11, 15
N6	16, 17
P1	1
P2	2, 3, 4, 5
C1	Does not apply for EnvelopeHandler

6.2.4.17 Method findAndCompareValue(byte tag, byte occurrence, short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)

Test Area Reference: API_2_ENH_FACRBBS_BSS

6.2.4.17.1 Conformance requirement

The method with following header 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
```

6.2.4.17.1.1 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.

6.2.4.17.1.2 Parameter errors

- 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.

6.2.4.17.1.3 Context errors

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

6.2.4.17.2 Test Suite files

Test Script:	API_2_ENH_FACRBBS_BSS_1.scr
Test Applet:	API_2_ENH_FACRBBS_BSS_1.java
Load Script:	API_2_ENH_FACRBBS_BSS_1.ldr
Cleanup Script:	API_2_ENH_FACRBBS_BSS_1.clr
Parameter File:	API_2_ENH_FACRBBS_BSS_1.par

6.2.4.17.3 Test procedure

Id	Description	API Expectation	APDU Expectation
	Fill the SMS PP with TLV: Tag 02 Value 22 44 Tag 33, Length C4 Value 01 02 …		
1	**findAndCompareValue() with a null compareBuffer**	NullPointerException is thrown	

2	**compareOffset ≥ compareBuffer.length**
	`tag = 06h, occurrence = 1`
	`valueOffset = 0`
	`compareBuffer.length = 6`
	`compareOffset = 6`
	`compareLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

3	**compareOffset < 0**
	`compareBuffer.length = 6`
	`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 ≥ Value Length**
	`tag = 06h, 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 > Value length**
	`valueOffset = 0`
	`compareBuffer.length = 15`
	`compareOffset = 0`
	`compareLength = 7`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

10	**valueOffset + compareLength > Value 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	**Select a TLV (tag 02h)**		

	findAndCompareValue()
	`tag = 06h`
	`occurrence = 2`
		ToolkitException.UNAVAILABLE_ELEMENT is thrown	

13	**Call the getValueLength() method**	ToolkitException.UNAVAILABLE_ELEMENT is thrown.	
14	**Initialise compareBuffer** `compareBuffer = 81 11 22 33 44 F5`		

	findAndCompareValue()
	`tag = 06h, occurrence = 1`
	`valueOffset = 0`
	`compareOffset = 0`
	`compareLength = 6`
		Result is 00h	

15	**Verify current TLV** `getValueLength()`	Result is 0006	
16	**Initialise compareBuffer** `compareBuffer = 81 11 22 33 44 F4`		
	Compare buffers with same parameters	Result is +1	
17	**Initialise compareBuffer** `compareBuffer = 81 11 22 33 44 F6`		

Id	Description	API Expectation	APDU Expectation
	Compare buffers with same parameters	Result is -1	

18	**Initialise compareBuffer**
	`compareBuffer =`
	`55 55 55 22 33 44 F5 55 55 55 55`		

	Compare buffers
	`valueOffset = 2`
	`compareOffset = 3`
	`compareLength = 4`
		Result is 00h	

19	**Initialise compareBuffer**
	`compareBuffer =`
	`55 55 55 22 33 45 F5 55 55 55 55`		

	Compare buffers with same parameters	Result is -1	

20	**Initialise compareBuffer**
	`compareBuffer =`
	`55 55 55 22 33 43 F5 55 55 55 55`		

	Compare buffers with same parameters	Result is +1	

21	**Initialise compareBuffer**
	`compareBuffer =`
	`83 81 55 55 55 55 55 55 55 55 55 55`		

	findAndCompareValue()
	`tag = 02h, occurrence = 1`
	`valueOffset = 0`
	`compareOffset = 0`
	`compareLength = 2`
		Result is 00h	

22	**Initialise compareBuffer**
	`compareBuffer =`
	`22 44 55 55 55 55 55 55 55 55 55 55`		

	findAndCompareValue()
	`tag = 02h, occurrence = 2`
	`valueOffset = 0`
	`compareOffset = 0`
	`compareLength = 2`
		Result is 00h	

23	**Initialise compareBuffer**
	`compareBuffer =`
	`22 45 55 55 55 55 55 55 55 55 55 55`		

	findAndCompareValue()
	`tag = 02h, occurrence = 2`
	`valueOffset = 0`
	`compareOffset = 0`
	`compareLength = 2`
		Result is -1	

24	**Initialise compareBuffer**
	`compareBuffer =`
	`83 81 55 55 55 55 55 55 55 55 55 55`		

	Successful call (with tag 02h)
	`tag = 02h, occurrence = 1`
	`valueOffset = 0`
	`compareBuffer.length = 12`
	`compareOffset = 0`
	`compareLength = 2`
		Result is 00h	

25	**Initialise compareBuffer** `compareBuffer = 01 02 … C4`		

	Successful call (with tag B3h)
	`tag = B3h, occurrence = 1`
	`valueOffset = 0`
	`compareBuffer.length = 00C4`
	`compareOffset = 0`
	`compareLength = 00C4`
		Result is 00h	

26	**Successful call, findAndCompareValue with length =0**
	`DstBuffer.length = C4`
	`DstOffset = C4`
	`DstLength = 0`
		Result of findAndCompareValue() is 00h	

	Send Formatted SMS PP with the maximum user data length = 0x010D, using 2 concatenated envelopes		

Id	Description	API Expectation	APDU Expectation
	Initialise compareBuffer `CompareBuffer = 23.048 Header + secured data (01 02 … FA)`		

27	**Successful call (with SMS TPDU TAG)**
	`tag = 0Bh, occurrence = 1`
	`valueOffset = 0x11`
	`compareBuffer.length = 0x010D`
	`compareOffset = 0`
	`compareLength = 0x010D`
		Result is 00h	

	Initialise compareBuffer `CompareBuffer = 55 55 … 55 EE EF F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA`		

28	**Successful call (with SMS TPDU TAG)**
	`tag = 0Bh, occurrence = 1`
	`valueOffset = 0x11`
	`compareBuffer.length = 0x010D`
	`compareOffset = 0x0100`
	`compareLength = 0x0D`
		Result is 00h	

6.2.4.17.4 Test Coverage

CRR number	Test case number
N1	12, 13
N2	15
N3	14, 18, 21, 22, 26, 27, 28
N4	16, 20
N5	17, 19, 23
N6	24, 25
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
P4	11
C1	Does not apply for EnvelopeHandler

6.2.4.18 Method getCapacity

Test Area Reference: API_2_ENH_GCAP

6.2.4.18.1 Conformance Requirement:

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

```
public byte getCapacity()
```

6.2.4.18.1.1 Normal execution

- CRRN1: The method shall return the maximum size of the Simple TLV list managed by the handler.

6.2.4.18.2 Test suite files

Test Script:	API_2_ENH_GCAP_1.scr
Test Applet:	API_2_ENH_GCAP_1.java
Load Script:	API_2_ENH_GCAP_1.ldr
Cleanup Script:	API_2_ENH_GCAP_1.clr
Parameter File:	API_2_ENH_GCAP_1.par

6.2.4.18.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**EnvelopeHandler available** 1 - Send envelope SMS-PP Formatted 2 - The applet calls the getLength() method 3 - The applet calls getCapacity()method	1 - Applet is triggered 2 - No exception is thrown 3 - No exception is thrown; the capacity is greater than the BER TLV Length	

6.2.4.18.4 Test Coverage

CRR number	Test case number
N1	1

6.2.4.19 Method getUserDataLength

Test Area Reference: API_2_ENH_GUDL

6.2.4.19.1 Conformance Requirement:

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

```
public short getUserDataLength()
```

6.2.4.19.1.1 Normal execution

- CRRN1: The method shall return the length of the User Data contained in the SMS TPDU TLV element.
- CRRN2: The length is from the first SMS TPDU TLV element.
- CRRN3: If the SMS TPDU TLV element is available, it becomes the selected TLV
- CRRN4: The method can be used if the event is EVENT_FORMATTED_SMS_PP_ENV.
- CRRN5: The method can be used if the event is EVENT_FORMATTED_SMS_PP_UPD.
- CRRN6: The method can be used if the event is EVENT_UNFORMATED_SMS_PP_ENV.
- CRRN7: The method can be used if the event is EVENT_UNFORMATTED_SMS_PP_UDP.

6.2.4.19.1.2 Context errors

- CRRC1: The method shall throw UNAVAILABLE_ELEMENT in case of unavailable TPDU TLV element.
- CRRC2: The method shall throw UNAVAILABLE_ELEMENT in case of wrong data format.

6.2.4.19.2 Test suite files

Specific triggering:

- UNFORMATTED_SMS_PP_ENV
- FORMATTED_SMS_PP_UPD
- UNFORMATED_SMS_PP_UPD
- UNRECOGNIZED_ENVELOPE
- For Formatted triggering if CC/RC/DS is used, the security parameters are those used for downloading applications.

Test Script:	API_2_ENH_GUDL_1.scr
Test Applet:	API_2_ENH_GUDL_1.java
Load Script:	API_2_ENH_GUDL_1.ldr
Cleanup Script:	API_2_ENH_GUDL_1.clr
Parameter File:	API_2_ENH_GUDL_1.par

6.2.4.19.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
	FORMATTED SMS PP ENV Triggering		
1	Test with FORMATTED_SMS_PP_ENV and TP-OA length of 2 and user data length of 0x3D	Returns 0x003D	
2	Test with TP-OA length of 12 and user data length of 0x3D	Returns 0x003D	
3	Test with RC/CC/DS length of 0 and secured data length of 0x10	Returns 0x0023	
4	Test with RC/CC/DS length of 8 and secured data length of 0x10	Returns 0x002B	
5	Test with PCNTR = 0, no RC/CC/DS and data length of 0x10	Returns 0x0023	
6	Test with PCNTR = 7, no RC/CC/DS and data length of 0x05	Returns 0x001F	
7	Test with SecuredDataLength = 00 and no RC/CC/DS	Returns 0x0013	
8	Test with UserDataLength = 0x7F	Returns 0x007F	
9	Test with UserDataLength = 0x80	Returns 0x0080	
10	Test with UserDataLength = maximum length (0x8C) for a single SMS	Returns 0x008C	
11	Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23	Returns 0x0018	
12	Send envelope SMS-PP Formatted. FindTLV() with TAG_DEVICE_IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0	GetValueByte() returns 0x40(23.040 first byte)	
13	Test with UserDataLength = 0xFF with 2 concatenated SMS	Returns 0x00FF	
14	Test with UserDataLength = 0x100 with 2 concatenated SMS	Returns 0x0100	
15	Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS	Returns 0x010D	
	FORMATTED SMS PP UPD Triggering		
16	Test with FORMATTED_SMS_PP_UPD and TP-OA length of 2 and user data length of 0x3D	Returns 0x003D	
17	Test with TP-OA length of 12 and user data length of 0x3D	Returns 0x003D	
18	Test with RC/CC/DS length of 0 and secured data length of 0x10	Returns 0x0023	
19	Test with RC/CC/DS length of 8 and secured data length of 0x10	Returns 0x002B	
20	Test with PCNTR = 0, no RC/CC/DS and data length of 0x10	Returns 0x0023	
21	Test with PCNTR = 7, no RC/CC/DS and data length of 0x05	Returns 0x001F	
22	Test with SecuredDataLength = 00 and no RC/CC/DS	Returns 0x0013	
23	Test with UserDataLength = 0x7F	Returns 0x007F	
24	Test with UserDataLength = 0x80	Returns 0x0080	
25	Test with UserDataLength = maximum length(0x8C) for a single SMS	Returns 0x008C	
26	Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23	Returns 0x0018	

27	Send envelope SMS-PP Formatted. FindTLV() with TAG_DEVICE_IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0	GetValueByte() returns 0x40(23.040 first byte)	
28	Test with UserDataLength = 0xFF with 2 concatenated SMS	Returns 0x00FF	
29	Test with UserDataLength = 0x100 with 2 concatenated SMS	Returns 0x0100	
30	Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS	Returns 0x010D	
	UNFORMATTED SMS PP ENV Triggering		
31	Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D	Returns 0x003D	
32	Test with TP-OA length of 12, and user data length of 0x3D	Returns 0x003D	
33	Test with UserDataLength = 0x00	Returns 0x0000	
34	Test with UserDataLength = 0x7F	Returns 0x007F	
35	Test with UserDataLength = 0x80	Returns 0x0080	
36	Test with UserDataLength = maximum length: 0x8C for a single SMS	Returns 0x008C	
37	Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23	Returns 0x0018	
38	Send envelope SMS-PP Unformatted. FindTLV() with TAG_DEVICE_IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0 (first user data = 0x55)	GetValueByte() returns 0x00 (23.040 first byte)	
	UNFORMATTED SMS PP UPD Triggering		
39	Test with UNFORMATTED_SMS_PP_UPD and TP-OA length of 2, and user data length of 0x3D	Returns 0x003D	
40	Test with TP-OA length of 12, and user data length of 0x3D	Returns 0x003D	
41	Test with UserDataLength = 0x00	Returns 0x0000	
42	Test with UserDataLength = 0x7F	Returns 0x007F	
43	Test with UserDataLength = 0x80	Returns 0x0080	
44	Test with UserDataLength = maximum length: 0x8C for a single SMS	Returns 0x008C	
45	Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23	Returns 0x0018	
46	Send envelope SMS-PP Unformatted. FindTLV() with TAG_DEVICE_IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0	GetValueByte() returns 0x00 (23.040 first byte)	
	UNRECOGNIZED_ENVELOPE Triggering		
47	Test with an UNRECOGNIZED_ENVELOPE	ToolkitException UNAVAILABLE_ELEMENT	

6.2.4.19.4 Test Coverage

CRR number	Test case number
N1	All test cases excepted: 53
N2	11, 26, 37, 45
N3	12, 27, 38, 46
N4	1 to 15
N5	16 to 30
N6	31 to 38
N7	39 to 46
C1	47
C2	Not applicable

6.2.4.20 Method getChannelIdentifier

Test Area Reference: API_2_ENH_GCID

6.2.4.20.1 Conformance Requirement:

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

```
public byte getChannelIdentifier()
                         throws ToolkitException
```

6.2.4.20.1.1 Normal execution

- CRRN1: The method shall return the channel identifier byte value.
- CRRN2: The channel identifier byte value returned shall be from the first Channel status TLV element.
- CRRN3: If the element is available it becomes the currently selected TLV.
- CRRN4: The channel identifier is available for all triggered toolkit applets from the invocation to the termination of their processToolkit method if the EnvelopeHandler is available.

6.2.4.20.1.2 Context errors

- CRRC1: The method shall throw ToolkitException (UNAVAILABLE_ELEMENT) if the Channel status TLV is not present.
- CRRC2: The method shall throw ToolkitException (OUT_OF_TLV_BOUNDARIES) if the Simple TLV Channel Status length is equal to 0.

6.2.4.20.2 Test suite files

Test Script:	API_2_ENH_GCID_1.scr
Test Applet:	API_2_ENH_GCID_1.java
Load Script:	API_2_ENH_GCID_1.ldr
Cleanup Script:	API_2_ENH_GCID_1.clr
Parameter File:	API_2_ENH_GCID_1.par

6.2.4.20.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	1- Applet1 is installed with maximum number of channel = 07. 2- Applet1 builds proactive commands OPEN CHANNEL with init() method in order to open all channels. ProactiveHandler.send() methods are called.		2- OPEN CHANNEL proactive command is fetched TERMINAL RESPONSE is issued with Channel Id from 01 to 07
1	1- Send envelope Event Download Channel Status with channel status TLV: channel status value = 0x8100. 2- Call EnvelopeHandler.getChannelIdentifier() method	1- Applet1 is triggered 2- Returns 0x01	
2	1- Send envelope Event Download Channel Status with two channel status TLV: first value = 0x8400 second value = 0x8500. 2- Call twice the EnvelopeHandler.getChannelIdentifier() method	2- Returns 0x04 Returns 0x04	
3	1- Send envelope Event Download Channel Status with channel status TLV: Channel Status value = 0x0605 ViewHandler.FindTLV() with Device IdentityTag. 2- Call EnvelopeHandler.getChannelIdentifier() method. 3- Compare EnvelopeHandler.getChannelIdentifier() and then ViewHandler.getValueByte(0).	2- Returns 0x06 3- GetChannelIdentifier() =getValueByte(0)	
4	1- Send envelope Menu Selection without Channel Status TLV. 2- Call EnvelopeHandler.getChannelIdentifier() method.	2- A Toolkit exception UNAVAILABLE_ELEMENT is thrown.	
5	1- Send Envelope Event Download Channel Status with Channel Status TLV: Channel status value = 0x0600 2- Call EnvelopeHandler.getChannelIdentifier() method.	1- Returns 0x06	
6	1- Send unrecognized envelope with a Channel Status TLV having a length equal to 0. 2- Call EnvelopeHandler.getChannelIdentifier() method.	2- A Toolkit exception OUT_OF_TLV_BOUNDARIES is thrown.	

6.2.4.20.4 Test Coverage

CRR number	Test case number
N1	1, 2
N2	3
N3	3
N4	5
C1	4
C2	6

6.2.5 Class EnvelopeResponseHandler

6.2.5.1 Method getTheHandler

Test Area Reference: API_2_ERH_GTHD

6.2.5.1.1 Conformance requirement:

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

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

6.2.5.1.1.1 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 Javacard 2.1 Runtime Environment (JCRE) Specification [12]).

6.2.5.1.1.1 Parameter errors

No requirements.

6.2.5.1.1.3 Context errors

- 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

6.2.5.1.2 Test suite files

Test Script:	API_2_ERH_GTHD_1.scr
Test Applet:	API_2_ERH_GTHD_1.java
Load Script:	API_2_ERH_GTHD_1.ldr
Cleanup Script:	API_2_ERH_GTHD_1.clr
Parameter File:	API_2_ERH_GTHD_1.par

6.2.5.1.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**getTheHandler twice**	The returned objects shall be the same	
2	**Verify that getTheHandler returns an EnvelopeHandler** `getTheHandler`	The reference returned shall be an EnvelopeResponseHandler (checkcast)	
3	**Verify the returned value is not null** `getTheHandler`	The reference returned shall not be null.	
4	**Send a proactive command, and then getTheHandler()**	ToolkitException HANDLER_NOT_AVAILABLE is thrown	

6.2.5.1.4 Test Coverage

CRR number	Test case number
N1	1, 2, 3
N2	Checked in Framework tests: FWK_API_HEPO (test case 2)
C1	Checked in Framework tests: FWK_MHA_ERHD
C2	4

6.2.5.2 Method post

Test Area Reference: API_2_ERH_POSTB

6.2.5.2.1 Conformance requirement

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

```
public void post(byte statusType)
          throws ToolkitException
```

6.2.5.2.1.1 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.

6.2.5.2.1.2 Parameter error

No requirements.

6.2.5.2.1.3 Context errors

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

6.2.5.2.2 Test suite files

Specific triggering: Unformatted SMS PP Env and Call control

Test Script:	API_2_ERH_POSTB_1.scr
Test Applet:	API_2_ERH_POSTB_1.java
Load Script:	API_2_ERH_POSTB_1.ldr
Cleanup Script:	API_2_ERH_POSTB_1.clr
Parameter File:	API_2_ERH_POSTB_1.par

6.2.5.2.3 Test procedure

Id	Description	API Expectation	APDU Expectation
	Call Control Triggering		
1	**getTheHandler and then post** (the handler is empty)		9000
2	**Fill the handler (appendTLV to have** bytes in it)and then post data with status 9F		9FFD data are retrieved with GET RESPONSE command
3	**Verify that after a post the handler is no more available** appendTLV, post and then appendTLV	ToolkitException HANDLER_NOT_AVAILABLE is thrown on the second appendTLV	
	Unformatted SMS PP Env triggering		
4	**construct the response (appendTLV with 0x10 data) and post it with status 9E and then send a display text**		9E12 and posted data retrieved by a GET RESPONSE with status 9113 and display text retrieved by a FETCH
5	**Verify that it is possible to send a proactive command after a post** getTheHandler and post , then send a display text		91 13 and display text is retrieved by a FETCH
6	**Verify it is not possible to post after a proactive command** getTheHandler, appendTLV, send a display text, post.	ToolkitException HANDLER_NOT_AVAILABLE is thrown	
7	**Verify that the handler is no more available**	ToolkitException	9E12 and posted data retrieved by a

	after a post `getTheHandler, appendTLV, post with status 9E, post with status 9F`	HANDLER_NOT_AVAILABLE is thrown	GET RESPONSE

6.2.5.2.4 Test Coverage

CRR number	Test case number
N1	3, 4, 7
N2	1, 2, 4, 7
N3	4, 5
N4	Checked in Framework tests: FWK_FWS_OUDA
C1	6

6.2.5.3 Method postAsBERTLV

Test Area Reference:API_2_ERH_POSTBB

6.2.5.3.1 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
```

6.2.5.3.1.1 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.

6.2.5.3.1.2 Parameter errors

No requirements.

6.2.5.3.1.3 Context errors

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

6.2.5.3.2 Test suite files

Specific triggering: Unformatted SMS PP Env and Call control

Test Script:	API_2_ERH_POSTBB_1.scr
Test Applet:	API_2_ERH_POSTBB_1.java
Load Script:	API_2_ERH_POSTBB_1.ldr
Cleanup Script:	API_2_ERH_POSTBB_1.clr
Parameter File:	API_2_ERH_POSTBB_1.par

6.2.5.3.3 Test procedure

Id	Description	API Expectation	APDU Expectation

	Call Control triggering		
1	**getTheHandler and then postAsBERTLV** `(the handler is empty)`		9F02 data are retrieved with GET RESPONSE command, the tag shall be 33 and the length is 00
2	**Fill the handler and then postAsBERTLV the data with status 9F, and tag 33**		9FFF data are retrieved with GET RESPONSE command, the tag shall be 33
3	**appendTLV, postAsBERTLV and then appendTLV**	ToolkitException HANDLER_NOT_AVAILABLE is thrown on the second appendTLV	
	Unformatted SMS PP Env triggering		
4	**construct the response (appendTLV with 0x10 data) and postAsBERTLV it with status 9E, tag 75 and then send a display text**		9E14 and posted data retrieved by a GET RESPONSE the tag shall be 75 with status 9113 and display text retrieved by a FETCH
5	**getTheHandler and postAsBERTLV, then send a display text**		9E02 and posted data retrieved by a GET RESPONSE the tag 33 (and the length 00) with status 9113 and display text is retrieved by a FETCH
6	**Verify it is not possible to postAsBERTLV after a proactive command** `getTheHandler, appendTLV, send a display text, postAsBERTLV.`	ToolkitException HANDLER_NOT_AVAILABLE is thrown on the postAsBERTLV	
7	**Verify that the handler is no more available after a postAsBERTLV** `getTheHandler, appendTLV(with data length = 0x10, postAsBERTLV with status 9E, tag 56, postAsBERTLV with status 9F, tag 28`	ToolkitException HANDLER_NOT_AVAILABLE is thrown on the second postAsBERTLV	9E14 and posted data retrieved by a GET RESPONSE the tag shall be 56 with status 9000

6.2.5.3.4 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

6.2.5.4 Method getLength

Test Area Reference: API_2_ERH_GLEN

6.2.5.4.1 Conformance requirement

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

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

6.2.5.4.1.1 Normal execution

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

6.2.5.4.1.2 Parameter errors

No requirements.

6.2.5.4.1.3 Context errors

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

6.2.5.4.2 Test Suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_GLEN_1.scr
Test Applet:	API_2_ERH_GLEN_1.java
Load Script:	API_2_ERH_GLEN_1.ldr
Cleanup Script:	API_2_ERH_GLEN_1.clr
Parameter File:	API_2_ERH_GLEN_1.par

6.2.5.4.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Clear the handler** `getLength()`	Result of getLength() is 0	
2	**appendTLV with length of 7** `getLength()`	Result of getLength() is 9	
3	**Clear the handler and appendTLV with Length of 250** `getLength()`	Result of getLength() is 253	
4	**Build a 7Fh Envelope response handler** `getLength()`	Result of getLength() is 81h	
5	**Build a 80h Envelope response handler** `getLength()`	Result of getLength() is 83h	
6	**Call the post() method** `getLength()`	A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	
NOTE: Test case 3 is limited to 253 and not 256 because the current 3GPP TS 43.019 [7] is not clear enough on this point. So this test allows the two possible implementations.			

6.2.5.4.4 Test Coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5
C1	6

6.2.5.5 Method copy

Test Area Reference: API_2_ERH_COPY_BSS

6.2.5.5.1 Conformance requirement

The method with following header 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
```

6.2.5.5.1.1 Normal execution

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

- CRRN2: returns dstOffset + dstLength.

6.2.5.5.1.2 Parameter errors

- 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 greater 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.

6.2.5.5.1.3 Context errors

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

6.2.5.5.2 Test Suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_COPY_BSS_1.scr
Test Applet:	API_2_ERH_COPY_BSS_1.java
Load Script:	API_2_ERH_COPY_BSS_1.ldr
Cleanup Script:	API_2_ERH_COPY_BSS_1.clr
Parameter File:	API_2_ERH_COPY_BSS_1.par

6.2.5.5.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**appendTLV with value length of 7** **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`
	`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`	Result of copy() is 12	

	`dstOffset = 3`
	`dstLength = 9`		

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	

14	**Successful call, copy with length =0**
	`dstBuffer.length = 15`
	`dstOffset = 15`
	`dstLength = 0`
		Result of copy() is 15	

15	**Call the post() method**
	`Call copy():`
	`dstBuffer.length = 10`
	`dstOffset = 0`
	`dstLength = 0`
		A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

6.2.5.5.4 Test Coverage

CRR number	Test case number
N1	9, 11, 13
N2	8, 10, 12, 14
P1	1
P2	2, 3, 4, 5, 6
P3	7
C1	15

6.2.5.6 Method findTLV

Test Area Reference: API_2_ERH_FINDBB

6.2.5.6.1 Conformance requirement

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

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

6.2.5.6.1.1 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, the current TLV is no longer defined and TLV_NOT_FOUND is returned.
- CRRN5: The search method is comprehension required flag independent.

6.2.5.6.1.2 Parameter errors

- 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.

6.2.5.6.1.3 Context errors

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

6.2.5.6.2 Test Suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_FINDBB_1.scr
Test Applet:	API_2_ERH_FINDBB_1.java
Load Script:	API_2_ERH_FINDBB_1.ldr
Cleanup Script:	API_2_ERH_FINDBB_1.clr
Parameter File:	API_2_ERH_FINDBB_1.par

6.2.5.6.3 Test procedure

Id	Description	API Expectation	APDU Expectation

1	**append the handler with TLVs:**
	`81 03 11 22 33`
	`82 02 99 77`		

	Invalid input parameter `Occurrence = 0`	ToolkitException.BAD_INPUT_PARAMETER is thrown	
2			

	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	**Select a TLV (tag 02h)**		

	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	

14	**Call the post() method then Search 1st TLV**
	`Tag = 81h`
	`Occurrence = 1`
		A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

6.2.5.6.4 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	14

6.2.5.7 Method getValueLength

Test Area Reference: API_2_ERH_GVLE

6.2.5.7.1 Conformance requirement

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

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

6.2.5.7.1.1 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.

6.2.5.7.1.2 Parameter errors

No requirements.

6.2.5.7.1.3 Context errors

- 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.

6.2.5.7.2 Test Suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_GVLE_1.scr
Test Applet:	API_2_ERH_GVLE_1.java
Load Script:	API_2_ERH_GVLE_1.ldr
Cleanup Script:	API_2_ERH_GVLE_1.clr
Parameter File:	API_2_ERH_GVLE_1.par

6.2.5.7.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**appendTLV 02 02 02 02** **findTLV with TAG 03**		
	getValueLength()	ToolkitException.UNAVAILABLE_ ELEMENT is thrown	
2	**appendTLV with TAG 0D and length 00**		
	Search TLV 0Dh		
	getValueLength()	Result is 00h	
3	**Clear the handler and append TLV with TAG**		

	0D and length 02		
	Search TLV 0Dh		
	`getValueLength()`	Result is 02h	
4	**Clear the handler and append TLV with TAG 0D and length 0x7F**		
	Search TLV 0Dh		
	getValueLength()	Result is 7Fh	
5	**Clear the handler and append TLV with TAG 0D and length 0x80**		
	Search TLV 0Dh		
	`getValueLength()`	Result is 80h	
6	**Clear the handler and append TLV with TAG 0D and length 0xF1**		
	Search TLV 0Dh		
	`getValueLength()`	Result is F1h	
7	**Call the post() method** `getValueLength()`	A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

6.2.5.7.4 Test Coverage

CRR number	Test case number
N1	2, 3, 4, 5, 6
C1	7
C2	1

6.2.5.8 Method getValueByte

Test Area Reference: API_2_ERH_GVBYS

6.2.5.8.1 Conformance requirement

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

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

6.2.5.8.1.1 Normal execution

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

6.2.5.8.1.2 Parameter errors

- 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.

6.2.5.8.1.3 Context errors

- 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.

6.2.5.8.2 Test Suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_GVBYS_1.scr
Test Applet:	API_2_ERH_GVBYS_1.java
Load Script:	API_2_ERH_GVBYS_1.ldr
Cleanup Script:	API_2_ERH_GVBYS_1.clr
Parameter File:	API_2_ERH_GVBYS_1.par

6.2.5.8.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**appendTLV 82 02 81 82, appendTLV 81 03 11 22 FE findTLV with TAG 03**		
	`getValueByte(0x00)`	ToolkitException.UNAVAILABLE_ ELEMENT is thrown	
2	**Search TLV 01h**		
	`getValueByte(0x03)`	ToolkitException.OUT_OF_TLV_ BOUNDARIES is thrown	
3	**Search TLV 01h**		
	`getValueByte(0x02)`	Result is FEh	
4	**Search TLV 02h**		
	`getValueByte(0x00)`	Result is 81h	
5	**appendTLV with TAG 0D, Length 0x7E, Value: 00, 01, ..., 7D**		
	`getValueByte(0x07D)`	Result is 7Dh	
6	**clear the handler, appendTLV with TAG 0D, Length 0x80, Value: 00, 01, ..., 7F**		
	`getValueByte(0x07E)`	Result is 7Eh	
7	`getValueByte(0x07F)`	Result is 7Fh	
8	**clear the handler, appendTLV with TAG 0D, Length 0xF1, Value: 00, 01, ..., F0**		
	`getValueByte(0x0F0)`	Result is F0h	
9	**Call the post() method** `getValueByte(0)`	A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

6.2.5.8.4 Test Coverage

CRR number	Test case number
N1	3, 4, 5, 6, 7, 8
P1	2
C1	9
C2	1

6.2.5.9 Method copyValue

Test Area Reference: API_2_ERH_CPYVS_BSS

6.2.5.9.1 Conformance requirement

The method with following header 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
```

6.2.5.9.1.1 Normal execution

- CRRN1: copies a part of the last TLV element which has been found, into a destination. buffer.
- CRRN2: returns dstOffset + dstLength.

6.2.5.9.1.2 Parameter errors

- 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.

6.2.5.9.1.3 Context errors

- 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.

6.2.5.9.2 Test Suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_CPYVS_BSS_1.scr
Test Applet:	API_2_ERH_CPYVS_BSS_1.java
Load Script:	API_2_ERH_CPYVS_BSS_1.ldr
Cleanup Script:	API_2_ERH_CPYVS_BSS_1.clr
Parameter File:	API_2_ERH_CPYVS_BSS_1.par

6.2.5.9.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**appendTLV with TAG: 0D and length 16 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`
	`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`	ArrayIndexOutOfBoundsExceptio	

Id	Description	API Expectation	APDU Expectation

	`dstOffset = 0`
	`dstLength = -1`
		n is thrown	

7	**clear the handler, appendTLV with TAG: 0D and length 6** **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	**clear the handler, appendTLV with TAG: 0D and value: 04 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	

16	**Successful call, copyValue with length =0**
	`dstBuffer.length = 20`
	`dstOffset = 20`
	`dstLength = 0`
		Result of copyValue() is 20	

17	**Call post() method then copyValue()**
	`dstBuffer.length = 20`
	`dstOffset = 0`
	`dstLength = 0`
		A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

6.2.5.9.4 Test Coverage

CRR number	Test case number
N1	13, 15
N2	12, 14, 16
P1	1
P2	2, 3, 4, 5, 6

P3	7, 8, 9, 10
C1	17
C2	11

6.2.5.10 Method compareValue

Test Area Reference: API_2_ERH_CPRVS_BSS

6.2.5.10.1 Conformance requirement

The method with following header 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
```

6.2.5.10.1.1 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.

6.2.5.10.1.2 Parameter errors

- 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.

6.2.5.10.1.3 Context errors

- 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.

6.2.5.10.2 Test Suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_CPRVS_BSS_1.scr
Test Applet:	API_2_ERH_CPRVS_BSS_1.java
Load Script:	API_2_ERH_CPRVS_BSS_1.ldr
Cleanup Script:	API_2_ERH_CPRVS_BSS_1.clr
Parameter File:	API_2_ERH_CPRVS_BSS_1.par

6.2.5.10.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**appendTLV with TAG: 0D and length 16** **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	**appendTLV with TAG: 0D and length 6**
	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	**appendTLV with TAG: 0D and value: 04 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	

Id	Description	API Expectation	APDU Expectation

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	

18	**Successful call, compareValue with length =0**
	`compareBuffer.length = 20`
	`compareOffset = 15`
	`compareLength = 0`
		Result of compareValue() is 0	

19	**Call post() method then compareValue()**
	`compareBuffer.length = 20`
	`compareOffset = 0`
	`compareLength = 0`
		A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

6.2.5.10.4 Test Coverage

CRR number	Test case number
N1	12, 15, 18
N2	13, 16
N3	14, 17
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	19
C2	11

6.2.5.11 Method findAndCopyValue(byte tag, byte[] dstBuffer, short valueOffset)

Test Area Reference: API_2_ERH_FACYB_BS

6.2.5.11.1 Conformance requirement

The method with following header 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
```

6.2.5.11.1.1 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.

6.2.5.11.1.2 Parameter errors

- CRRP1: if dstBuffer is null NullPointerException shall be thrown.
- CRRP2: if dstOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

6.2.5.11.1.3 Context errors

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

6.2.5.11.2 Test Suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_FACYB_BS_1.scr
Test Applet:	API_2_ERH_FACYB_BS_1.java
Load Script:	API_2_ERH_FACYB_BS_1.ldr
Cleanup Script:	API_2_ERH_FACYB_BS_1.clr
Parameter File:	API_2_ERH_FACYB_BS_1.par

6.2.5.11.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialise the handler**		
	findAndCopyValue() with a null dstBuffer	NullPointerException is thrown	
2	**appendTLV with TAG: 0D and length 16** **Select Text String TLV**		

	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	**dstOffset + length >dstBuffer.length**
	`dstBuffer.length = 20`
	`dstOffset = 5`
		ArrayIndexOutOfBoundsException is thrown	

5	**length > dstBuffer.length**
	`dstBuffer.length = 15`
	`dstOffset = 0`
		ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation

6	**clear the handler, appendTLV with TAG 02 and Length 02**
	Select a TLV (tag 02h)
	`findAndCopyValue()`
	`tag = 03h`
	Call the getValueLength() method
		ToolkitException.UNAVAILABLE_ ELEMENT is thrown
		ToolkitException.UNAVAILABLE_ ELEMENT is thrown.	

7	**appendTLV with TAG: 0D and value: 04 00 01 … 0F**
	Successful call
	`Tag = 0Dh`
	`dstBuffer.length = 17`
	`dstOffset = 0`
		Result of findAndCopyValue() 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 findAndCopyValue() 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	**clear the handler, appendTLV with TAG: 0D and value: 04 00 01 … 0F**
	append a 2nd Text String TLV
	Successful call
	`tag = 0Dh`
	`dstBuffer.length = 17`
	`dstOffset = 0`
		Result of findAndCopyValue() is 17	

12	**Compare buffer** `buffer = 04 00 01 … 0F`	Result is 00h	

13	**clear the handler, appendTLV with TAG: 0D and value: 04 00 01 … 0F**
	Successful call (with tag 8Dh)
	`tag = 8Dh`
	`dstBuffer.length = 17`
	`dstOffset = 0`
		Result of findAndCopyValue() 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 findAndCopyValue() is 16	

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

17	**Call post() method then findAndCopyValue()**
	`tag = 8Fh`
	`dstBuffer.length = 0`
	`dstOffset = 0`
		A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

6.2.5.11.4 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	17

6.2.5.12 Method findAndCopyValue(byte tag, byte occurrence, short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)

Test Area Reference: API_2_ERH_FACYBBS_BSS

6.2.5.12.1 Conformance requirement

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

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

6.2.5.12.1.1 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.

6.2.5.12.1.2 Parameter errors

- 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.

6.2.5.12.1.3 Context errors

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

6.2.5.12.2 Test Suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_FACYBBS_BSS_1.scr
Test Applet:	API_2_ERH_FACYBBS_BSS_1.java
Load Script:	API_2_ERH_FACYBBS_BSS_1.ldr
Cleanup Script:	API_2_ERH_FACYBBS_BSS_1.clr
Parameter File:	API_2_ERH_FACYBBS_BSS_1.par

6.2.5.12.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialise the handler** findAndCopyValue() with a null dstBuffer	NullPointerException is thrown	

2	**appendTLV with TAG: 0D and length 16**
	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	**appendTLV with TAG: 0D and length 6**
	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	**clear the handler, appendTLV with TAG 02 and Length 02**
	Select a TLV (tag 02h)
	`findAndCopyValue()`
	`tag = 0Dh`
	`occurrence = 2`
	`Call the getValueLength() method`
		ToolkitException.UNAVAILABLE_ELEMENT is thrown
		ToolkitException.UNAVAILABLE_ELEMENT is thrown.	

12	**clear the handler and appendTLV with TAG: 0D and value: 04 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	

Id	Description	API Expectation	APDU Expectation

14	**initialise dstBuffer**
	`dstBuffer = 55 55 … 55`
	Successful call
	`tag = 0Dh, occurrence = 1`
	`valueOffset = 2`
	`dstBuffer.length = 20`
	`dstOffset = 3`
	`dstLength = 12`
		Result of findAndCopyValue() 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 = 20`
	`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	**clear the handler and appendTLV with TAG: 0D and value: 04 00 01 … 0F**
	Successful call (with tag 8Dh)
	`tag = 8Dh`
	`occurrence = 1`
	`valueOffset = 0`
	`dstBuffer.length = 17`
	`dstOffset = 0`
	`dstLength = 17`
		Result of findAndCopyValue () 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`
	`occurrence = 1`
	`valueOffset = 0`
	`dstBuffer.length = 16`
	`dstOffset = 0`
	`dstLength = 16`
		Result of findAndCopyValue () is 16	

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

24	**Successful call, findAndCopyValue with length =0**
	`dstBuffer.length = 16`
	`dstOffset = 16`
	`dstLength = 0`
		Result of findAndCopyValue () is 16	

25	**Call post() method then findAndCopyValue()**
	`dstBuffer.length = 16`
	`dstOffset = 0`
	`dstLength = 0`
		A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

6.2.5.12.4 Test Coverage

CRR number	Test case number

N1	13, 15, 17, 19
N2	11
N3	12, 14, 16, 18, 24
N4	20, 21, 22, 23
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	25

6.2.5.13 Method findAndCompareValue(byte tag, byte[] compareBuffer, short compareOffset)

Test Area Reference: API_2_ERH_FACRB_BS

6.2.5.13.1 Conformance requirement

The method with following header 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
```

6.2.5.13.1.1 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.

6.2.5.13.1.2 Parameter errors

- CRRP1: if compareBuffer is null NullPointerException shall be thrown.
- CRRP2: if compareOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

6.2.5.13.1.3 Context errors

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

6.2.5.13.2 Test Suite files

Specific triggering: Unrecognized Envelope

Test Script: API_2_ERH_FACRB_BS_1.scr

Test Applet: API_2_ERH_FACRB_BS_1.java

Load Script: API_2_ERH_FACRB_BS_1.ldr

Cleanup Script: API_2_ERH_FACRB_BS_1.clr

Parameter File: API_2_ERH_FACRB_BS_1.par

6.2.5.13.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**appendTLV with TAG: 0D and length 16** findAndCompareValue() with a null dstBuffer and tag 0Dh	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	**compareOffset + length > compareBuffer.length**
	`compareBuffer.length = 20`
	`compareOffset = 5`
		ArrayIndexOutOfBoundsException is thrown	

5	**length > compareBuffer.length**
	`compareBuffer.length = 15`
	`compareOffset = 0`
		ArrayIndexOutOfBoundsException is thrown	

6	**clear the handler, appendTLV with TAG 02 and Length 02**
	Select a TLV (tag 02h)
	`findAndCompareValue()`
	`tag = 03h`
		ToolkitException.UNAVAILABLE_ELEMENT is thrown	

7	**Verify current TLV** `getValueLength()`	ToolkitException.UNAVAILABLE_ELEMENT is thrown.	

8	**clear the handler and appendTLV with TAG: 0D and value: 04 00 01 … 0F**
	Initialise compareBuffer
	`compareBuffer =`
	`04 00 01 … 0F`
	Compare buffers
	`tag = 0Dh`
	`compareOffset = 0`
		Result is 00h	

9	**Verify current TLV** `getValueLength()`	Result is 17	

10	**Initialise compareBuffer**
	`compareBuffer =`
	`04 00 01 … 10`
	Compare buffers with same parameters
		Result is -1	

11	**Initialise compareBuffer**
	`compareBuffer =`
	`03 00 01 … 0F`
	Compare buffers with same parameters
		Result is +1	

12	**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	**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	

Id	Description	API Expectation	APDU Expectation

14	**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	

15	**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	

16	**clear the handler and appendTLV with TAG: 0D and value: 04 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	

17	**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	

18	**Call post() method then findAndCompareValue()**
	`tag = 8Fh`
	`compareBuffer.length = 0`
	`compareOffset = 0`
		A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

6.2.5.13.4 Test Coverage

CRR number	Test case number
N1	6,7
N2	7,9
N3	8, 13, 12
N4	10, 14
N5	11, 15
N6	17, 16
P1	1
P2	2, 3, 4, 5
C1	18

6.2.5.14 Method findAndCompareValue(byte tag, byte occurrence, short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)

Test Area Reference: API_2_ERH_FACRBBS_BSS

6.2.5.14.1 Conformance requirement

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

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

6.2.5.14.1.1 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.

6.2.5.14.1.2 Parameter errors

- 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. occurrence = 0) an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.BAD_INPUT_PARAMETER.

6.2.5.14.1.3 Context errors

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

6.2.5.14.2 Test Suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_FACRBBS_BSS_1.scr
Test Applet:	API_2_ERH_FACRBBS_BSS_1.java
Load Script:	API_2_ERH_FACRBBS_BSS_1.ldr
Cleanup Script:	API_2_ERH_FACRBBS_BSS_1.clr
Parameter File:	API_2_ERH_FACRBBS_BSS_1.par

6.2.5.14.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialise the handler** **findAndCompareValue() with a null compareBuffer**	NullPointerException is thrown	

Id	Description	API Expectation	APDU Expectation

2	**clear the handler and appendTLV with TAG: 0D and value: 04 00 01 … 0F**
	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	**clear the handler and appendTLV with TAG and length of 6**
	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	**appendTLV with TAG 02 and length 02**
	Select a TLV (tag 02h)
	`findAndCompareValue()`
	`tag = 0Dh`
	`occurrence = 2`
		ToolkitException.UNAVAILABLE_ELEMENT is thrown	

13	**Verify current TLV** `getValueLength()`	ToolkitException.UNAVAILABLE_ELEMENT is thrown.	

14	**clear the handler and appendTLV with TAG: 0D and value: 04 00 01 … 0F**
	Initialise compareBuffer
	`compareBuffer =`
	`04 00 01 … 0F`
	findAndCompareValue()
	`tag = 0Dh, occurrence = 1`
	`valueOffset = 0`
	`compareOffset = 0`
	`compareLength = 17`
		Result is 00h	

15	**Verify current TLV** `getValueLength()`	Result is 17	

Id	Description	API Expectation	APDU Expectation
16	**Initialise compareBuffer** compareBuffer = 04 00 01 … 10 **Compare buffers with same parameters**	Result is -1	
17	**Initialise compareBuffer** compareBuffer = 03 00 01 … 0F **Compare buffers with same parameters**	Result is +1	
18	**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	
19	**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	
20	**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	

21	**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	

22	**Initialise compareBuffer**
	compareBuffer =
	00 11 22 33 44 55
	findAndCompareValue()
	tag = 0Dh, occurrence = 2
	valueOffset = 0
	compareOffset = 0
	compareLength = 6
		Result is 00h	

23	**Initialise compareBuffer**
	compareBuffer =
	00 11 22 33 44 66
	findAndCompareValue()
	tag = 0Dh, occurrence = 2
	valueOffset = 0
	compareOffset = 0
	compareLength = 6
		Result is -1	

24	**clear the handler and appendTLV with TAG: 0D and value: 04 00 01 … 0F**
	Initialise compareBuffer
	compareBuffer = 04 00 01 … 0F
	Successful call (with tag 8Dh)
	tag = 8Dh, occurrence = 1
	valueOffset = 0
	compareBuffer.length = 17
	compareOffset = 0
	compareLength = 17
		Result is 00h	

Id	Description	API Expectation	APDU Expectation

25	**Append tag 0Fh**
	`buffer = 00 01 … 0F`
	Initialise compareBuffer
	`compareBuffer = 00 01 … 0F`
	Successful call (with tag 8Fh)
	`tag = 8Fh, occurrence = 1`
	`valueOffset = 0`
	`compareBuffer.length = 16`
	`compareOffset = 0`
	`compareLength = 16`
		Result is 00h	

26	**Successful call, findAndCompareValue with length =0**
	`CompareBuffer.length = 16`
	`compareOffset = 16`
	`compareLength = 0`
		Result of findAndCompareValue () is 00	

27	**Call post() method then findAndCompareValue()**
	`CompareBuffer.length = 16`
	`compareOffset = 0`
	`compareLength = 0`
		A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

6.2.5.14.4 Test Coverage

CRR number	Test case number
N1	12,13
N2	15,13
N3	14, 18, 22, 21, 26
N4	16, 19, 23
N5	17, 19
N6	25, 24
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
P4	11
C1	27

6.2.5.15 Method appendArray

Test Area Reference: API_2_ERH_APDA_BSS

6.2.5.15.1 Conformance requirement:

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

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

6.2.5.15.1.1 Normal execution

- CRRN1: appends a buffer into the EditHandler buffer.
- CRRN2: a successful append does not modify the TLV selected.

6.2.5.15.1.2 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.

6.2.5.15.1.3 Context errors

- 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.

6.2.5.15.2 Test suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_APDA_BSS_1.scr
Test Applet:	API_2_ERH_APDA_BSS_1.java
Load Script:	API_2_ERH_APDA_BSS_1.ldr
Cleanup Script:	API_2_ERH_APDA_BSS_1.clr
Parameter File:	API_2_ERH_APDA_BSS_1.par

6.2.5.15.3 Test procedure

Id	Description	API Expectation	APDU Expectation
	Initialise the envelope response handler with a TLV of length 1		
1	**Null buffer**	NullPointerException is thrown	

2	**offset ≥ buffer.length**
	`buffer.length = 5`
	`offset = 5`
	`length = 1`
		ArrayIndexOutOfBoundsException is thrown	

3	**offset < 0**
	`buffer.length = 5`
	`offset = -1`
	`length = 1`
		ArrayIndexOutOfBoundsException is thrown	

4	**length > buffer.length**
	`buffer.length = 5`
	`offset = 0`
	`length = 6`
		ArrayIndexOutOfBoundsException is thrown	

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

6	**length < 0**
	`buffer.length = 5`
	`offset = 0`
	`length = -1`
		ArrayIndexOutOfBoundsException is thrown	

7	**Handler overflow**
	`buffer.length = 256`
	`offset = 0`
	`length = 256`
		ToolkitException.HANDLER_OVERFLOW is thrown	

8	**append the handler with TLVs:**
	`81 03 11 22 33`
	`82 02 99 77`
	findTLV 0x81
	Successful call
	`buffer = FF FE … F8`
	`offset = 0`
	`length = 8`
	Verify Current TLV: Call getValueLength()
		Result is 03h	

9	**Clear the handler**
	Successful call
	`buffer = FF FE … F8`
	`offset = 0`
	`length = 8`
	Call copy() method
	Compare handler
	`compareBuffer = FF FE … F8`
		Result is 00h	

Id	Description	API Expectation	APDU Expectation

10	**Successful call**
	`buffer = 00 01 … 07`
	`offset = 2`
	`length = 6`
	Call copy() method
	Compare handler
	`compareBuffer = FF FE … F8 02 03 … 07`
		Result is 00h	

11	**Successful call**
	`buffer = 11 22 … 88`
	`offset = 2`
	`length = 4`
	Call copy() method
	Compare handler
	`compareBuffer = FF FE … F8 02 03 … 07 33 44 55 66`
		Result is 00h	

12	**Call post() method then appendArray()**
	`buffer = 11 22 … 88`
	`offset = 2`
	`length = 4`
		A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

6.2.5.15.4 Test Coverage

CRR number	Test case number
N1	9, 10, 11
N2	8
P1	1
P2	2, 3, 4, 5, 6
C1	7
C2	12

6.2.5.16 Method appendTLV(byte tag, byte value)

Test Area Reference: API_2_ERH_APTLBB

6.2.5.16.1 Conformance requirement:

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

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

6.2.5.16.1.1 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.

6.2.5.16.1.2 Parameters error

No requirements

6.2.5.16.1.3 Context errors

- 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.

6.2.5.16.2 Test suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_APTLBB_1.scr
Test Applet:	API_2_ERH_APTLBB_1.java
Load Script:	API_2_ERH_APTLBB_1.ldr
Cleanup Script:	API_2_ERH_APTLBB_1.clr
Parameter File:	API_2_ERH_APTLBB_1.par

6.2.5.16.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call appendArray()** `length = 253` **Handler Overflow: Call twice the appendTLV()method**	ToolkitException.HANDLER_OVERFLOW is thrown by one of the two.	

2	**append the handler with TLVs:**
	`81 03 11 22 33`
	`82 02 99 77`
	Select Command Details TLV
	Call the appendTLV() method
	Verify Current TLV: Call getValueLength()
		Result is 03h	

3	**Clear the handler**
	Successful call
	`tag = 84h`
	`value = 00h`
	Call copy() method
	Compare handler
	`compareBuffer = 84 01 00`
		Result is 00h	

4	**Successful call**
	`tag = 01h`
	`value = FEh`
	Call copy() method
	Compare handler
	`compareBuffer = 84 01 00 01 01 FE`
		Result is 00h	

5	**Call post() method then appendTLV()**
	`tag = 01h`
	`value = FEh`
		A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

NOTE: Test case 1 call twice appendTLV because the current 3GPP TS 43.019 [7] is not clear enough on this point. So this test allows the two possible implementations.			

6.2.5.16.4 Test Coverage

CRR number	Test case number
N1	3, 4
N2	2
C1	1
C2	5

6.2.5.17 Method appendTLV(byte tag, byte value1, byte value2)

Test Area Reference: API_2_ERH_APTLBBB

6.2.5.17.1 Conformance requirements:

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

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

6.2.5.17.1.1 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.

6.2.5.17.1.2 Parameters error

No requirements

6.2.5.17.1.3 Context errors

- 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.

6.2.5.17.2 Test suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_APTLBBB_1.scr
Test Applet:	API_2_ERH_APTL BBB_1.java
Load Script:	API_2_ERH_APTL BBB_1.ldr
Cleanup Script:	API_2_ERH_APTLBBB_1.clr
Parameter File:	API_2_ERH_APTLBBB_1.par

6.2.5.17.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the appendArray with length of 253** **Handler Overflow: Call the appendTLV() method**	ToolkitException.HANDLER_OVERFLOW is thrown	

2	**clear the handler, append the handler with TLVs:**
	`81 03 11 22 33`
	`82 02 99 77`
	Select Command Details TLV
	Call the appendTLV() method
	Verify Current TLV: Call getValueLength()
		Result is 03h	

3	**Clear the handler**
	Successful call
	`tag = 84h`
	`value1 = 00h`
	`value2 = 01h`
	Call copy() method
	Compare handler
	`compareBuffer = 84 02 00 01`
		Result is 00h	

4	**Successful call**
	`tag = 01h`
	`value1 = FEh`
	`value2 = FDh`
	Call copy() method
	Compare handler
	`compareBuffer = 84 02 00 01 01 02 FE FD`
		Result is 00h	

5	**Call post() method then appendTLV()**
	`tag = 01h`
	`value1 = FEh`
		A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

Id	Description	API Expectation	APDU Expectation
	value2 = FDh		

6.2.5.17.4 Test Coverage

CRR number	Test case number
N1	3, 4
N2	2
C1	1
C2	5

6.2.5.18 Method appendTLV(byte tag, byte[ ] value, short valueoffset, short valuelength)

Test Area Reference: API_2_ERH_APTLB_BSS

6.2.5.18.1 Conformance requirement:

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

```
public void appendTLV(byte tag,
                      byte[] value,
                      short valueOffset,
                      short valueLength)
               throws java.lang.NullPointerException,
                      java.lang.ArrayIndexOutOfBoundsException,
                      ToolkitException
```

6.2.5.18.1.1 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.

6.2.5.18.1.2 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.

6.2.5.18.1.3 Context errors

- 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.

6.2.5.18.2 Test suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_APTLB_BSS_1.scr
Test Applet:	API_2_ERH_APTLB_BSS_1.java
Load Script:	API_2_ERH_APTLB_BSS_1.ldr

Cleanup Script: API_2_ERH_APTLB_BSS_1.clr

Parameter File: API_2_ERH_APTLB_BSS_1.par

6.2.5.18.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Null value**	NullPointerException is thrown	

2	**valueOffset ≥ value.length**
	`value.length = 5`
	`valueOffset = 5`
	`valueLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

3	**valueOffset < 0**
	`value.length = 5`
	`valueOffset = -1`
	`valueLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

4	**valueLength > value.length**
	`value.length = 5`
	`valueOffset = 0`
	`valueLength = 6`
		ArrayIndexOutOfBoundsException is thrown	

5	**ValueOffset + valueLength > value.length**
	`value.length = 5`
	`valueOffset = 3`
	`valueLength = 3`
		ArrayIndexOutOfBoundsException is thrown	

6	**valueLength < 0**
	`value.length = 5`
	`valueOffset = 0`
	`valueLength = -1`
		ArrayIndexOutOfBoundsException is thrown	

7	**Handler overflow**
	`value.length = 254`
	`valueOffset = 0`
	`valueLength = 254`
		ToolkitException.HANDLER_OVERFLOW is thrown	

8	**Bad parameter**
	`value.length = 256`
	`valueOffset = 0`
	`valueLength = 256`
		ToolkitException.BAD_INPUT_PARAMETER is thrown	

9	**clear the handler, append the handler with TLVs:**
	`81 03 11 22 33`
	`82 02 99 77`
	Select Command Details TLV
	Successful call
	`tag = 04`
	`value = FF FE … F8`
	`valueOffset = 0`
	`valueLength = 8`
	Verify Current TLV: Call getValueLength()
		Result is 03h	

10	**Clear the handler**
	Successful call
	`tag = 04`
	`value = FF FE … F8`
	`valueOffset = 0`
	`valueLength = 8`
	Call copy() method
	Compare handler
	`CompareBuffer = 04 08 FF FE … F8`
		Result is 00	

11	**Successful call**
	`tag = 85h`
	`value = 00 01 … 07`
	`valueOffset = 2`
	`valueLength = 6`
	Call copy() method
	Compare handler
	`compareBuffer = 04 08 FF FE … F8 85 06 02 03 … 07`
		Result is 00	

Id	Description	API Expectation	APDU Expectation

12	**Successful call**
	`tag = 01`
	`value = 11 22 … 88`
	`valueOffset = 2`
	`valueLength = 4`
	Call copy() method
	Compare handler
	`compareBuffer = 04 08 FF FE … F8 85 06 02 03 … 07 01 04 33 44 55 66`
		Result is 00	

13	**Clear the handler**
	Successful call
	`tag = 04`
	`value = 00 01 … 7F`
	`valueOffset = 0`
	`valueLength = 80h`
	Call copy() method
	Compare handler
	`compareBuffer = 04 81 80 00 01…7F`
		Result is 00	

14	**Call post() method then appendTLV()**
	`tag = 04`
	`value = 00 01 … 7F`
	`valueOffset = 0`
	`valueLength = 80h`
		A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

6.2.5.18.4 Test Coverage

CRR number	Test case number
N1	10, 11, 12, 13
N2	9
P1	1
P2	2, 3, 4, 5, 6
C1	7
C2	14
C3	8

6.2.5.19 Method appendTLV(byte tag, byte value1, byte[ ] value2, short value2offset, short value2length)

Test Area Reference: API_2_ERH_APTLBB_BSS

6.2.5.19.1 Conformance requirement:

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

```
public void appendTLV(byte tag,
                      byte value1,
                      byte[] value2,
                      short value2Offset,
                      short value2Length)
               throws java.lang.NullPointerException,
                      java.lang.ArrayIndexOutOfBoundsException,
                      ToolkitException
```

6.2.5.19.1.1 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.

6.2.5.19.1.2 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.

6.2.5.19.1.3 Context errors

- 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.

6.2.5.19.2 Test suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_APTLBB_BSS_1.scr
Test Applet:	API_2_ERH_APTLBB_BSS_1.java
Load Script:	API_2_ERH_APTLBB_BSS_1.ldr
Cleanup Script:	API_2_ERH_APTLBB_BSS_1.clr
Parameter File:	API_2_ERH_APTLBB_BSS_1.par

6.2.5.19.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Null value2**	NullPointerException is thrown	

2	**value2Offset ≥ value2.length**
	`value2.length = 5`
	`value2Offset = 5`
	`value2Length = 1`
		ArrayIndexOutOfBoundsException is thrown	

3	**value2Offset < 0**
	`value2.length = 5`
	`value2Offset = -1`
	`value2Length = 1`
		ArrayIndexOutOfBoundsException is thrown	

4	**value2Length > value2.length**
	`value2.length = 5`
	`value2Offset = 0`
	`value2Length = 6`
		ArrayIndexOutOfBoundsException is thrown	

5	**value2Offset + value2Length > value2.length**
	`value2.length = 5`
	`value2Offset = 3`
	`value2Length = 3`
		ArrayIndexOutOfBoundsException is thrown	

6	**value2Length < 0**
	`value2.length = 5`
	`value2Offset = 0`
	`value2Length = -1`
		ArrayIndexOutOfBoundsException is thrown	

7	**Handler overflow**
	`value2.length = 254`
	`value2Offset = 0`
	`value2Length = 254`
		ToolkitException.HANDLER_OVERFLOW is thrown	

Id	Description	API Expectation	APDU Expectation

8	**Bad parameter**
	`value2.length = 256`
	`value2Offset = 0`
	`value2Length = 256`
		ToolkitException.BAD_INPUT_PARAMETER is thrown	

9	**clear the handler, append the handler with TLVs:**
	`81 03 11 22 33`
	`82 02 99 77`
	Select Command Details TLV
	Successful call
	`tag = 04`
	`value1 = 05`
	`value2 = FF FE … F8`
	`value2Offset = 0`
	`value2Length = 8`
	Verify Current TLV: Call getValueLength()
		Result is 03h	

10	**Clear the handler**
	Successful call
	`tag = 04`
	`value1 = 05`
	`value2 = FF FE … F8`
	`value2Offset = 0`
	`value2Length = 8`
	Call copy() method
	Compare handler
	`CompareBuffer = 04 09 05 FF FE … F8`
		Result is 00	

11	**Successful call**
	`tag = 85h`
	`value1 = 55h`
	`value2 = 00 01 … 07`
	`value2Offset = 2`
	`value2Length = 6`
	Call copy() method
	Compare handler
	`compareBuffer =`
	`04 09 05 FF FE … F8`
	`85 07 55 02 03 … 07`
		Result is 00	

12	**Successful call**
	`tag = 01`
	`value1 = 44h`
	`value2 = 11 22 … 88`
	`value2Offset = 2`
	`value2Length = 4`
	Call copy() method
	Compare handler
	`CompareBuffer =`
	`04 09 05 FF FE … F8`
	`85 07 55 02 03 … 07`
	`01 05 44 33 44 55 66`
		Result is 00	

13	**Clear the handler**
	Successful call
	`tag = 04`
	`value1 = 00`
	`value2 = 01 … 7F`
	`value2Offset = 0`
	`value2Length = 7Fh`
	Call copy() method
	Compare handler
	`compareBuffer = 04 81 80 00 01…7F`
		Result is 00	

14	**Call post() method then appendTLV()**
	`tag = 04`
	`value1 = 00`
	`value2 = 01 … 7F`
	`value2Offset = 0`
	`value2Length = 7Fh`
		A toolkit Exception with HANDLER_NOT_AVAILABLE reason is thrown.	

6.2.5.19.4 Test Coverage

CRR number	Test case number
N1	10, 11, 12, 13
N2	9
P1	1
P2	2, 3, 4, 5, 6
C1	7
C2	Does not apply for EnvelopeResponseHandler
C3	8

6.2.5.20 Method clear

Test Area Reference: API_2_ERH_CLER

6.2.5.20.1 Conformance requirement:

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

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

6.2.5.20.1.1 Normal execution

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

6.2.5.20.1.2 Parameters error

No requirements

6.2.5.20.1.3 Context errors

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

6.2.5.20.2 Test suite files

Specific triggering: Unrecognized Envelope

Test Script:	API_2_ERH_CLER_1.scr
Test Applet:	API_2_ERH_CLER_1.java
Load Script:	API_2_ERH_CLER_1.ldr
Cleanup Script:	API_2_ERH_CLER_1.clr
Parameter File:	API_2_ERH_CLER_1.par

6.2.5.20.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**append the handler with TLVs:** 81 03 11 22 33 82 02 99 77 Select Command Details TLV Call the getLength() method **Clear the handler** Call the getLength() method	Result of getLength() is not null Result of getLength() is 0	
2	**Call the getValueLength() method**	ToolkitException.UNAVAILABLE_ ELEMENT shall be thrown	
3	**Call post() method then clear()**	A toolkit Exception with HANDLER_NOT_AVAILABLE	

		reason is thrown.	

6.2.5.20.4 Test Coverage

CRR number	Test case number
N1	1, 2
C1	3

6.2.5.21 Method getCapacity

Test Area Reference: API_2_ERH_GCAP

6.2.5.21.1 Conformance Requirement:

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

```
Public byte getCapacity()
```

6.2.5.21.1.1 Normal execution

- CRRN1: The method shall return the maximum size of the Simple TLV list managed by the handler.

6.2.5.21.1.2 Context errors

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

6.2.5.21.2 Test suite files

Test Script:	API_2_ERH_GCAP_1.scr
Test Applet:	API_2_ERH_GCAP_1.java
Load Script:	API_2_ERH_GCAP_1.ldr
Cleanup Script:	API_2_ERH_GCAP_1.clr
Parameter File:	API_2_ERH_GCAP_1.par

6.2.5.21.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**EnvelopeResponseHandler available** 1- Send envelope SMS-PP Formatted 2- The applet calls getTheHandler() method 3- The applet calls getCapacity() method on the EnvelopeResponseHandler 4- The applet fills the handler with the maximum capacity using AppendTLV() method 5- The applet calls clear() method on the EnvelopeResponseHandler 6- The applet fills the handler with the maximum capacity plus one, using AppendTLV() method	1- Applet is triggered 2- No exception is thrown 3- No exception is thrown 4- No exception is thrown 5- No exception is thrown 6- HANDLER_OVERFLOW exception is thrown	

6.2.5.21.4 Test Coverage

CRR number	Test case number
N1	1
C1	Tested in Framework part: FWK_MHA_ERHD

6.2.6 Class MEProfile

6.2.6.1 Method check (byte index)

Test Area Reference: API_2_MEP_CHECB

6.2.6.1.1 Conformance requirement:

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

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

6.2.6.1.1.1 Normal execution

- CRRN1: The method checks a facility in the handset profile: returns true if supported and false otherwise.

6.2.6.1.1.2 Parameters error

No requirements.

6.2.6.1.1.3 Context errors

- CRRC1: The method shall throw ME_PROFILE_NOT_AVAILABLE ToolkitException if Terminal Profile data are not available

6.2.6.1.2 Test suite files

Specific triggering:

EVENT_STATUS_COMMAND

No Additional requirements for the GSM personalization:

Test Script:	API_2_MEP_CHECB_1.scr
Test Applet:	API_2_MEP_CHECB_1.java
Load Script:	API_2_MEP_CHECB_1.ldr
Cleanup Script:	API_2_MEP_CHECB_1.clr
Parameter File:	API_2_MEP_CHECB_1.par

6.2.6.1.3 Test procedure

Id	Description	API Expectation	APDU Expectation

1	**No Terminal Profile is registered**
	`Triggered by status command`
	`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	

6.2.6.1.4 Test Coverage

CRR number	Test case number
N1	2,3
C1	1

6.2.6.2 Method check (byte [ ] mask, short offset, short length)

Test Area Reference: API_2_MEP_CHEC_BSS

6.2.6.2.1 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
```

6.2.6.2.1.1 Normal execution

- CRRN1: The method checks all the facilities corresponding to bits set to 1 in the mask buffer: returns true if they are all supported and false if not.
- CRRN2: The method returns true if the length to check is 0.

6.2.6.2.1.2 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.

6.2.6.2.1.3 Context errors

No requirements.

6.2.6.2.2 Test suite files

Specific triggering:

UNFORMATTED_SMS_PP_UPD

No Additional requirements for the GSM personalization:

Test Script:	API_2_MEP_CHEC_BSS_1.scr
Test Applet:	API_2_MEP_CHEC_BSS_1.java
Load Script:	API_2_MEP_CHEC_BSS_1.ldr (the applet is loaded without INI after the reset (RST))
Cleanup Script:	API_2_MEP_CHEC_BSS_1.clr
Parameter File:	API_2_MEP_CHEC_BSS_1.par

6.2.6.2.3 Test procedure

Id	Description	API Expectation	APDU Expectation

1	**No Terminal Profile is registered**
	Triggered by unformatted 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**	ArrayIndexOutOfBoundsExc	

	mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F
	Offset = 17
		eption is thrown	

4	**Offset < 0**
	mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F
	Offset = -1
		ArrayIndexOutOfBoundsException is thrown	

5	**Length > mask.length**
	mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F
	Offset = 0
	Length = 18
		ArrayIndexOutOfBoundsException is thrown	

6	**Offset + length > mask.length**
	Mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F
	Offset = 9
	Length = 9
		ArrayIndexOutOfBoundsException is thrown	

7	**length = 0**
	mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F
	Offset = 0
	Length = 0
		true is returned	

8	**Check all the Terminal Profile**
	mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F
	Offset = 0
	Length = 16
		false is returned by the method because facility 15 is not supported	

9	**Check a part of the Terminal Profile**
	mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F
	Offset = 15
	Length = 2
		true is returned by the method: the 16 first facilities except facility 15 have been successfully checked	

10	**Check a part of the Terminal Profile**
	mask = 0x0080
	Offset = 0
	Length = 2
		false is returned by the method only facility 15 is checked and not supported.	

6.2.6.2.4 Test Coverage

CRR number	Test case number
N1	8, 9, 10
N2	7
P1	2
P2	3, 4, 5, 6
P3	1

6.2.6.3 Method check (short index)

Test Area Reference: API_2_MEP_CHECS

6.2.6.3.1 Conformance requirement:

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

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

6.2.6.3.1.1 Normal execution

- CRRN1: The method checks a facility in the handset profile: returns true if the facility is supported, false if facility is not supported, or if facility-index outside MEProfile data.

6.2.6.3.1.2 Parameters error

No requirements.

6.2.6.3.1.3 Context errors

- CRRC1: The method shall throw ME_PROFILE_NOT_AVAILABLE ToolkitException if Terminal Profile data are not available

6.2.6.3.2 Test suite files

Specific triggering:

EVENT_STATUS_COMMAND

No Additional requirements for the GSM personalization:

Test Script:	API_2_MEP_CHECS_1.scr
Test Applet:	API_2_MEP_CHECS_1.java
Load Script:	API_2_MEP_CHECS_1.ldr
Cleanup Script:	API_2_MEP_CHECS_1.clr
Parameter File:	API_2_MEP_CHECS_1.par

6.2.6.3.3 Test procedure

Id	Description	API Expectation	APDU Expectation

1	**No Terminal Profile is registered**
	`Triggered by status command`
	`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	
4	**Facility index is outside MEProfile data** `index = 0x0099`	false is returned by the method	

6.2.6.3.4 Test Coverage

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

6.2.6.4 Method getValue (short indexMSB, short indexLSB)

Test Area Reference: API_2_MEP_GVALSS

6.2.6.4.1 Conformance requirement:

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

```
public static short getValue(short indexMSB, short indexLSB)
                      throws ToolkitException
```

6.2.6.4.1.1 Normal execution

- CRRN1: The method returns the binary value of a parameter, delimited by two indexes, from the handset profile.

6.2.6.4.1.2 Parameters error

- CRRP1: The method shall throw BAD_INPUT_PARAMETER ToolkitException if (indexMSB >= indexLSB +16) or (indexMSB < indexLSB) or (indexMSB < 0) or (indexLSB < 0).

6.2.6.4.1.3 Context errors

- CRRC1: The method shall throw ME_PROFILE_NOT_AVAILABLE ToolkitException if Terminal Profile data are not available.

6.2.6.4.2 Test suite files

Specific triggering:

EVENT_STATUS_COMMAND

No Additional requirements for the GSM personalization:

Test Script:	API_2_MEP_GVALSS_1.scr
Test Applet:	API_2_MEP_GVALSS_1.java
Load Script:	API_2_MEP_GVALSS_1.ldr
Cleanup Script:	API_2_MEP_GVALSS_1.clr
Parameter File:	API_2_MEP_GVALSS_1.par

6.2.6.4.3 Test procedure

TP = FF 01 D2 F0 00 00 00 00 00 00 00 00 00 8D FF

Id	Description	API Expectation	APDU Expectation

1	**No Terminal Profile is registered**
	`Triggered by status command`
	`indexMSB = 15, indexLSB = 0`
		ME_PROFILE_NOT_AVAILABLE
		ToolkitException is thrown	

2	**Retrieve number of character down ME display in Terminal Profile which is 13** `indexMSB = 108, indexLSB = 104`	13 is returned by the method	

3	**Retrieve byte 3 and byte 4 from terminal profile.**
	`Byte 3 = 0xD2, Byte 4 = 0xF0`
	`indexMSB = 31, indexLSB = 16`
		0xF0D2 is returned by the method	

4	**indexMSB is negative** `indexMSB = 0xFFFF, indexLSB = 0xFFFD`	BAD_INPUT_PARAMETER ToolkitException is thrown	
5	**indexLSB is negative** `indexMSB = 0x0002, indexLSB = 0xFFFD`	BAD_INPUT_PARAMETER ToolkitException is thrown	
6	**indexMSB < indexLSB** `indexMSB = 0x0002, indexLSB = 0x0003`	BAD_INPUT_PARAMETER ToolkitException is thrown	
7	**indexMSB > indexLSB + 16** `indexMSB = 0x0021, indexLSB = 0x0010`	BAD_INPUT_PARAMETER ToolkitException is thrown	

6.2.6.4.4 Test Coverage

CRR number	Test case number
N1	2,3
P1	4, 5, 6, 7
C1	1

6.2.6.5 Method copy (short startOffset, byte[] dstBuffer, short dstOffset, short dstLength)

Test Area Reference: API_2_MEP_COPYS_BSS

6.2.6.5.1 Conformance requirement:

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

```
public static short copy(short startOffset, byte[] dstBuffer, short dstOffset, short dstLength)
                  throws ToolkitException
```

6.2.6.5.1.1 Normal execution

- CRRN1: The method copies a part of the handset profile in a buffer.

- CRRN2: The method returns dstOffset + dstLength.

6.2.6.5.1.2 Parameters error

- CRRP1: if dstBuffer is null NullPointerException is thrown.
- CRRP2 : *If* dstOffset *or* dstLength *parameter is negative an* ArrayIndexOutOfBoundsException *exception is thrown and no copy is performed*
- CRRP3: *If* dstOffset+dstLength *is greater than* dstBuffer.length, *the length of the* dstBuffer *array an* ArrayIndexOutOfBoundsException *exception is thrown and no copy is performed*

6.2.6.5.1.3 Context errors

- CRRC1: The method shall throw ME_PROFILE_NOT_AVAILABLE ToolkitException if Terminal Profile data are not available.

6.2.6.5.2 Test suite files

Specific triggering:

EVENT_STATUS_COMMAND

No Additional requirements for the GSM personalization:

Test Script:	API_2_MEP_COPYS_BSS_1.scr
Test Applet:	API_2_MEP_COPYS_BSS_1.java
Load Script:	API_2_MEP_COPYS_BSS_1.ldr
Cleanup Script:	API_2_MEP_COPYS_BSS_1.clr
Parameter File:	API_2_MEP_COPYS_BSS_1.par

6.2.6.5.3 Test procedure

TP = FF 01 D2 F0 01 02 00 00 00 00 00 00 00 8D FF

Id	Description	API Expectation	APDU Expectation

1	**No Terminal Profile is registered**
	`Triggered by status command`
	`startOffset = 0`
	`dstBuffer.length = 6`
	`dstOffset = 0`
	`dstLength = 6`
		ME_PROFILE_NOT_AVAILABLE ToolkitException is thrown	

2	**dstBuffer is null**	NullPointerException is thrown	

3	**dstOffset ≥ dstBuffer.length**
	`startOffset = 0`
	`dstBuffer.length = 5`
	`dstOffset = 5`
	`dstLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation

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

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

6	**dstLength >dstBuffer.length**
	`startOffset = 0`
	`dstBuffer.length = 5`
	`dstOffset = 0`
	`dstLength = 6`
		ArrayIndexOutOfBoundsException is thrown	

7	**dstOffset + dstLength >dstBuffer.length**
	`startOffset = 0`
	`dstBuffer.length = 5`
	`dstOffset = 3`
	`dstLength = 3`
		ArrayIndexOutOfBoundsException is thrown	

8	**Successful call extreme values**
	`startOffset = 0`
	`dstBuffer.length = 6`
	`dstOffset = 0`
	`dstLength = 6`
		Result of copy() is 6	

9	**Successful call any values**
	`startOffset = 1`
	`dstBuffer.length = 20`
	`dstOffset = 3`
	`dstLength = 4`
		Result of copy() is 7	

10	**Successful call, copy with length =0**
	`startOffset = 0`
	`dstBuffer.length = 20`
	`dstOffset = 20`
	`dstLength = 0`
		Result of copy() is 20	

11	**Value outside MEProfile data available**
	`startOffset = 13`
	`dstBuffer.length = 6`
	`dstOffset = 0`
	`dstLength = 6`
		Result of copy() is 6	

6.2.6.5.4 Test Coverage

CRR number	Test case number
N1	8, 9, 10, 11
N3	8, 9, 10, 11
P1	2
P2	4, 5
P3	3, 6, 7
C1	1

6.2.7 Class ProactiveHandler

6.2.7.1 Method getTheHandler

Test Area Reference: API_2_PAH_GTHD

6.2.7.1.1 Conformance requirement

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

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

6.2.7.1.1.1 Normal execution

- CRRN1: The method shall return the single system instance of the ProactiveHandler class.
- CRRN2: The EnvelopeHandler is a Temporary JCRE Entry Point Object

6.2.7.1.1.2 Parameter errors

No requirements.

6.2.7.1.1.3 Context errors

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

6.2.7.1.2 Test Suite files

Test Script:	API_2_PAH_GTHD_1.scr
Test Applet:	API_2_PAH_GTHD_1.java
Load Script:	API_2_PAH_GTHD_1.ldr
Cleanup Script:	API_2_PAH_GTHD_1.clr
Parameter File:	API_2_PAH_GTHD_1.par

6.2.7.1.3 Test procedure

Id	Description	API Expectation	APDU Expectation
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	

6.2.7.1.4 Test Coverage

CRR number	Test case number
N1	1, 2, 3
N2	Checked in Framework tests: FWK_API_HEPO (test case 3)
C1	Checked in Framework tests: FWK_MHA_PAHD

6.2.7.2 Method init

Test Area Reference: API_2_PAH_INITBBB

6.2.7.2.1 Conformance requirement

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

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

6.2.7.2.1.1 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.

- CRRN2: The Command number may take any value between 01h and FEh.
- CRRN3: The init() method clears the ProactiveHandler before initializing 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.

6.2.7.2.1.2 Parameter errors

No requirements.

6.2.7.2.1.3 Context errors

No requirements.

6.2.7.2.2 Test Suite files

Test Script:	API_2_PAH_INITBBB_1.scr
Test Applet:	API_2_PAH_INITBBB_1.java
Load Script:	API_2_PAH_INITBBB_1.ldr
Cleanup Script:	API_2_PAH_INITBBB_1.clr
Parameter File:	API_2_PAH_INITBBB_1.par

6.2.7.2.3 Test procedure

Id	Description	API Expectation	APDU Expectation

1	**Call the init() method**
	`type = 01h`
	`qualifier = 02h`
	`dstDevice = 03h`
	Copy ProactiveHandler in a byte array (source)
	Compare the byte array
	`reference =`
	`81h 03h xxh 01h 02h`
	`82h 02h 81h 03h`
		source and reference are identical	

2	**Verify the command number value**	01h-FEh	

3	**Call the init() method**
	`type = FFh`
	`qualifier = FEh`
	`destination = FDh`
	Copy ProactiveHandler in a byte array (source)
	Compare the byte array
	`reference =`
	`81h 03h xxh FFh FEh`
	`82h 02h 81h FDh`
		source and reference are identical	

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()	

6.2.7.2.4 Test Coverage

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

6.2.7.3 Method initDisplayText

Test Area Reference: API_2_PAH_INDTBB_BSS

6.2.7.3.1 Conformance requirement

The method with following header 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
```

6.2.7.3.1.1 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.
- CRRN6: If length is equal to zero, then the Text String TLV inserted in the command is a null text string TLV as defined in TS 11.14 [4].

6.2.7.3.1.2 Parameter errors

- 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.

6.2.7.3.1.3 Context errors

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

6.2.7.3.2 Test Suite files

Test Script:	API_2_PAH_INDTBB_BSS_1.scr
Test Applet:	API_2_PAH_INDTBB_BSS_1.java
Load Script:	API_2_PAH_INDTBB_BSS_1.ldr
Cleanup Script:	API_2_PAH_INDTBB_BSS_1.clr

Parameter File: API_2_PAH_INDTBB_BSS_1.par

6.2.7.3.3 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 = 5`
	`length = 0`
		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`
		No exception is thrown	

	Verify the command number value	Command number between 01h and FEh	

8	**Send the command**		DISPLAY TEXT Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextA"`

9	**Successful call, buffer is part of a buffer with the end part**
	Send the command
	`qualifier = 0`
	`dcs = 4`
	`buffer = "12TextB"`
	`offset = 2`
	`length = 5`
			DISPLAY TEXT Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextB"`

10	**Successful call, buffer is part of a buffer with the first part**
	Send the command
	`qualifier = 0`
	`dcs = 4`
	`buffer = "TextC12"`
	`offset = 0`
	`length = 5`
			DISPLAY TEXT Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextC"`

11	**Successful call, buffer is part of a buffer**
	Send the command
	`qualifier = 0`
	`dcs = 4`
	`buffer = "12TextD34"`
	`offset = 2`
	`length = 5`
			DISPLAY TEXT Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextD"`

12	**Successful call, qualifier = 81h**
	Send the command
	`qualifier = 81h`
	`dcs = 4`
	`buffer = "TextE"`
	`offset = 0`
	`length = 5`
			DISPLAY TEXT Proactive command
			`qualifier = 81h`
			`dcs = 4`
			`Text = "TextE"`

Id	Description	API Expectation	APDU Expectation

13	**Successful call, DCS=0 (7 bits)**
	Send the command
	`qualifier = 0`
	`dcs = 0`
	`buffer = "TextF"`
	`offset = 0`
	`length = 5`
			DISPLAY TEXT Proactive command
			`qualifier = 00h`
			`dcs = 0`
			`Text = "TextF"`

14	**Successful call, DCS=8 (UCS2)**
	Send the command
	`qualifier = 0`
	`dcs = 8`
	`buffer = "TextG"`
	`offset = 0`
	`length = 5`
			DISPLAY TEXT Proactive command
			`qualifier = 00h`
			`dcs = 8`
			`Text = "TextG"`

15	**Call the initDisplayText() method with any value**
	Then build and send a DISPLAY TEXT command
	`qualifier = 0`
	`dcs = 4`
	`buffer = "TextHTextH"`
	`offset = 0`
	`length = 10`
			DISPLAY TEXT Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextHTextH"`

16	**Successful call, text length is zero**
	Send the command
	`qualifier = 0`
	`dcs = 4`
	`buffer = "TextHTextH"`
	`offset = 0`
	`length = 0`
			DISPLAY TEXT Proactive command
			`qualifier = 00h`
			`Text String TLV = 8D 00`

17	**Select a TLV in the ProactiveHandler** **Call the initDisplayText() 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`
			DISPLAY TEXT Proactive 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 Proactive 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 Proactive 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**		No proactive command shall be sent expected status is '9000'

6.2.7.3.4 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
N6	16
P1	1
P2	2, 3, 4, 5, 6
C1	21

6.2.7.4 Method initGetInkey

Test Area Reference: API_2_PAH_INGKBB_BSS

6.2.7.4.1 Conformance requirement

The method with following header 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
```

6.2.7.4.1.1 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.
- CRRN6: If length is equal to zero, then the Text String TLV inserted in the command is a null text string TLV as defined in TS 11.14 [4].

6.2.7.4.1.2 Parameter errors

- CRRP1: The method shall throw NullPointerException if buffer is null.
- CRRP1: If offset or length or both would cause access outside array bounds, a ArrayIndexOutOfBoundsException shall be thrown.

6.2.7.4.1.3 Context errors

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

6.2.7.4.2 Test Suite files

Test Script:	API_2_PAH_INGKBB_BSS_1.scr
Test Applet:	API_2_PAH_INGKBB_BSS_1.java

Load Script: API_2_PAH_INGKBB_BSS_1.ldr

Cleanup Script: API_2_PAH_INGKBB_BSS_1.clr

Parameter File: API_2_PAH_INGKBB_BSS_1.par

6.2.7.4.3 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 = 5`
		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`
		No exception is thrown	

	Verify the command number value	Command number between 01h and FEh	

8	**Send the command**		GET INKEY Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextA"`

9	**Successful call, buffer is part of a buffer with the end part**
	`qualifier = 0`
	`dcs = 4`
	`buffer = "12TextB"`
	`offset = 2`
	`length = 5`
			GET INKEY Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextB"`

10	**Successful call, buffer is part of a buffer with the first part**
	`qualifier = 0`
	`dcs = 4`
	`buffer = "TextC12"`
	`offset = 0`
	`length = 5`
			GET INKEY Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextC"`

11	**Successful call, buffer is part of a buffer Send the command**
	`qualifier = 0`
	`dcs = 4`
	`buffer = "12TextD34"`
	`offset = 2`
	`length = 5`
			GET INKEY Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextD"`

12	**Successful call, qualifier = 81h**
	`qualifier = 81h`
	`dcs = 4`
	`buffer = "TextE"`
	`offset = 0`
	`length = 5`
			GET INKEY Proactive command
			`qualifier = 81h`
			`dcs = 4`
			`Text = "TextE"`

Id	Description	API Expectation	APDU Expectation

13	**Successful call, DCS=0 (7 bits)**
	`qualifier = 0`
	`dcs = 0`
	`buffer = "TextF"`
	`offset = 0`
	`length = 5`
			GET INKEY Proactive command
			`qualifier = 00h`
			`dcs = 0`
			`Text = "TextF"`

14	**Successful call, DCS=8 (UCS2)**
	`qualifier = 0`
	`dcs = 8`
	`buffer = "TextG"`
	`offset = 0`
	`length = 5`
			GET INKEY Proactive command
			`qualifier = 00h`
			`dcs = 8`
			`Text = "TextG"`

15	**Call the initGetInkey() method with any value Then build and send a GET INKEY command**
	`qualifier = 0`
	`dcs = 4`
	`buffer = "TextHTextH"`
	`offset = 0`
	`length = 10`
			GET INKEY Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextHTextH"`

16	**Successful call, text length is zero**
	Send the command
	`qualifier = 0`
	`dcs = 4`
	`buffer = "TextHTextH"`
	`offset = 0`
	`length = 0`
			GET INKEY Proactive 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 Proactive 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 Proactive 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`
			GET INKEY Proactive command
			`Text String TLV = 8D 81 F1 04 55 55…`

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**		No proactive command shall be sent expected status is '9000'

6.2.7.4.4 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
N6	16
P1	1
P2	2, 3, 4, 5, 6
C1	21

6.2.7.5 Method initGetInput

Test Area Reference: API_2_PAH_INGPBB_BSSSS

6.2.7.5.1 Conformance requirement

The method with following header 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
```

6.2.7.5.1.1 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.

- CRRN6: If length is equal to zero, then the Text String TLV inserted in the command is a null text string TLV as defined in TS 11.14 [4].

6.2.7.5.1.2 Parameter errors

- 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.

6.2.7.5.1.3 Context errors

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

6.2.7.5.2 Test Suite files

Test Script: API_2_PAH_INGPBB_BSSSS_1.scr

Test Applet: API_2_PAH_INGPBB_BSSSS_1.java

Load Script: API_2_PAH_INGPBB_BSSSS_1.ldr

Cleanup Script: API_2_PAH_INGPBB_BSSSS_1.clr

Parameter File: API_2_PAH_INGPBB_BSSSS_1.par

6.2.7.5.3 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 = 5`
		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`
	`minRespLength = 00h`
	`maxRespLength = FFh`
		No exception is thrown	

	Verify the command number value	Command number between 01h and FEh	

8	**Send the command**		GET INPUT Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextA"`
			`Min Length = 00h`
			`Max Length = FFh`

9	**Successful call, buffer is part of a buffer with the end part**
	Send the command
	`qualifier = 0`
	`dcs = 4`
	`buffer = "12TextB"`
	`offset = 2`
	`length = 5`
	`minRespLength = 10h`
	`maxRespLength = FFh`
			GET INPUT Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextB"`
			`Min Length = 10h`
			`Max Length = FFh`

10	**Successful call, buffer is part of a buffer with the first part**
	Send the command
	`qualifier = 0`
	`dcs = 4`
	`buffer = "TextC12"`
	`offset = 0`
	`length = 5`
	`minRespLength = FFh`
	`maxRespLength = FFh`
			GET INPUT Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextC"`
			`Min Length = FFh`
			`Max Length = FFh`

Id	Description	API Expectation	APDU Expectation

11	**Successful call, buffer is part of a buffer**
	Send the command
	`qualifier = 0`
	`dcs = 4`
	`buffer = "12TextD34"`
	`offset = 2`
	`length = 5`
	`minRespLength = 00h`
	`maxRespLength = 00h`
			GET INPUT Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextD"`
			`Min Length = 00h`
			`Max Length = 00h`

12	**Successful call, qualifier = 81h**
	`qualifier = 81h`
	`dcs = 4`
	`buffer = "TextE"`
	`offset = 0`
	`length = 5`
	`minRespLength = 00h`
	`maxRespLength = 10h`
			GET INPUT Proactive command
			`qualifier = 81h`
			`dcs = 4`
			`Text = "TextE"`
			`Min Length = 00h`
			`Max Length = 10h`

13	**Successful call, DCS=0 (7 bits)**
	`qualifier = 0`
	`dcs = 0`
	`buffer = "TextF"`
	`offset = 0`
	`length = 5`
	`minRespLength = 10h`
	`maxRespLength = 10h`
			GET INPUT Proactive command
			`qualifier = 00h`
			`dcs = 0`
			`Text = "TextF"`
			`Min Length = 10h`
			`Max Length = 10h`

14	**Successful call, DCS=8 (UCS2)**
	`qualifier = 0`
	`dcs = 8`
	`buffer = "TextG"`
	`offset = 0`
	`length = 5`
	`minRespLength = 00h`
	`maxRespLength = FFh`
			GET INPUT Proactive command
			`qualifier = 00h`
			`dcs = 8`
			`Text = "TextG"`
			`Min Length = 00h`
			`Max Length = FFh`

15	**Call the initGetInput() method with any value**
	Then build and send a GET INPUT command
	`qualifier = 0`
	`dcs = 4`
	`buffer = "TextHTextH"`
	`offset = 0`
	`length = 10`
	`minRespLength = 00h`
	`maxRespLength = 10h`
			GET INPUT Proactive command
			`qualifier = 00h`
			`dcs = 4`
			`Text = "TextHTextH"`
			`Min Length = 00h`
			`Max Length = 10h`

16	**Successful call, text length is zero**
	Send the command
	`qualifier = 0`
	`dcs = 4`
	`buffer = "TextHTextH"`
	`offset = 0`
	`length = 0`
	`minRespLength = 00h`
	`maxRespLength = 10h`
			GET INPUT Proactive 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 Proactive command
			`Text String TLV =`
			`8D 7F 04 55 55…`
			`Min Length = 00h`
			`Max Length = 10h`

Id	Description	API Expectation	APDU Expectation

19	**Successful call, buffer length = 7Fh**
	`qualifier = 0`
	`dcs = 4`
	`buffer = "UUU…"`
	`offset = 0`
	`length = 7Fh`
	`minRespLength = 00h`
	`maxRespLength = 10h`
			GET INPUT Proactive command
			`Text String TLV = 8D 81 80 04 55 55…`
			`Min Length = 00h`
			`Max Length = 10h`

20	**Successful call, buffer length = 236**
	`Qualifier = 0`
	`dcs = 4`
	`buffer = "UUU…"`
	`offset = 0`
	`length = 236`
	`minRespLength = 00h`
	`maxRespLength = 10h`
			GET INPUT Proactive command
			`Text String TLV = 8D 81 ED 04 55 55…`

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**		No proactive command shall be sent expected status is '9000'

6.2.7.5.4 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
N6	16
P1	1
P2	2, 3, 4, 5, 6
C1	21

6.2.7.6 Method send

Test Area Reference: API_2_PAH_SEND

6.2.7.6.1 Conformance requirement

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

```
public byte send()
```

6.2.7.6.1.1 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.

6.2.7.6.1.2 Parameter errors

No requirements.

6.2.7.6.1.3 Context errors

- CRRC1: A ToolkitException.UNAVAILABLE_ELEMENT shall be thrown is the Result Simple TLV is missing in Terminal Response.
- CRRC2: A ToolkitException.OUT_OF_TLV_BOUNDARIES shall be thrown if the general result byte is missing in the Result Simple TLV in Terminal Response.
- CRRC3: A ToolkitException COMMAND_NOT_ALLOWED shall be thrown if the proactive command to be sent is not allowed by the SIM Toolkit Framework.
- CRRC4: A ToolkitException COMMAND_NOT_ALLOWED shall be thrown if one parameter of the proactive command to be sent is not allowed by the SIM Toolkit Framework.

6.2.7.6.2 Test Suite files

Test Script:	API_2_PAH_SEND_1.scr
Test Applet:	API_2_PAH_SEND_1.java
Load Script:	API_2_PAH_SEND_1.ldr
Cleanup Script:	API_2_PAH_SEND_1.clr
Parameter File:	API_2_PAH_SEND_1.par

6.2.7.6.3 Test procedure

Id	Description	API Expectation	APDU Expectation

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

8	**Terminal Response with General Result = 02** `Result TLV = 03 04 02 65 43 21 (Missing`	Result of send() is 02h	

Id	Description	API Expectation	APDU Expectation
	information)		

9	**Build and send a 7Fh byte command (DISPLAY TEXT)**
	qualifier = 00h
	dcs = 04h
	buffer = "UUUUU…"
	length = 73h
			DISPLAY TEXT Proactive 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 Proactive 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 Proactive command
			BER-TLV = D0 81 FD
			Text String TLV = 8D 81 F1 04 55 55…

12	**Verify that the Proactive Handler is not modified after a send()** **Build a DISPLAY TEXT command** **Copy ProactiveHandler to source byte array** **Send command** **Copy ProactiveHandler to destination byte array** **Compare source and destination**	Source and destination are identical	
13	**Build and send a DISPLAY TEXT command Verify there is no invocation of select() or deselect() method.**		DISPLAY TEXT Proactive command
14	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive 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	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive command
	Terminal Response without Result Simple TLV	ToolkitException.UNAVAILABLE_ ELEMENT is thrown by send()	
16	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive command
	Terminal Response without general result byte in the Simple TLV Result TLV = 03 00	ToolkitException.OUT_OF_TLV_ BOUNDARIES is thrown by send()	

6.2.7.6.4 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	checked in Framework tests: FWK_API_TRAN
C1	15

C2	16
C3	checked in the Framework test : FWK_PCS_PCCO (test case 1)
C4	checked in the Framework test : FWK_PCS_PCCO (test cases 2 to 3)

6.2.7.7 Method getLength

Test Area Reference API_2_PAH_GLEN

6.2.7.7.1 Conformance requirement

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

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

6.2.7.7.1.1 Normal execution

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

6.2.7.7.1.2 Parameter errors

No requirements.

6.2.7.7.1.3 Context errors

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

6.2.7.7.2 Test Suite files

Test Script:	API_2_PAH_GLEN_1.scr
Test Applet:	API_2_PAH_GLEN_1.java
Load Script:	API_2_PAH_GLEN_1.ldr
Cleanup Script:	API_2_PAH_GLEN_1.clr
Parameter File:	API_2_PAH_GLEN_1.par

6.2.7.7.3 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	

6.2.7.7.4 Test Coverage

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

6.2.7.8 Method copy

Test Area Reference API_2_PAH_COPY_BSS

6.2.7.8.1 Conformance requirement

The method with following header 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
```

6.2.7.8.1.1 Normal execution

- CRRN1: copies the simple TLV list contained in the handler to the destination byte array.
- CRRN2: returns dstOffset + dstLength.

6.2.7.8.1.2 Parameter errors

- 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.

6.2.7.8.1.3 Context errors

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

6.2.7.8.2 Test Suite files

Test Script: API_2_PAH_COPY_BSS_1.scr

Test Applet: API_2_PAH_COPY_BSS_1.java

Load Script: API_2_PAH_COPY_BSS_1.ldr

Cleanup Script: API_2_PAH_COPY_BSS_1.clr

Parameter File: API_2_PAH_COPY_BSS_1.par

6.2.7.8.3 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 = 6`
	`dstLength = 0`
		ArrayIndexOutOfBoundsException is thrown	

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

Id	Description	API Expectation	APDU Expectation

	`dstOffset = -1`
	`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	**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	

6.2.7.8.4 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 ProactiveHandler

6.2.7.9 Method findTLV

Test Area Reference API_2_PAH_FINDBB

6.2.7.9.1 Conformance requirement

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

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

6.2.7.9.1.1 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, the current TLV is no longer defined and TLV_NOT_FOUND is returned.
- CRRN5: The search method is comprehension required flag independent.

6.2.7.9.1.2 Parameter errors

- 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.

6.2.7.9.1.3 Context errors

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

6.2.7.9.2 Test Suite files

Test Script:	API_2_PAH_FINDBB_1.scr
Test Applet:	API_2_PAH_FINDBB_1.java
Load Script:	API_2_PAH_FINDBB_1.ldr
Cleanup Script:	API_2_PAH_FINDBB_1.clr
Parameter File:	API_2_PAH_FINDBB_1.par

6.2.7.9.3 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	**Select a TLV (tag 02h)**		

	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`	Result is TLV_FOUND_CR_SET	

	Occurrence = 1		

13	**Search tag 84h**
	Tag = 84h
	Occurrence = 1
		Result is TLV_FOUND_CR_NOT_SET	

6.2.7.9.4 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

6.2.7.10 Method getValueLength

Test Area Reference API_2_PAH_GVLE

6.2.7.10.1 Conformance requirement

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

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

6.2.7.10.1.1 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.

6.2.7.10.1.2 Parameter errors

No requirements.

6.2.7.10.1.3 Context errors

- 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.

6.2.7.10.2 Test Suite files

Test Script:	API_2_PAH_GVLE_1.scr
Test Applet:	API_2_PAH_GVLE_1.java
Load Script:	API_2_PAH_GVLE_1.ldr
Cleanup Script:	API_2_PAH_GVLE_1.clr
Parameter File:	API_2_PAH_GVLE_1.par

6.2.7.10.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the init() method**		
	getValueLength()	ToolkitException.UNAVAILABLE_ ELEMENT is thrown	

Id	Description	API Expectation	APDU Expectation

2	**Call the appendTLV() method**
	`tag = 0D`
	`valueOffset = 0`
	`valueLength = 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	

6.2.7.10.4 Test Coverage

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

6.2.7.11 Method getValueByte

Test Area Reference API_2_PAH_GVBYS

6.2.7.11.1 Conformance requirement

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

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

6.2.7.11.1.1 Normal execution

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

6.2.7.11.1.2 Parameter errors

- 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.

6.2.7.11.1.3 Context errors

- 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.

6.2.7.11.2 Test Suite files

Test Script:	API_2_PAH_GVBYS_1.scr
Test Applet:	API_2_PAH_GVBYS_1.java
Load Script:	API_2_PAH_GVBYS_1.ldr
Cleanup Script:	API_2_PAH_GVBYS_1.clr
Parameter File:	API_2_PAH_GVBYS_1.par

6.2.7.11.3 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	

6.2.7.11.4 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

6.2.7.12 Method copyValue

Test Area Reference API_2_PAH_CPYVS_BSS

6.2.7.12.1 Conformance requirement

The method with following header 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
```

6.2.7.12.1.1 Normal execution

- CRRN1: copies a part of the last TLV element which has been found, into a destination. buffer.
- CRRN2: returns dstOffset + dstLength.

6.2.7.12.1.2 Parameter errors

- 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 is negative or valueOffset + dstLength > current TLV length, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

6.2.7.12.1.3 Context errors

- 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.

6.2.7.12.2 Test Suite files

Test Script:	API_2_PAH_CPYVS_BSS_1.scr
Test Applet:	API_2_PAH_CPYVS_BSS_1.java
Load Script:	API_2_PAH_CPYVS_BSS_1.ldr
Cleanup Script:	API_2_PAH_CPYVS_BSS_1.clr
Parameter File:	API_2_PAH_CPYVS_BSS_1.par

6.2.7.12.3 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 = 6`
	`dstLength = 0`
		ArrayIndexOutOfBoundsException is thrown	

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

Id	Description	API Expectation	APDU Expectation

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 = 7`
	`dstBuffer.length = 15`
	`dstOffset = 0`
	`dstLength = 0`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

8	**[Select Text String TLV]**
	`valueOffset < 0`
	`valueOffset = -1`
	`dstBuffer.length = 15`
	`dstOffset = 0`
	`dstLength = 1`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

9	**[Select Text String TLV]**
	`dstLength > Text String length`
	`valueOffset = 0`
	`dstBuffer.length = 15`
	`dstOffset = 0`
	`dstLength = 7`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

10	**[Select Text String TLV]**
	`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	

6.2.7.12.4 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

6.2.7.13 Method compareValue

Test Area Reference API_2_PAH_CPRVS_BSS

6.2.7.13.1 Conformance requirement

The method with following header 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
```

6.2.7.13.1.1 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.

6.2.7.13.1.2 Parameter errors

- 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 is negative or valueOffset + dstLength > current TLV length, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

6.2.7.13.1.3 Context errors

- 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.

6.2.7.13.2 Test Suite files

Test Script:	API_2_PAH_CPRVS_BSS_1.scr
Test Applet:	API_2_PAH_CPRVS_BSS_1.java
Load Script:	API_2_PAH_CPRVS_BSS_1.ldr
Cleanup Script:	API_2_PAH_CPRVS_BSS_1.clr

Parameter File: API_2_PAH_CPRVS_BSS_1.par

6.2.7.13.3 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 = 6`
	`compareLength = 0`
		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** **Select Text String TLV**		

	valueOffset > Text String Length
	`valueOffset = 7`
	`compareBuffer.length = 15`
	`compareOffset = 0`
	`compareLength = 0`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

8	**[Select Text String TLV]**
	`valueOffset < 0`
	`valueOffset = -1`
	`compareBuffer.length = 15`
	`compareOffset = 0`
	`compareLength = 1`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

9	**[Select Text String TLV]**
	`compareLength > Text String length`
	`valueOffset = 0`
	`compareBuffer.length = 15`
	`compareOffset = 0`
	`compareLength = 7`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

10	**[Select Text String TLV]**
	`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	

Id	Description	API Expectation	APDU Expectation

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	

18	**Initialise compareBuffer**
	`compareBuffer =`
	`55 55 55 99 03`
	`03 04 05 06 07`
	`08 09 0A 0B 0C`
	`55 55 55 55 55`		

	Compare buffers with same parameters	Result is +1	

6.2.7.13.4 Test Coverage

CRR number	Test case number
N1	12, 15
N2	13, 16
N3	14, 17, 18
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for Proactive Handler
C2	11

6.2.7.14 Method findAndCopyValue(byte tag, byte[] dstBuffer, short valueOffset)

Test Area Reference API_2_PAH_FACYB_BS

6.2.7.14.1 Conformance requirement

The method with following header 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
```

6.2.7.14.1.1 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.

6.2.7.14.1.2 Parameter errors

- CRRP1: if dstBuffer is null NullPointerException shall be thrown.

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

6.2.7.14.1.3 Context errors

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

6.2.7.14.2 Test Suite files

Test Script:	API_2_PAH_FACYB_BS_1.scr
Test Applet:	API_2_PAH_FACYB_BS_1.java
Load Script:	API_2_PAH_FACYB_BS_1.ldr
Cleanup Script:	API_2_PAH_FACYB_BS_1.clr
Parameter File:	API_2_PAH_FACYB_BS_1.par

6.2.7.14.3 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 = 21`
		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 findAndcopyValue() 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 findAndcopyValue() 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 findAndcopyValue() 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 findAndcopyValue() is 17	

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

Id	Description	API Expectation	APDU Expectation
15	**Append tag 0Fh** `buffer = 00 01 … 0F`		

	Successful call (with tag 8Fh)
	`tag = 8Fh`
	`dstBuffer.length = 16`
	`dstOffset = 0`
		Result of findAndcopyValue() is 16	

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

6.2.7.14.4 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

6.2.7.15 Method findAndCopyValue(byte tag, byte occurence, short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)

Test Area Reference API_2_PAH_FACYBBS_BSS

6.2.7.15.1 Conformance requirement

The method with following header 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
```

6.2.7.15.1.1 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.

6.2.7.15.1.2 Parameter errors

- 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 is negative or valueOffset + dstLength > current TLV length, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

6.2.7.15.1.3 Context errors

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

6.2.7.15.2 Test Suite files

Test Script:	API_2_PAH_FACYBBS_BSS_1.scr
Test Applet:	API_2_PAH_FACYBBS_BSS_1.java
Load Script:	API_2_PAH_FACYBBS_BSS_1.ldr
Cleanup Script:	API_2_PAH_FACYBBS_BSS_1.clr
Parameter File:	API_2_PAH_FACYBBS_BSS_1.par

6.2.7.15.3 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 = 6`
	`dstLength = 0`
		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 = 7`
	`dstBuffer.length = 15`
	`dstOffset = 0`
	`dstLength = 0`
		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	

Id	Description	API Expectation	APDU Expectation
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 findAndcopyValue() 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`
	`occurrence = 1`
	`valueOffset = 0`
	`dstBuffer.length = 17`
	`dstOffset = 0`
	`dstLength = 17`
		Result of findAndcopyValue() is 17	

21	**Compare buffer** `buffer = 04 00 01 … 0F`	Result is 00h	

Id	Description	API Expectation	APDU Expectation
22	**Append tag 0Fh** `buffer = 00 01 … 0F`		

	Successful call (with tag 8Fh)
	`tag = 8Fh`
	`occurrence = 1`
	`valueOffset = 0`
	`dstBuffer.length = 16`
	`dstOffset = 0`
	`dstLength = 16`
		Result of findAndcopyValue() is 16	

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

6.2.7.15.4 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 ProactiveHandler

6.2.7.16 Method findAndCompareValue(byte tag, byte[] compareBuffer, short compareOffset)

Test Area Reference API_2_PAH_FACRB_BS

6.2.7.16.1 Conformance requirement

The method with following header 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
```

6.2.7.16.1.1 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.

6.2.7.16.1.2 Parameter errors

- CRRP1: if compareBuffer is null NullPointerException shall be thrown.

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

6.2.7.16.1.3 Context errors

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

6.2.7.16.2 Test Suite files

Test Script:	API_2_PAH_FACRB_BS_1.scr
Test Applet:	API_2_PAH_FACRB_BS_1.java
Load Script:	API_2_PAH_FACRB_BS_1.ldr
Cleanup Script:	API_2_PAH_FACRB_BS_1.clr
Parameter File:	API_2_PAH_FACRB_BS_1.par

6.2.7.16.3 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 = 21`
		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	

Id	Description	API Expectation	APDU Expectation

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	

17	**Initialise compareBuffer** `compareBuffer = 00 99 01 03 … 0F`		

	Successful call (with tag 8Fh)
	`tag = 8Fh`
	`compareBuffer.length = 16`
	`compareOffset = 0`
		Result is +1	

6.2.7.16.4 Test Coverage

CRR number	Test case number
N1	6
N2	8
N3	7, 11, 12, 17
N4	9, 13

CRR number	Test case number
N5	10, 14
N6	15, 16
P1	1
P2	2, 3, 4, 5
C1	Does not apply for Proactive Handler

6.2.7.17 Method findAndCompareValue(byte tag, byte occurence, short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)

Test Area Reference API_2_PAH_FACRBBS_BSS

6.2.7.17.1 Conformance requirement

The method with following header 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
```

6.2.7.17.1.1 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.

6.2.7.17.1.2 Parameter errors

- 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 is negative or valueOffset + dstLength > current TLV length, 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. occurrence = 0) an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException BAD_INPUT_PARAMETER.

6.2.7.17.1.3 Context errors

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

6.2.7.17.2 Test Suite files

Test Script:	API_2_PAH_FACRBBS_BSS_1.scr
Test Applet:	API_2_PAH_FACRBBS_BSS_1.java
Load Script:	API_2_PAH_FACRBBS_BSS_1.ldr
Cleanup Script:	API_2_PAH_FACRBBS_BSS_1.clr
Parameter File:	API_2_PAH_FACRBBS_BSS_1.par

6.2.7.17.3 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 = 6`
	`compareLength = 0`
		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 = 7`
	`compareBuffer.length = 15`
	`compareOffset = 0`
	`compareLength = 0`
		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	

Id	Description	API Expectation	APDU Expectation
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	

Id	Description	API Expectation	APDU Expectation

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, occurrence = 1`
	`valueOffset = 0`
	`compareBuffer.length = 17`
	`compareOffset = 0`
	`compareLength = 17`
		Result is 00h	

24	**Append tag 0Fh** `buffer = 00 01 … 0F`		
	Initialise compareBuffer `compareBuffer = 00 01 … 0F`		

	Successful call (with tag 8Fh)
	`tag = 8Fh, occurrence = 1`
	`valueOffset = 0`
	`compareBuffer.length = 16`
	`compareOffset = 0`
	`compareLength = 16`
		Result is 00h	

25	**Initialise compareBuffer** `compareBuffer =0099 02 … 0F`		

	findAndCompareValue()
	`tag = 0Dh, occurrence = 1`
	`valueOffset = 0`
	`compareOffset = 0`
	`compareLength = 17`
		Result is +1	

6.2.7.17.4 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

6.2.7.18 Method appendArray

Test Area Reference: API_2_PAH_APDA_BSS

6.2.7.18.1 Conformance requirement:

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

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

6.2.7.18.1.1 Normal execution

- CRRN1: appends a buffer into the Edithandler buffer.
- CRRN2: a successful append does not modify the TLV selected.

6.2.7.18.1.2 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.

6.2.7.18.1.3 Context errors

- 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.

6.2.7.18.2 Test suite files

Test Script:	API_2_PAH_APDA_BSS_1.scr
Test Applet:	API_2_PAH_APDA_BSS_1.java
Load Script:	API_2_PAH_APDA_BSS_1.ldr
Cleanup Script:	API_2_PAH_APDA_BSS_1.clr
Parameter File:	API_2_PAH_APDA_BSS_1.par

6.2.7.18.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Null buffer**	NullPointerException is thrown	

2	**offset > buffer.length**
	buffer.length = 5
	offset = 6
	length = 0
		ArrayIndexOutOfBoundsException is thrown	

3	**offset < 0**
	buffer.length = 5
	offset = -1
	length = 1
		ArrayIndexOutOfBoundsException is thrown	

4	**length > buffer.length**
	buffer.length = 5
	offset = 0
	length = 6
		ArrayIndexOutOfBoundsException is thrown	

5	**offset + length > buffer.length**
	buffer.length = 5
	offset = 3
	length = 3
		ArrayIndexOutOfBoundsException is thrown	

6	**length < 0**
	buffer.length = 5
	offset = 0
	length = -1
		ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation

7	**Handler overflow**
	`buffer.length = 256`
	`offset = 0`
	`length = 256`
		ToolkitException.HANDLER_OVERFLOW is thrown	

8	**Initialise handler**		
	Select Command Details TLV		

	Successful call
	`buffer = FF FE … F8`
	`offset = 0`
	`length = 8`		

	Verify Current TLV: Call getValueLength()	Result is 03h	
9	**Clear the handler**		

	Successful call
	`buffer = FF FE … F8`
	`offset = 0`
	`length = 8`		

	Call copy() method		
	Compare the arrays `compareBuffer = FF FE … F8`	Result of javacard.framework.Util.arrayCompare() is 00h	

10	**Successful call**
	`buffer = 00 01 … 07`
	`offset = 2`
	`length = 6`		

	Call copy() method		
	Compare the arrays `compareBuffer = FF FE … F8 02 03 … 07`	Result of javacard.framework.Util.arrayCompare() is 00h	

11	**Successful call**
	`buffer = 11 22 … 88`
	`offset = 2`
	`length = 4`		

	Call copy() method		
	Compare the arrays `compareBuffer = FF FE … F8 02 03 … 07 33 44 55 66`	Result of javacard.framework.Util.arrayCompare() is 00h	
12	**Clear the handler**		

	Successful call
	`buffer = 00 01 … FC`
	`offset = 0`
	`length = 253`		

	Call getLength() method	result = 253	
	Call copy() method		
	Compare handler `compareBuffer = 00 01 … FC`	Result of javacard.framework.Util.arrayCompare() is 00h	

6.2.7.18.4 Test Coverage

CRR number	Test case number
N1	9, 10, 11, 12
N2	8
P1	1
P2	2, 3, 4, 5, 6
C1	7
C2	Does not apply for ProactiveHandler

6.2.7.19 Method appendTLV(byte tag, byte value)

Test Area Reference: API_2_PAH_APTLBB

6.2.7.19.1 Conformance requirement:

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

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

6.2.7.19.1.1 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.

6.2.7.19.1.2 Parameters error

No requirements

6.2.7.19.1.3 Context errors

- 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.

6.2.7.19.2 Test suite files

Test Script:	API_2_PAH_APTLBB_1.scr
Test Applet:	API_2_PAH_APTLBB_1.java
Load Script:	API_2_PAH_APTLBB_1.ldr
Cleanup Script:	API_2_PAH_APTLBB_1.clr
Parameter File:	API_2_PAH_APTLBB_1.par

6.2.7.19.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call appendArray()** `length = 251`		
	Handler Overflow: Call the appendTLV() method	ToolkitException.HANDLER_OVERFLOW is thrown	
2	**Initialise handler**		
	Select Command Details TLV		
	Call the appendTLV() method		
	Verify Current TLV: Call getValueLength()	Result is 03h	
3	**Clear the handler**		

	Successful call
	`tag = 84h`
	`value = 00h`		

	Call copy() method		
	Compare the arrays `compareBuffer = 84 01 00`	Result of javacard.framework.Util.arrayCompare() is 00h	

4	**Successful call**
	`tag = 01h`
	`value = FEh`		

	Call copy() method		
	Compare the arrays `compareBuffer = 84 01 00 01 01 FE`	Result of javacard.framework.Util.arrayCompare() is 00h	

Id	Description	API Expectation	APDU Expectation

5	**Clear the handler**
	Call appendArray()
	`length = 250`
	`buffer = 00 81 F7 03 04 … F9`
	Successful call
	`tag = 84h`
	`value = 00h`
	Call getLength() method
	Call copy() method
	Compare the array
	`compareBuffer = 00 81 F7 03 04 … F9 84 01 00`
		result = 253
		Result of javacard.framework.Util.arrayCompare() is 00h	

6.2.7.19.4 Test Coverage

CRR number	Test case number
N1	3, 4, 5
N2	2
C1	1
C2	Does not apply for Proactive Handler

6.2.7.20 Method appendTLV(byte tag, byte value1, byte value2)

Test Area Reference: API_2_PAH_APTLBBB

6.2.7.20.1 Conformance requirements:

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

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

6.2.7.20.1.1 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.

6.2.7.20.1.2 Parameters error

No requirements

6.2.7.20.1.3 Context errors

- 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.

6.2.7.20.2 Test suite files

Test Script:	API_2_PAH_APTLBBB_1.scr
Test Applet:	API_2_PAH_APTLBBB_1.java
Load Script:	API_2_PAH_APTLBBB_1.ldr

Cleanup Script: API_2_PAH_APTLBBB_1.clr

Parameter File: API_2_PAH_APTLBBB_1.par

6.2.7.20.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the initDisplayText()** `length = 250`		
	Handler Overflow: Call the appendTLV() method	ToolkitException.HANDLER_OVERFLOW is thrown	
2	**Initialise handler**		
	Select Command Details TLV		
	Call the appendTLV() method		
	Verify Current TLV: Call getValueLength()	Result is 03h	
3	**Clear the handler**		

	Successful call
	`tag = 84h`
	`value1 = 00h`
	`value2 = 01h`		

	Call copy() method		
	Compare the arrays `compareBuffer = 84 02 00 01`	Result of javacard.framework.Util.arrayCompare() is 00h	

4	**Successful call**
	`tag = 01h`
	`value1 = FEh`
	`value2 = FDh`		

	Call copy() method		
	Compare the arrays `compareBuffer = 84 02 00 01 01 02 FE FD`	Result of javacard.framework.Util.arrayCompare() is 00h	
5	**Clear the handler**		

	Call appendArray()
	`length = 249`
	`buffer = 00 81 F6 03 04 … F8`		

	Successful call
	`tag = 84h`
	`value1 = 00h`
	`value2 = 01h`		

	Call getLength() method	result = 253	
	Call copy() method		
	Compare handler `compareBuffer = 00 81 F6 03 04 … F8 84 02 00 01`	Result of javacard.framework.Util.arrayCompare() is 00h	

6.2.7.20.4 Test Coverage

CRR number	Test case number
N1	3, 4, 5
N2	2
C1	1
C2	Does not apply for Proactive Handler

6.2.7.21 Method appendTLV(byte tag, byte[] value, short valueoffset, short valuelength)

Test Area Reference: API_2_PAH_APTLB_BSS

6.2.7.21.1 Conformance requirement

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

```
void appendTLV(byte tag,
               byte[] value,
               short valueoffset,
               short valuelength)
        throws java.lang.NullPointerException,
               java.lang.ArrayIndexOutOfBoundsException,
               ToolkitException
```

6.2.7.21.1.1 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.

6.2.7.21.1.2 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.

6.2.7.21.1.3 Context errors

- 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.

6.2.7.21.2 Test suite files

Test Script:	API_2_PAH_APTLB_BSS_1.scr
Test Applet:	API_2_PAH_APTLB_BSS_1.java
Load Script:	API_2_PAH_APTLB_BSS_1.ldr
Cleanup Script:	API_2_PAH_APTLB_BSS_1.clr
Parameter File:	API_2_PAH_APTLB_BSS_1.par

6.2.7.21.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Null value**	NullPointerException is thrown	

2	**valueOffset > value.length**
	`value.length = 5`
	`valueOffset = 6`
	`valueLength = 0`
		ArrayIndexOutOfBoundsException is thrown	

3	**valueOffset < 0**
	`value.length = 5`
	`valueOffset = -1`
	`valueLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

4	**valueLength > value.length**
	`value.length = 5`
	`valueOffset = 0`
	`valueLength = 6`
		ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation

5	**valueOffset + valueLength > value.length**
	value.length = 5
	valueOffset = 3
	valueLength = 3
		ArrayIndexOutOfBoundsException is thrown	

6	**valueLength < 0**
	value.length = 5
	valueOffset = 0
	valueLength = -1
		ArrayIndexOutOfBoundsException is thrown	

7	**Handler overflow**
	value.length = 256
	valueOffset = 0
	valueLength = 251
		ToolkitException.HANDLER_OVERFLOW is thrown	

8	**Bad parameter**
	value.length = 256
	valueOffset = 0
	valueLength = 256
		ToolkitException.BAD_INPUT_PARAMETER is thrown	

9	**Initialise handler**		
	Select Command Details TLV		

	Successful call
	tag = 04
	value = FF FE … F8
	valueOffset = 0
	valueLength = 8		

	Verify Current TLV: Call getValueLength()	Result is 03h	
10	**Clear the handler**		

	Successful call
	tag = 04
	value = FF FE … F8
	valueOffset = 0
	valueLength = 8		

	Call copy() method		
	Compare the arrays compareBuffer = 04 08 FF FE … F8	Result of javacard.framework.Util.arrayCompare() is 00h	

11	**Successful call**
	tag = 85h
	value = 00 01 … 07
	valueOffset = 2
	valueLength = 6		

	Call copy() method		
	Compare the arrays compareBuffer = 04 08 FF FE … F8 85 06 02 03 … 07	Result of javacard.framework.Util.arrayCompare() is 00h	

12	**Successful call**
	tag = 01
	value = 11 22 … 88
	valueOffset = 2
	valueLength = 4		

	Call copy() method		
	Compare the arrays compareBuffer = 04 08 FF FE … F8 85 06 02 03 … 07 01 04 33 44 55 66	Result of javacard.framework.Util.arrayCompare() is 00h	
13	**Clear the handler**		

	Successful call
	tag = 04
	value = 00 01 … 7F
	valueOffset = 0
	valueLength = 80h		

	Call copy() method		
	Compare the arrays compareBuffer = 04 81 80 00 01…7F	Result of javacard.framework.Util.arrayCompare() is 00h	

Id	Description	API Expectation	APDU Expectation

14	Clear the handler
	Successful call
	`tag = 04`
	`value = 00 01 … F9`
	`valueOffset = 0`
	`valueLength = 250`
	Call getLength() method
	Call copy() method
	Compare handler
	`compareBuffer = 04 81 FA 00 01…F9`
		result = 253
		Result of javacard.framework.Util.arrayCompare() is 00h	

6.2.7.21.4 Test Coverage

CRR number	Test case number
N1	10, 11, 12, 13, 14
N2	9
P1	1
P2	2, 3, 4, 5, 6
C1	7
C2	Does not apply for Proactive Handler
C3	8

6.2.7.22 Method appendTLV(byte tag, byte value1, byte[] value2, short value2offset, short value2length)

Test Area Reference: API_2_PAH_APTLBB_BSS

6.2.7.22.1 Conformance requirement:

The method with following header shall be compliant 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
```

6.2.7.22.1.1 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.

6.2.7.22.1.2 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.

6.2.7.22.1.3 Context errors

- 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.

6.2.7.22.2 Test suite files

Test Script:	API_2_PAH_APTLBB_BSS_1.scr
Test Applet:	API_2_PAH_APTLBB_BSS_1.java
Load Script:	API_2_PAH_APTLBB_BSS_1.ldr
Cleanup Script:	API_2_PAH_APTLBB_BSS_1.clr
Parameter File:	API_2_PAH_APTLBB_BSS_1.par

6.2.7.22.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Null value2**	NullPointerException is thrown	

2	**value2Offset > value2.length**
	`value2.length = 5`
	`value2Offset = 6`
	`value2Length = 0`
		ArrayIndexOutOfBoundsException is thrown	

3	**value2Offset < 0**
	`value2.length = 5`
	`value2Offset = -1`
	`value2Length = 1`
		ArrayIndexOutOfBoundsException is thrown	

4	**value2Length > value2.length**
	`value2.length = 5`
	`value2Offset = 0`
	`value2Length = 6`
		ArrayIndexOutOfBoundsException is thrown	

5	**value2Offset + value2Length > value2.length**
	`value2.length = 5`
	`value2Offset = 3`
	`value2Length = 3`
		ArrayIndexOutOfBoundsException is thrown	

6	**value2Length < 0**
	`value2.length = 5`
	`value2Offset = 0`
	`value2Length = -1`
		ArrayIndexOutOfBoundsException is thrown	

7	**Handler overflow**
	`value2.length = 254`
	`value2Offset = 0`
	`value2Length = 254`
		ToolkitException.HANDLER_OVERFLOW is thrown	

8	**Bad parameter**
	`value2.length = 256`
	`value2Offset = 0`
	`value2Length = 256`
		ToolkitException.BAD_INPUT_PARAMETER is thrown	

9	**Initialise handler**		
	Select Command Details TLV		

	Successful call
	`tag = 04`
	`value1 = 05`
	`value2 = FF FE … F8`
	`value2Offset = 0`
	`value2Length = 8`		

	Verify Current TLV: Call getValueLength()	Result is 03h	
10	**Clear the handler**		

	Successful call
	`tag = 04`
	`value1 = 05`
	`value2 = FF FE … F8`
	`value2Offset = 0`
	`value2Length = 8`		

	Call copy() method		
	Compare the arrays `CompareBuffer = 04 09 05 FF FE … F8`	Result of javacard.framework.Util.arrayCompare() is 00h	

Id	Description	API Expectation	APDU Expectation

11	**Successful call**
	tag = 85h
	value1 = 55h
	value2 = 00 01 … 07
	value2Offset = 2
	value2Length = 6		

	Call copy() method		
	Compare the arrays compareBuffer = 04 09 05 FF FE … F8 85 07 55 02 03 … 07	Result of javacard.framework.Util.arrayCompare() is 00h	

12	**Successful call**
	tag = 01
	value1 = 44h
	value2 = 11 22 … 88
	value2Offset = 2
	value2Length = 4		

	Call copy() method		
	Compare the arrays CompareBuffer = 04 09 05 FF FE … F8 85 07 55 02 03 … 07 01 05 44 33 44 55 66	Result of javacard.framework.Util.arrayCompare() is 00h	
13	**Clear the handler**		

	Successful call
	tag = 04
	value1 = 00
	value2 = 01 … 7F
	value2Offset = 0
	value2Length = 7Fh		

	Call copy() method		
	Compare the arrays compareBuffer = 04 81 80 00 01…7F	Result of javacard.framework.Util.arrayCompare() is 00h	
14	**Clear the handler**		

	Successful call
	tag = 04
	value1 = 00
	value2 = 01 … F9
	value2Offset = 0
	value2Length = 249		

	Call getLength() method	result = 253	
	Call copy() method		
	Compare handler compareBuffer = 04 81 FA 00 01…F9	Result of javacard.framework.Util.arrayCompare() is 00h	

6.2.7.22.4 Test Coverage

CRR number	Test case number
N1	10, 11, 12, 13, 14
N2	9
P1	1
P2	2, 3, 4, 5, 6
C1	7
C2	Does not apply for Proactive Handler
C3	8

6.2.7.23 Method clear

Test Area Reference: API_2_PAH_CLER

6.2.7.23.1 Conformance requirement

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

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

6.2.7.23.1.1 Normal execution

- CRRN1: Clears the TLV list of an EditHandler
- CRRN2: Resets the current TLV selected.

6.2.7.23.1.2 Parameters error

No requirements

6.2.7.23.1.3 Context errors

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

6.2.7.23.2 Test suite files

Test Script:	API_2_PAH_CLER_1.scr
Test Applet:	API_2_PAH_CLER_1.java
Load Script:	API_2_PAH_CLER_1.ldr
Cleanup Script:	API_2_PAH_CLER_1.clr
Parameter File:	API_2_PAH_CLER_1.par

6.2.7.23.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialise the handler** Select Command Details TLV Call the getLength() method	Result of getLength() is not null	
	Clear the handler Call the getLength() method	Result of getLength() is 0	
2	**Call the getValueLength() method**	ToolkitException.UNAVAILABLE_ELEMENT is thrown	

6.2.7.23.4 Test Coverage

CRR number	Test case number
N1	1
N2	2
C1	Does not apply for Proactive Handler

6.2.7.24 Method getCapacity

Test Area Reference: API_2_PAH_GCAP

6.2.7.24.1 Conformance Requirement

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

```
public byte getCapacity()
```

6.2.7.24.1.1 Normal execution

- CRRN1: The method shall return the maximum size of the Simple TLV list managed by the handler.

6.2.7.24.2 Test suite files

Test Script:	API_2_PAH_GCAP_1.scr
Test Applet:	API_2_PAH_GCAP_1.java
Load Script:	API_2_PAH_GCAP_1.ldr
Cleanup Script:	API_2_PAH_GCAP_1.clr
Parameter File:	API_2_PAH_GCAP_1.par

6.2.7.24.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**ProactiveHandler available** 1- Send envelope SMS-PP Formatted 2- The applet calls getTheHandler() 3- The applet calls getCapacity() on the ProactiveHandler 4- The applet fills the handler with the maximum capacity, using appendTLV() method 5- The applet calls clear() on the proactive handler 6- The applet fills the handler with the maximum capacity plus one, using appendTLV() method	1- Applet is triggered 2- No exception is thrown 3- No exception is thrown, the capacity shall not be null 4- No exception is thrown 5- No exception is thrown 6- HANDLER_OVERFLOW exception is thrown	

6.2.7.24.4 Test Coverage

CRR number	Test case number
N1	1

6.2.7.25 Method initCloseChannel

Test Area Reference: API_2_PAH_ICCHB

6.2.7.25.1 Conformance requirement

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

```
public void initCloseChannel(byte bChannelIdentifier)
```

6.2.7.25.1.1 Normal execution

- CRRN1: The method shall build a Close Channel Proactive command, using Channel Identifier. Comprehension Required flags are set.
- CRRN2: A call to this method clears the handler then initialises it with Close Channel Proactive command.
- CRRN3: After the method invocation, no TLV is selected.
- CRRN4: The Close Channel Proactive command is not sent by the method.

6.2.7.25.2 Test suite files

Test Script:	API_2_PAH_ICCHB_1.scr
Test Applet:	API_2_PAH_ICCHB_1.java

Load Script: API_2_PAH_ICCHB_1.ldr

Cleanup Script: API_2_PAH_ICCHB_1.clr

Parameter File: API_2_PAH_ICCHB_1.par

6.2.7.25.3 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Applet1 is installed with maximum number of channel = 01.		
1	**Call initCloseChannel() method** 1- Call ProactiveHandler.init() method to Open a Channel. Call the ProactiveHandler.send() method. 2- Send an EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS Envelope. 3- Call the ProactiveHandler.initCloseChannel() method with Channel Id = 01. 4- Call the ProactiveHandler.send() method. 5- Send an EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS Envelope.	2- Applet1 is triggered. 5- Applet1 is not triggered.	1- OPEN CHANNEL proactive command is fetched. TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM with Channel Id = 01. 4- CLOSE CHANNEL proactive command is fetched. TERMINAL RESPONSE of CLOSE CHANNEL is sent to the SIM.
2	**Call the initCloseChannel () method with any value then build and send a CLOSE CHANNEL command** 1- Call ProactiveHandler.init() to Open a Channel and ProactiveHandler.send() methods. 2- ProactiveHandler.initCloseChannel() with Channel Id = 2 3- ProactiveHandler.initCloseChannel() with the Channel Id = 1. 4- call the send() method. 5- Send an EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS Envelope.	5- Applet1 is not triggered.	1- OPEN CHANNEL proactive command is fetched. TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM with Channel Id = 01. 4- CLOSE CHANNEL proactive command is fetched. TERMINAL RESPONSE of CLOSE CHANNEL is sent to the SIM.
3	**Select a TLV in the ProactiveHandler Call the initCloseChannel () method** 1- Call ProactiveHandler.init() method to open a Channel and call the ProactiveHandler.send() method. Select 1st TLV of the Proactive Handler. 2- Call ProactiveHandler.initCloseChannel() method with Channel Id = 01. 3- Call the ViewHandler.getValueLength() method. 4- Call ProactiveHandler.send() method.	3- UNAVAILABLE_ELEMENT ToolkitException is thrown by getValueLength() method.	1- OPEN CHANNEL proactive command is fetched. TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM with Channel Id = 01. 4- CLOSE CHANNEL proactive command is fetched. TERMINAL RESPONSE of CLOSE CHANNEL is sent to the SIM.
4	**Call the initCloseChannel() without sending the command** 1- Call ProactiveHandler.init() method to	3- Applet1 is triggered.	1- OPEN CHANNEL proactive command is fetched.

	open a Channel and call the ProactiveHandler.send() method. 2- Call the ProactiveHandler.initCloseChannel() method with Channel Id = 01 without ProactiveHandler.send(). 3- Send an EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS Envelope.		TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM with Channel Id = 01. No proactive command shall be sent. Expected status is '9000'

6.2.7.25.4 Test Coverage

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

6.2.8 Class ProactiveResponseHandler

6.2.8.1 Method copyAdditionalInformation

Test Area Reference: API_2_PRH_CPAI_BSS

6.2.8.1.1 Conformance requirement

The method with following header 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
```

6.2.8.1.1.1 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.

6.2.8.1.1.2 Parameter errors

- 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.

6.2.8.1.1.3 Context errors

- 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.

6.2.8.1.2 Test Suite files

Test Script:	API_2_PRH_CPAI_BSS_1.scr
Test Applet:	API_2_PRH_CPAI_BSS_1.java
Load Script:	API_2_PRH_CPAI_BSS_1.ldr
Cleanup Script:	API_2_PRH_CPAI_BSS_1.clr
Parameter File:	API_2_PRH_CPAI_BSS_1.par

6.2.8.1.3 Test procedure

Id	Description	API Expectation	APDU Expectation

1	**Build and send a DISPLAY TEXT command**
	`qualifier = 0`
	`dcs = 4`
	`buffer = "Text"`
			DISPLAY TEXT Proactive 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 = 11`
	`dstLength = 0`
		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 Proactive command
	Terminal Response with 5 additional bytes `Result TLV = 03 06 01 01 23 45 67 89`		

	Successful 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 Proactive command
	Terminal Response with 6 additional bytes		

Id	Description	API Expectation	APDU Expectation
	Result TLV = 03 07 01 AB CD EF FE DC BA		

	Successful 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 Proactive command
	Terminal Response with 7 additional bytes Result TLV = 03 08 01 FE DC BA 98 76 54 32		

	Successful 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 Proactive command
	Terminal Response with 8 additional bytes Result TLV = 03 09 01 00 11 22 33 44 55 66 77		

	Successful 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 Proactive command
	Terminal Response with F2h additional bytes Result TLV = 03 81 F3 01 00 01 02 03…		

	Successful 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 Proactive command
	Terminal Response with 5 additional bytes Result TLV = 03 06 01 00 11 22 33 44		
	dstLength > data available dstBuffer.length = 6	OUT_OF_TLV_BOUNDARIES	

Id	Description	API Expectation	APDU Expectation

	`dstOffset = 0`
	`dstLength = 6`
		ToolkitException is thrown	

20	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive 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 Proactive 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`		

	Successful 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.	
24	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive command
	Terminal Response without Result Simple TLV	ToolkitException.UNAVAILABLE_ELEMENT is thrown by send()	
	ProactiveResponseHandler, getTheHandler call copyAdditionalInformation()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	

6.2.8.1.4 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	24
C2	19

6.2.8.2 Method copyTextString

Test Area Reference: API_2_PRH_CPTS_BS

6.2.8.2.1 Conformance requirement

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

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

6.2.8.2.1.1 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).

6.2.8.2.1.2 Parameter errors

- CRRP1: A NullPointerException shall be thrown if dstBuffer is null.
- CRRP2: A ArrayIndexOutOfBoundsException shall be thrown if dstOffset or dstOffset + (length of the TextString to be copied, without the Data Coding Scheme included), as specified for the returned value, would cause access outside array bounds.

6.2.8.2.1.3 Context errors

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

6.2.8.2.2 Test Suite files

Test Script:	API_2_PRH_CPTS_BS_1.scr
Test Applet:	API_2_PRH_CPTS_BS_1.java
Load Script:	API_2_PRH_CPTS_BS_1.ldr
Cleanup Script:	API_2_PRH_CPTS_BS_1.clr
Parameter File:	API_2_PRH_CPTS_BS_1.par

6.2.8.2.3 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 Proactive command

	Terminal Response `Text String TLV = 0D 02 04 41`		

	ProactiveResponseHandler.getTheHandler() ; call the copyTextString() method with a null dstBuffer
	`dstBuffer = null`
	`dstOffset = 0`
		NullPointerException is thrown	

Id	Description	API Expectation	APDU Expectation
2	**Build and send a GET INPUT command**		GET INPUT Proactive command Proactive
	Terminal Response `Text String TLV = 0D 04 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 Proactive command

	Terminal Response without Text String TLV		
	ProactiveResponseHandler.getTheHandler() ; `call the copyTextString() method`	UNAVAILABLE_ELEMENT ToolkitException is thrown	
5	**Build and send a GET INPUT command**		GET INPUT Proactive command Proactive
	Terminal Response with a null Text String TLV `Text String TLV = 0D 00`		
	Initialise dstBuffer `dstBuffer = {F00h, F01h, F02h, F03h}`		

	Call the copyTextString() method
	`dstBuffer.length = 04h`
	`dstOffset = 02h`
		Result of copyTextString() is 02h	

6	**Compare dstBuffer using arrayCompare()**
	`src = {0F0h, 0F1h, 0F2h, 0F3h}`
	`srcOffset = 00h`
	`dest = dstBuffer`
	`destOffset = 00h`
	`length = 04h`
		Result of arrayCompare() is 00h	

7	**Build and send a GET INPUT command**		GET INPUT Proactive command Proactive
	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}`
	`srcOffset = 00h`
	`dest = dstBuffer`
	`destOffset = 00h`
	`length = 04h`
		Result of arrayCompare() is 00h	

9	**Build and send a GET INPUT command**		GET INPUT Proactive command Proactive
	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`	Result of copyTextString() is 04h	

Id	Description	API Expectation	APDU Expectation
	`dstOffset = 02h`		

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 Proactive 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 Proactive 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 Proactive 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 Proactive command
	Terminal Response with two Text String TLV 1st `Text String TLV = 0D 03 04 42 43`		

Id	Description	API Expectation	APDU Expectation
	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	

6.2.8.2.4 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

6.2.8.3 Method getAdditionalInformationLength

Test Area Reference: API_2_PRH_GTIL

6.2.8.3.1 Conformance requirement

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

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

6.2.8.3.1.1 Normal execution

- CRRN1: This method returns the length of the additional information field from the first Result TLV in the ProactiveResponseHandler.
- CRRN2: After a successful execution of the method, the Result TLV becomes the selected TLV of the ProactiveResponseHandler.

6.2.8.3.1.2 Parameter errors

No requirements.

6.2.8.3.1.3 Context errors

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

6.2.8.3.2 Test Suite files

Test Script:	API_2_PRH_GTIL_1.scr
Test Applet:	API_2_PRH_GTIL_1.java
Load Script:	API_2_PRH_GTIL_1.ldr

Cleanup Script: API_2_PRH_GTIL_1.clr

Parameter File: API_2_PRH_GTIL_1.par

6.2.8.3.3 Test procedure

Id	Description	API Expectation	APDU Expectation

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

	Terminal Response without additional information		
	ProactiveResponseHandler.getTheHandler() **; 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 ProactiveProactive command
	Terminal Response with 1 additional byte `Result TLV = 03 02 02 55`		
	ProactiveResponseHandler.getTheHandler() ; 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 ProactiveProactive command
	Terminal Response with 7Eh additional bytes `Result TLV = 03 7F 02 55 55 55 …`		
	ProactiveResponseHandler.getTheHandler() ; 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 Proactive command
	Terminal Response with 7Fh additional `bytes` `Result TLV = 03 81 80 02 55 55 55 …`		
	ProactiveResponseHandler.getTheHandler() ; 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 Proactive command
	Terminal Response with 80h additional bytes **Result TLV = 03 81 81 02 55 55 55 …**		
	ProactiveResponseHandler.getTheHandler() ; 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 Proactive command
	Terminal Response with F2h additional bytes `Result TLV = 03 81 F3 02 55 55 55 …`		
	ProactiveResponseHandler.getTheHandler() ; call the getAdditionalInformationLength() method	Result is F2h	

Id	Description	API Expectation	APDU Expectation
12	**Call the getValueLength() method**	Result is F3h	
13	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive command
	Terminal Response with 2 Result TLV 1st `Result TLV = 03 03 02 01 23` 2nd `Result TLV = 03 01 00`		
	ProactiveResponseHandler.getTheHandler() ; call the getAdditionalInformationLength() method	Result is 02h	
14	**Call the getValueLength() method**	Result is 03h	
15	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive command
	Terminal Response without Result Simple TLV	ToolkitException.UNAVAILABLE_ELEMENT is thrown by send()	
	Get ProactiveResponseHandler		
	Call the getAdditionalInformationLength() method	ToolkitException.UNAVAILABLE_ELEMENT is thrown by getAdditionalInformationLength ()	

6.2.8.3.4 Test Coverage

CRR number	Test case number
N1	1, 3, 5, 7, 9, 11, 13
N2	2, 4, 6, 8, 10, 12, 14
C1	15

6.2.8.4 Method getGeneralResult

Test Area Reference: API_2_PRH_GTGR

6.2.8.4.1 Conformance requirement

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

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

6.2.8.4.1.1 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.

6.2.8.4.1.2 Parameter errors

No requirements.

6.2.8.4.1.3 Context errors

- 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 the general result byte is missing in the Result Simple TLV.

6.2.8.4.2 Test Suite files

Test Script: API_2_PRH_GTGR_1.scr

Test Applet: API_2_PRH_GTGR_1.java

Load Script: API_2_PRH_GTGR_1.ldr

Cleanup Script: API_2_PRH_GTGR_1.clr

Parameter File: API_2_PRH_GTGR_1.par

6.2.8.4.3 Test procedure

Id	Description	API Expectation	APDU Expectation

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

	Terminal Response with General Result = 00 (command performed successfully)		
	ProactiveResponseHandler.getTheHandler() **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 Proactive command
	Terminal Response with General Result = 01, without Additional information on result (command performed with partial comprehension)		
	ProactiveResponseHandler.getTheHandler() **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 Proactive command
	Terminal Response with General Result = 01, with Additional information on result `Result TLV = 03 02 01 55 (command performed with partial comprehension)`		
	ProactiveResponseHandler.getTheHandler() 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 Proactive command
	Terminal Response with General Result = `02` `Result TLV = 03 04 02 65 43 21 (Missing information)`		
	ProactiveResponseHandler.getTheHandler() 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 Proactive command
	Terminal Response with 7Fh additional `bytes` `Result TLV = 03 81 80 02 55 55 55 …`		
	ProactiveResponseHandler.getTheHandler() ; call the getGeneralResult() method	Result is 02h	

Id	Description	API Expectation	APDU Expectation
10	**Call the getValueLength() method**	Result is 80h	
11	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive command
	Terminal Response with 2 Result TLV 1st Result TLV = 03 02 02 12 2nd Result TLV = 03 03 03 34 56		
	ProactiveResponseHandler.getTheHandler() ; call the getGeneralResult() method	Result is 02h	
12	**Call the getValueLength() method**	Result is 02h	
13	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive command
	Terminal Response without Result Simple TLV		
	ProactiveResponseHandler.getTheHandler() ; call the getGeneralResult() method	UNAVAILABLE_ELEMENT ToolkitException is thrown	
14	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive command
	Terminal Response without General Result Byte in Result Simple TLV		
	ProactiveResponseHandler.getTheHandler() ; call the getGeneralResult() method Result TLV = 03 00	OUT_OF_TLV_BOUNDARIES ToolkitException is thrown	

6.2.8.4.4 Test Coverage

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

6.2.8.5 Method getItemIdentifier

Test Area Reference: API_2_PRH_GTII

6.2.8.5.1 Conformance requirement

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

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

6.2.8.5.1.1 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.

6.2.8.5.1.2 Parameter errors

No requirements.

6.2.8.5.1.3 Context errors

- CRRC1: A ToolkitException.UNAVAILABLE_ELEMENT shall be thrown in case of unavailable Item Identifier TLV element.
- CRRC2: A ToolkitException.OUT_OF_TLV_BOUNDARIES shall be thrown if the item identifier byte is missing in the Item Identifier Simple TLV.

6.2.8.5.2 Test Suite files

Test Script:	API_2_PRH_GTII_1.scr
Test Applet:	API_2_PRH_GTII_1.java
Load Script:	API_2_PRH_GTII_1.ldr
Cleanup Script:	API_2_PRH_GTII_1.clr
Parameter File:	API_2_PRH_GTII_1.par

6.2.8.5.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive 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 Proactive 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 Proactive 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 Proactive 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 Proactive 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	

Id	Description	API Expectation	APDU Expectation
9	**Call the getValueByte() method** `valueOffset = 00h`	Result is FFh	
10	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive command
	Terminal Response without item identifier in the Item Identifier Simple TLV `Item Identifier TLV = 10 00`		
	Call to getItemIdentifier()	OUT_OF_TLV_BOUNDARIES ToolkitException is thrown	

6.2.8.5.4 Test Coverage

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

6.2.8.6 Method getTextStringCodingScheme

Test Area Reference: API_2_PRH_GTCS

6.2.8.6.1 Conformance requirement

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

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

6.2.8.6.1.1 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.

6.2.8.6.1.2 Parameter errors

No requirements.

6.2.8.6.1.3 Context errors

- 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.

6.2.8.6.2 Test Suite files

Test Script:	API_2_PRH_GTCS_1.scr
Test Applet:	API_2_PRH_GTCS_1.java
Load Script:	API_2_PRH_GTCS_1.ldr
Cleanup Script:	API_2_PRH_GTCS_1.clr
Parameter File:	API_2_PRH_GTCS_1.par

6.2.8.6.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive 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 Proactive command
	Terminal Response with a null Text String TLV `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 Proactive command
	Terminal Response with text length = 01h, `DCS = 04h` `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 Proactive command
	Terminal Response with text length = 02h, DCS = 00h `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 Proactive command
	Terminal Response with text length = 7Eh, DCS = 08h `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 Proactive command
	Terminal Response with text length = 7Fh, DCS = 04h `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	

Id	Description	API Expectation	APDU Expectation
11	**Build and send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response with text length = EFh, DCS = 08h `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 Proactive command

	Terminal Response with 2 Text String TLV
	`1st Text String TLV = 0D 02 04 41`
	`2nd Text String TLV = 0D 03 08 42 43`		

	Call the getTextStringCodingScheme() method	Result is 04h	
14	**Call the getValueLength() method**	Result is 02h	

6.2.8.6.4 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

6.2.8.7 Method GetTextStringLength

Test Area Reference: API_2_PRH_GTTL

6.2.8.7.1 Conformance requirement

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

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

6.2.8.7.1.1 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.

6.2.8.7.1.2 Parameter errors

No requirements.

6.2.8.7.1.3 Context errors

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

6.2.8.7.2 Test Suite files

Test Script: API_2_PRH_GTTL_1.scr

Test Applet: API_2_PRH_GTTL_1.java

Load Script: API_2_PRH_GTTL_1.ldr

Cleanup Script: API_2_PRH_GTTL_1.clr

Parameter File: API_2_PRH_GTTL_1.par

6.2.8.7.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive 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 Proactive command
	Terminal Response with a null Text String TLV `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 Proactive command
	Terminal Response with text length = 01h, DCS = 04h `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 Proactive command
	Terminal Response with text length = 02h, DCS = 00h `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 Proactive command
	Terminal Response with text length = 7Eh, DCS = 08h `Text String TLV = 0D 7F 08 01 02 … 7E`		
	Call the getTextStringLength() method	Result is 7Eh	
9	**Call the getValueLength() method**	Result is 7Fh	

Id	Description	API Expectation	APDU Expectation
10	**Build and send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response with text length = 7Fh, DCS = 04h `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 Proactive command
	Terminal Response with text length = EFh, DCS = 04h `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 Proactive command
	Terminal Response with 2 Text String TLV 1st `Text String TLV = 0D 02 04 41` 2nd `Text String TLV = 0D 03 08 42 43`		
	Call the getTextStringLength() method	Result is 01h	
15	**Call the getValueLength() method**	Result is 02h	

6.2.8.7.4 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

6.2.8.8 Method getTheHandler

Test Area Reference: API_2_PRH_GTHD

6.2.8.8.1 Conformance requirement

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

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

6.2.8.8.1.1 Normal execution

- CRRN1: The method shall return the single system instance of the ProactiveHandler class.
- CRRN2: The EnvelopeHandler is a Temporary JCRE Entry Point Object

6.2.8.8.1.2 Parameter errors

No requirements.

6.2.8.8.1.3 Context errors

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

6.2.8.8.2 Test Suite files

Test Script:	API_2_PRH_GTHD_1.scr
Test Applet:	API_2_PRH_GTHD_1.java
Load Script:	API_2_PRH_GTHD_1.ldr
Cleanup Script:	API_2_PRH_GTHD_1.clr
Parameter File:	API_2_PRH_GTHD_1.par

6.2.8.8.3 Test procedure

Id	Description	API Expectation	APDU Expectation
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	

6.2.8.8.4 Test Coverage

CRR number	Test case number
N1	1, 2, 3
N2	Checked in Framework tests: FWK_API_HEPO (test case 4)
C1	Checked in Framework tests: FWK_MHA_PRHD

6.2.8.9 Method getLength

Test Area Reference API_2_PRH_GLEN

6.2.8.9.1 Conformance requirement

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

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

6.2.8.9.1.1 Normal execution

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

6.2.8.9.1.2 Parameter errors

No requirements.

6.2.8.9.1.3 Context errors

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

6.2.8.9.2 Test Suite files

Test Script:	API_2_PRH_GLEN_1.scr
Test Applet:	API_2_PRH_GLEN_1.java
Load Script:	API_2_PRH_GLEN_1.ldr
Cleanup Script:	API_2_PRH_GLEN_1.clr
Parameter File:	API_2_PRH_GLEN_1.par

6.2.8.9.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Build and send a Display Text command**		DISPLAY TEXT Proactive command
	Terminal Response without additional information in General Result TLV		
	ProactiveResponseHandler.getTheHandler() getLength()	Result of getLength() is 12	
2	**Build and send a Display Text command**		DISPLAY TEXT Proactive command
	Terminal Response with F2h additional information in General Result TLV		
	ProactiveResponseHandler.getTheHandler() getLength()	Result of getLength() is FFh	

6.2.8.9.4 Test Coverage

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

6.2.8.10 Method copy

Test Area Reference API_2_PRH_COPY_BSS

6.2.8.10.1 Conformance requirement

The method with following header 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
```

6.2.8.10.1.1 Normal execution

- CRRN1: copies the simple TLV list contained in the handler to the destination byte array.
- CRRN2: returns dstOffset + dstLength.

6.2.8.10.1.2 Parameter errors

- 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.

6.2.8.10.1.3 Context errors

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

6.2.8.10.2 Test Suite files

Test Script:	API_2_PRH_COPY_BSS_1.scr
Test Applet:	API_2_PRH_COPY_BSS_1.java
Load Script:	API_2_PRH_COPY_BSS_1.ldr
Cleanup Script:	API_2_PRH_COPY_BSS_1.clr
Parameter File:	API_2_PRH_COPY_BSS_1.par

6.2.8.10.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a DISPLAY TEXT command**		DISPLAY TEXT Proactive command
	Terminal Response without Additional Information in General Result TLV: `81 03 01 21 00 02 02 82 81 03 01 00`		
	ProactiveResponseHandler.getTheHandler() copy() with NULL as parameter to dstBuffer	NullPointerException is thrown	

2	**dstOffset > dstBuffer.length**
	`dstBuffer.length = 5`
	`dstOffset = 6`
	`dstLength = 0`
		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 = 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 with buffer:** `81 03 01 21 00 02 02 82 81 03 01 00`	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	

Id	Description	API Expectation	APDU Expectation

11	**Compare the whole buffer**
	`Reference =`
	`00 01 02`
	`81 03 01 21 00`
	`02 02 82 81`
	`03 01 00`
	`0F 10 11 12 13`
		Result of arrayCompare() is 0	

12	**Initialise dstBuffer** `dstBuffer = 00h 01h 02h … 13h`		

	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**
	`Reference =`
	`00 01 02`
	`81 03 01 21 00`
	`02 02 82 81`
	`0C 0D 0E`
	`0F 10 11 12 13`
		Result of arrayCompare() is 0	

6.2.8.10.4 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

6.2.8.11 Method findTLV

Test Area Reference API_2_PRH_FINDBB

6.2.8.11.1 Conformance requirement

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

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

6.2.8.11.1.1 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, the current TLV is no longer defined and TLV_NOT_FOUND is returned.
- CRRN5: The search method is comprehension required flag independent.

6.2.8.11.1.2 Parameter errors

- 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.

6.2.8.11.1.3 Context errors

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

6.2.8.11.2 Test Suite files

Test Script:	API_2_PRH_FINDBB_1.scr
Test Applet:	API_2_PRH_FINDBB_1.java
Load Script:	API_2_PRH_FINDBB_1.ldr
Cleanup Script:	API_2_PRH_FINDBB_1.clr
Parameter File:	API_2_PRH_FINDBB_1.par

6.2.8.11.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a DISPLAY TEXT command**		DISPLAY TEXT Proactive command

	Terminal Response with 2 General Result TLV
	`81 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_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	**Select a TLV (tag 02h)**		

	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 83h**
	`Tag = 83h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_NOT_SET	

15	**Search tag 82h**
	`Tag = 82h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_SET	

6.2.8.11.4 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

6.2.8.12 Method getValueLength

Test Area Reference API_2_PRH_GVLE

6.2.8.12.1 Conformance requirement

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

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

6.2.8.12.1.1 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.

6.2.8.12.1.2 Parameter errors

No requirements.

6.2.8.12.1.3 Context errors

- 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.

6.2.8.12.2 Test Suite files

Test Script:	API_2_PRH_GVLE_1.scr
Test Applet:	API_2_PRH_GVLE_1.java
Load Script:	API_2_PRH_GVLE_1.ldr
Cleanup Script:	API_2_PRH_GVLE_1.clr
Parameter File:	API_2_PRH_GVLE_1.par

6.2.8.12.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response `Text String TLV = 0D 00`		

	ProactiveResponseHandler.getTheHandler()
	`GetValueLength()`
		ToolkitException.UNAVAILABLE_ ELEMENT is thrown	

2	**Search TLV 0Dh**		
	`getValueLength()`	Result is 00h	

Id	Description	API Expectation	APDU Expectation
3	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response `Text String TLV = 0D 02 04 41`		
	Search TLV 0Dh (Text String TLV)		
	`getValueLength()`	Result is 02h	
4	**Send a GET INPUT command**		GET INPUT Proactive 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 Proactive 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 Proactive 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	

6.2.8.12.4 Test Coverage

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

6.2.8.13 Method getValueByte

Test Area Reference API_2_PRH_GVBYS

6.2.8.13.1 Conformance requirement

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

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

6.2.8.13.1.1 Normal execution

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

6.2.8.13.1.2 Parameter errors

- 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.

6.2.8.13.1.3 Context errors

- 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.

6.2.8.13.2 Test Suite files

Test Script:	API_2_PRH_GVBYS_1.scr
Test Applet:	API_2_PRH_GVBYS_1.java
Load Script:	API_2_PRH_GVBYS_1.ldr
Cleanup Script:	API_2_PRH_GVBYS_1.clr
Parameter File:	API_2_PRH_GVBYS_1.par

6.2.8.13.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 7Eh `Text String TLV = 0D 7F 04 01 02 … 7E`		
	ProactiveResponseHandler.getTheHandler()		
	`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 Proactive command
	Terminal Response, Text String length = EFh `Text String TLV = 0D 81 F0 04 01 02 … 7E 7F … EF`		
	Search TLV 0Dh (Text String TLV)		
	`getValueByte(7E)`	Result is 7Eh	
7	`GetValueByte(7F)`	Result is 7Fh	
8	`GetValueByte(EF)`	Result is EFh	

6.2.8.13.4 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

6.2.8.14 Method copyValue

Test Area Reference API_2_PRH_CPYVS_BSS

6.2.8.14.1 Conformance requirement

The method with following header 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
```

6.2.8.14.1.1 Normal execution

- CRRN1: copies a part of the last TLV element which has been found, into a destination. buffer.
- CRRN2: returns dstOffset + dstLength.

6.2.8.14.1.2 Parameter errors

- 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.

6.2.8.14.1.3 Context errors

- 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.

6.2.8.14.2 Test Suite files

Test Script:	API_2_PRH_CPYVS_BSS_1.scr
Test Applet:	API_2_PRH_CPYVS_BSS_1.java
Load Script:	API_2_PRH_CPYVS_BSS_1.ldr
Cleanup Script:	API_2_PRH_CPYVS_BSS_1.clr
Parameter File:	API_2_PRH_CPYVS_BSS_1.par

6.2.8.14.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 5 `Text String TLV = 0D 06 04 01 02 … 05`		

	ProactiveResponseHandler.getTheHandler()
	`Select Text String TLV`		

	`CopyValue() with a null dstBuffer`	NullPointerException is thrown	

2	**dstOffset > dstBuffer.length**
	`dstBuffer.length = 5`
	`dstOffset = 6`
	`dstLength = 0`
		ArrayIndexOutOfBoundsException is thrown	

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

Id	Description	API Expectation	APDU Expectation

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 = 7`
	`dstBuffer.length = 15`
	`dstOffset = 0`
	`dstLength = 0`
		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 Proactive command
	Terminal Response, Text String length = 16 `Text String TLV = 0D 11 04 00 01 … 0F`		
	`ProactiveResponseHandler.getTheHandler`		
	`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	

6.2.8.14.4 Test Coverage

CRR number	Test case number
N1	13, 15
N2	12, 14
P1	1

CRR number	Test case number
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for Proactive Response Handler
C2	11

6.2.8.15 Method compareValue

Test Area Reference API_2_PRH_CPRVS_BSS

6.2.8.15.1 Conformance requirement

The method with following header 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
```

6.2.8.15.1.1 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.

6.2.8.15.1.2 Parameter errors

- 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.

6.2.8.15.1.3 Context errors

- 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.

6.2.8.15.2 Test Suite files

Test Script:	API_2_PRH_CPRVS_BSS_1.scr
Test Applet:	API_2_PRH_CPRVS_BSS_1.java
Load Script:	API_2_PRH_CPRVS_BSS_1.ldr
Cleanup Script:	API_2_PRH_CPRVS_BSS_1.clr
Parameter File:	API_2_PRH_CPRVS_BSS_1.par

6.2.8.15.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 5 `Text String TLV = 0D 06 04 01 02 … 05`		

	ProactiveResponseHandler.getTheHandler()
	`Select Text String TLV`		

	compareValue() with a null compareBuffer	NullPointerException is thrown	

2	**compareOffset > compareBuffer.length**
	`compareBuffer.length = 5`
	`compareOffset = 6`
	`compareLength = 0`
		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 = 7`
	`compareBuffer.length = 15`
	`compareOffset = 0`
	`compareLength = 0`
		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 Proactive command
	Terminal Response, Text String length = 16 `Text String TLV = 0D 11 04 00 01 … 0F`		
	`ProactiveResponseHandler.getTheHandler()`		
	`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	

Id	Description	API Expectation	APDU Expectation
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	

6.2.8.15.4 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

6.2.8.16 Method findAndCopyValue(byte tag, byte[] dstBuffer, short valueOffset)

Test Area Reference API_2_PRH_FACYB_BS

6.2.8.16.1 Conformance requirement

The method with following header 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
```

6.2.8.16.1.1 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.

6.2.8.16.1.2 Parameter errors

- CRRP1: if dstBuffer is null NullPointerException shall be thrown.
- CRRP2: if dstOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

6.2.8.16.1.3 Context errors

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

6.2.8.16.2 Test Suite files

Test Script:	API_2_PRH_FACYB_BS_1.scr
Test Applet:	API_2_PRH_FACYB_BS_1.java
Load Script:	API_2_PRH_FACYB_BS_1.ldr
Cleanup Script:	API_2_PRH_FACYB_BS_1.clr
Parameter File:	API_2_PRH_FACYB_BS_1.par

6.2.8.16.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 15 `Text String TLV = 0D 10 04 01 02 … 0F`		
	`ProactiveResponseHandler.getTheHandler()`		
	`FindAndCopyValue() with a null dstBuffer`	NullPointerException is thrown	

2	**dstOffset > dstBuffer.length**
	`tag = 0Dh`
	`dstBuffer.length = 20`
	`dstOffset = 21`
		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	

Id	Description	API Expectation	APDU Expectation
6	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 16 `Text String TLV = 0D 11 04 00 01 … 0F`		
	ProactiveResponseHandler.getTheHandler()		
	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 findAndcopyValue() 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 findAndcopyValue() 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 Proactive command

	Terminal Response, with 2 Text String TLV
	`0D 11 04 00 01 … 0F`
	`0D 02 04 41`		

	ProactiveResponseHandler.getTheHandler()		

	Successful call
	`Tag = 0Dh`
	`DstBuffer.length = 17`
	`DstOffset = 0`
		Result of findAndcopyValue() is 17	

12	**Compare buffer** `Buffer = 04 00 01 … 0F`	Result is 00h	
13	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 16 `Text String TLV = 0D 11 04 00 01 … 0F`		
	ProactiveResponseHandler.getTheHandler()		

	Successful call (with tag 8Dh)
	`Tag = 8Dh`
	`DstBuffer.length = 17`
	`DstOffset = 0`
		Result of findAndcopyValue() is 17	

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

6.2.8.16.4 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

6.2.8.17 Method findAndCopyValue(byte tag, byte occurence, short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)

Test Area Reference API_2_PRH_FACYBBS_BSS

6.2.8.17.1 Conformance requirement

The method with following header 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
```

6.2.8.17.1.1 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.

6.2.8.17.1.2 Parameter errors

- 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.

6.2.8.17.1.3 Context errors

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

6.2.8.17.2 Test Suite files

Test Script:	API_2_PRH_FACYBBS_BSS_1.scr
Test Applet:	API_2_PRH_FACYBBS_BSS_1.java
Load Script:	API_2_PRH_FACYBBS_BSS_1.ldr
Cleanup Script:	API_2_PRH_FACYBBS_BSS_1.clr
Parameter File:	API_2_PRH_FACYBBS_BSS_1.par

6.2.8.17.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 15 `Text String TLV = 0D 10 04 01 02 … 0F`		
	ProactiveResponseHandler.getTheHandler()		
	findAndCopyValue() with a null dstBuffer	NullPointerException is thrown	

2	**dstOffset > dstBuffer.length**
	`tag = 0Dh, occurrence = 1`
	`valueOffset = 0`
	`dstBuffer.length = 5`
	`dstOffset = 6`
	`dstLength = 0`
		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	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 5 `Text String TLV = 0D 06 04 01 02 … 05`		
	ProactiveResponseHandler.getTheHandler()		

	valueOffset > Text String Length
	`tag = 0Dh, occurrence = 1`
	`valueOffset = 7`
	`dstBuffer.length = 15`
	`dstOffset = 0`
	`dstLength = 0`
		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 Proactive command
	Terminal Response, Text String length = 16 `Text String TLV = 0D 11 04 00 01 … 0F`		
	ProactiveResponseHandler.getTheHandler()		
	Select a TLV (tag 02h)		

	findAndCopyValue()
	`tag = 0Dh`
	`occurrence = 2`
		ToolkitException.UNAVAILABLE_ELEMENT is thrown	

	Call the getValueLength() method	ToolkitException.UNAVAILABLE_ELEMENT is thrown.	

Id	Description	API Expectation	APDU Expectation

12	**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 findAndcopyValue() 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 Proactive command

	Terminal Response, with 2 Text String TLV
	`0D 11 04 00 01 02 … 0F`
	`0D 06 00 11 22 33 44 55 (no specific DCS byte)`		

	ProactiveResponseHandler.getTheHandler()		

	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 Proactive command
	Terminal Response, Text String length = 16 `Text String TLV = 0D 11 04 00 01 … 0F`		
	ProactiveResponseHandler.getTheHandler()		

	Successful call (with tag 8Dh)
	`Tag = 8Dh, occurrence = 1`
	`ValueOffset = 0`
	`DstBuffer.length = 17`
	`DstOffset = 0`
	`DstLength = 17`
		Result of findAndcopyValue() is 17	

21	**Compare buffer** `Buffer = 04 00 01 … 0F`	Result is 00h	

6.2.8.17.4 Test Coverage

CRR number	Test case number
N1	13, 15, 17, 19
N2	11
N3	12, 14, 16, 18
N4	20, 21

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

6.2.8.18 Method findAndCompareValue(byte tag, byte[] compareBuffer, short compareOffset)

Test Area Reference API_2_PRH_FACRB_BS

6.2.8.18.1 Conformance requirement

The method with following header 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
```

6.2.8.18.1.1 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.

6.2.8.18.1.2 Parameter errors

- CRRP1: if compareBuffer is null NullPointerException shall be thrown.
- CRRP2: if compareOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

6.2.8.18.1.3 Context errors

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

6.2.8.18.2 Test Suite files

Test Script:	API_2_PRH_FACRB_BS_1.scr
Test Applet:	API_2_PRH_FACRB_BS_1.java
Load Script:	API_2_PRH_FACRB_BS_1.ldr
Cleanup Script:	API_2_PRH_FACRB_BS_1.clr
Parameter File:	API_2_PRH_FACRB_BS_1.par

6.2.8.18.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 15 `Text String TLV = 0D 10 04 01 02 … 0F`		
	ProactiveResponseHandler.getTheHandler()		
	FindAndCompareValue() with a null dstBuffer	NullPointerException is thrown	

2	**compareOffset > compareBuffer.length**
	`tag = 0Dh`
	`compareBuffer.length = 20`
	`compareOffset = 21`
		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	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 16 `Text String TLV = 0D 11 04 00 01 … 0F`		
	ProactiveResponseHandler.getTheHandler()		
	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	

Id	Description	API Expectation	APDU Expectation
12	**Send a GET INPUT command**		GET INPUT Proactive command

	Terminal Response, with 2 Text String TLV
	`0D 11 04 00 01 … 0F`
	`0D 06 00 11 22 33 44 55`		

	ProactiveResponseHandler.getTheHandler()		

	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 Proactive command
	Terminal Response, Text String length = 16 `Text String TLV = 0D 11 04 00 01 … 0F`		
	ProactiveResponseHandler.getTheHandler()		

	Initialise compareBuffer
	`CompareBuffer =`
	`04 00 01 … 0F`		

	Compare buffers (with tag 8Dh)
	`Tag = 8Dh`
	`CompareOffset = 0`
		Result is 00h	

6.2.8.18.4 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

6.2.8.19 Method findAndCompareValue(byte tag, byte occurence, short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)

Test Area Reference API_2_PRH_FACRBBS_BSS

6.2.8.19.1 Conformance requirement

The method with following header 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
```

6.2.8.19.1.1 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.

6.2.8.19.1.2 Parameter errors

- 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.

6.2.8.19.1.3 Context errors

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

6.2.8.19.2 Test Suite files

Test Script:	API_2_PRH_FACRBBS_BSS_1.scr
Test Applet:	API_2_PRH_FACRBBS_BSS_1.java
Load Script:	API_2_PRH_FACRBBS_BSS_1.ldr
Cleanup Script:	API_2_PRH_FACRBBS_BSS_1.clr
Parameter File:	API_2_PRH_FACRBBS_BSS_1.par

6.2.8.19.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 15 `Text String TLV = 0D 10 04 01 02 … 0F`		
	ProactiveResponseHandler.getTheHandler()		
	findAndCompareValue() with a null compareBuffer	NullPointerException is thrown	

2	**compareOffset > compareBuffer.length**
	`tag = 0Dh, occurrence = 1`
	`valueOffset = 0`
	`compareBuffer.length = 5`
	`compareOffset = 6`
	`compareLength = 0`
		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	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 5 `Text String TLV = 0D 06 04 01 02 … 05`		
	ProactiveResponseHandler.getTheHandler()		

	valueOffset ≥ Text String Length
	`tag = 0Dh, occurrence = 1`
	`valueOffset = 7`
	`compareBuffer.length = 15`
	`compareOffset = 0`
	`compareLength = 0`
		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	

Id	Description	API Expectation	APDU Expectation
12	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 16 `Text String TLV = 0D 11 04 00 01 … 0F`		
	ProactiveResponseHandler.getTheHandler()		
	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	

Id	Description	API Expectation	APDU Expectation
20	**Send a GET INPUT command**		GET INPUT Proactive command

	Terminal Response, with 2 Text String TLV
	`0D 11 04 00 01 … 0F`
	`0D 06 00 11 22 33 44 55`		

	ProactiveResponseHandler.getTheHandler()		

	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	**Send a GET INPUT command**		GET INPUT Proactive command
	Terminal Response, Text String length = 16 `Text String TLV = 0D 11 04 00 01 … 0F`		
	ProactiveResponseHandler.getTheHandler()		

	Initialise compareBuffer
	`CompareBuffer =`
	`04 00 01 … 0F`		

	Compare buffers (with tag 8Dh)
	`tag = 8Dh, occurrence = 1`
	`valueOffset = 0`
	`compareOffset = 0`
	`compareLength = 17`
		Result is 00h	

6.2.8.19.4 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

6.2.8.20 Method getCapacity

Test Area Reference: API_2_PRH_GCAP

6.2.8.20.1 Conformance Requirement

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

```
public byte getCapacity()
```

6.2.8.20.1.1 Normal execution

- CRRN1: The method shall return the maximum size of the Simple TLV list managed by the handler.

6.2.8.20.2 Test suite files

Test Script:	API_2_PRH_GCAP_1.scr
Test Applet:	API_2_PRH_GCAP_1.java
Load Script:	API_2_PRH_GCAP_1.ldr
Cleanup Script:	API_2_PRH_GCAP_1.clr
Parameter File:	API_2_PRH_GCAP_1.par

6.2.8.20.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**ProactiveResponseHandler available** 1- Send envelope SMS-PP Formatted 2- The applet sends a proactive command 3- Fetch the proactive command and send Terminal Response 4- The applet calls method getCapacity() method 5- The applet calls method getLength() method	1- Applet is triggered 4-No exception is thrown 5- The Capacity result is greater or equal to getLength() result	2- 91 XX 3- The proactive command is fetched

6.2.8.20.4 Test Coverage

CRR number	Test case number
N1	1

6.2.8.21 Method getChannelIdentifier

Test Area Reference: API_2_PRH_GCID

6.2.8.21.1 Conformance Requirement:

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

```
public byte getChannelIdentifier()
                              throws ToolkitException
```

6.2.8.21.1.1 Normal execution

- CRRN1:The method shall return the channel identifier byte value.
- CRRN2:The channel identifier byte value returned shall be from the first Channel status TLV element.
- CRRN3: If the element is available it becomes the currently selected TLV.

6.2.8.21.1.2 Context errors

- CRRC1: The method shall throw ToolkitException (UNAVAILABLE_ELEMENT) if the Channel status TLV is not present.

- CRRC2: The method shall throw ToolkitException (OUT_OF_TLV_BOUNDARIES) if the Simple TLV Channel Status length is equal to 0.

6.2.8.21.2 Test suite files

Test Script:	API_2_PRH_GCID_1.scr
Test Applet:	API_2_PRH_GCID_1.java
Load Script:	API_2_PRH_GCID_1.ldr
Cleanup Script:	API_2_PRH_GCID_1.clr
Parameter File:	API_2_PRH_GCID_1.par

6.2.8.21.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	Applet1 is installed with maximum number of channel = 01.		
1	**Channel status TLV is not present** 1- Build and send a DISPLAY TEXT command 2- Call ProactiveResponseHandler.getChannelIdentifier() method.	2- UNAVAILABLE_ELEMENT ToolkitException is thrown	1- DISPLAY TEXT Proactive command is fetched. TERMINAL RESPONSE with no Channel status TLV available.
2	**Channel status TLV with a length equal to 0** 1- Build and send a OPEN CHANNEL proactive command 2- Call ProactiveResponseHandler.getChannelIdentifier() method.	2- OUT_OF_TLV_BOUNDARIES ToolkitException is thrown	1- OPEN CHANNEL Proactive command is fetched. TERMINAL RESPONSE with Channel status TLV length equal to 0.
3	**Get channel identifier value** 1- Call ProactiveHandler.init() and ProactiveHandler.send() methods to open a channel. 2- Call ProactiveResponseHandler.getChannelIdentifier() method. 3- Call ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods.	2- Returns 0x01	1- OPEN CHANNEL Proactive Command is fetched. TERMINAL RESPONSE is issued with channel status value = 0x8100.
4	**Get channel identifier value with 2 TLV** 1- Call ProactiveHandler.init()and ProactiveHandler.send() methods to open a channel 2- Call ProactiveResponseHandler.getChannelIdentifier() 3- Call ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods.	2- Returns 0x01	1- OPEN CHANNEL Proactive Command is fetched. TERMINAL RESPONSE is issued with channel status value = 0x8100 and 0x8200.
5	**Channel status TLV is currently selected TLV** 1- Call ProactiveHandler.init() and ProactiveHandler.send() methods to open a channel. ViewHandler.FindTLV with Device Identity Tag. 2- Call ProactiveResponseHandler.getChannelIdentifier() method. 3- Compare ProactiveResponseHandler.getChannelIdentifier() and then ViewHandler.getValueByte(0) methods.	2- Returns 0x03 3- Check getChannelIdentifier() =getValueByte(0)	1- OPEN CHANNEL Proactive Command is fetched. TERMINAL RESPONSE is issued with channel status value = 0x0305~~.~~

6.2.8.21.4 Test Coverage

CRR number	Test case number
N1	3
N2	4
N3	5
C1	1

C2	2

6.2.8.22 Method copyChannelData

Test Area Reference: API_2_PRH_CCHD_BSS

6.2.8.22.1 Conformance Requirement:

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

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

6.2.8.22.1 Normal execution

- CRRN1: The method shall copy a part of the Channel data string field.
- CRRN2: The Channel data string field value returned shall be the first Channel data TLV element of the current response data field.
- CRRN3: If the element is available it becomes the currenly selected TLV.
- CRRN4: Returns dstOffset + dstLength.

6.2.8.22.2 Parameters error

- CRRP1: If dstBuffer is null, a NullPointerException is thrown.
- CRRP2: If dstOffset or dstLength parameter is negative an ArrayIndexOutOfBoundsException exception is thrown and no copy is performed.
- CRRP3: If dstOffset+dstLength is greater than dstBuffer.length, the length of the dstBuffer array an ArrayIndexOutOfBoundsException exception is thrown and no copy is performed.
- CRRP4: If dstLength is greater than the value field of the available TLV, a OUT_OF_TLV_BOUNDARIES ToolkitException is thrown.

6.2.8.22.3 Context errors

- CRRC1: The method shall throw a UNAVAILABLE_ELEMENT ToolkitException if the Result TLV is not present.

6.2.8.22.2 Test suite files

Test Script:	API_2_PRH_CCHD_BSS_1.scr
Test Applet:	API_2_PRH_CCHD_BSS_1.java
Load Script:	API_2_PRH_CCHD_BSS_1.ldr
Cleanup Script:	API_2_PRH_CCHD_BSS_1.clr
Parameter File:	API_2_PRH_CCHD_BSS_1.par

6.2.8.22.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	1- Applet1 is installed with maximum number of channel = 01. 2- Applet1 builds proactive commands OPEN CHANNEL with init() method in order to open one channel. ProactiveHandler.send() method is called.		2- OPEN CHANNEL proactive command is fetched TERMINAL RESPONSE is issued with Channel Id = 01

1	**CopyChannelData() with NULL dstBuffer**
	Build and send a RECEIVE DATA command
	Call ProactiveResponseHandler.copyChannelData
	dstBuffer = NULL
	dstOffset = 0
	dstLength = 1
		NullPointerException is thrown	RECEIVE DATA Proactive command is fetched.
			TERMINAL RESPONSE with not empty Channel Data TLV is issued.

2	**CopyChannelData() with negative dstOffset** 1- call init() method for the RECEIVE DATA proactive command. 2- call ProactiveResponseHandler.copyChannelData() dstBuffer.length = 8 dstOffset = -1 dstLength = 1 3- check dstBuffer is empty.	2- an ArrayIndexOutOfBoundsException exception is thrown. 3- no copy is performed.	1- RECEIVE DATA proactive command is fetched. TERMINAL RESPONSE with 6 bytes avalaible ("Hello1")

3	**CopyChannelData() with negative dstLength**
	1- call ProactiveResponseHandler.copyChannelData()
	dstBuffer.length = 8
	dstOffset = 0
	dstLength = -1
	2- check dstBuffer is empty.
		1- an ArrayIndexOutOfBoundsException exception is thrown.
		2- no copy is performed.	

4	**CopyChannelData() with dstOffset+dstLength greater than dstBuffer.length**
	1- call ProactiveResponseHandler.copyChannelData() with dstOffset+dstLength greater than dstBuffer.length.
	dstBuffer.length = 8
	dstOffset = 5
	dstLength = 5
	2- check dstBuffer is empty.
		1- an ArrayIndexOutOfBoundsException exception is thrown.
		2- no copy is performed.	

5	**CopyChannelData() with dstLength too large**
	Call ProactiveResponseHandler.copyChannelData() with dstLength greater than the value field of the available TLV.
	dstBuffer.length = 8
	dstOffset = 0
	dstLength = 7
		a OUT_OF_TLV_BOUNDARIES ToolkitException is thrown.	

6	**CopyChannelData() without Channel Data TLV element** 1- call init() method for the RECEIVE DATA proactive command. Call send() method. 2- call ProactiveResponseHandler.copyChannelData() dstBuffer.length = 8 dstOffset = 0	2- a UNAVAILABLE_ELEMENT ToolkitException is thrown.	1- RECEIVE DATA proactive command is fetched TERMINAL RESPONSE without ChannelData TLV element.

	dstLength = 6		

7	**Successful copyChannelData()**
	`Call init() method for the RECEIVE DATA proactive command.`
	`Call send() method.`
	`2- Call findTLV() with TAG of DEVICE IDENTITY.`
	`3- Call ProactiveResponseHandler.copyChannelData()`
	`dstBuffer.length = 8`
	`dstOffset = 0`
	`dstLength = 6`
	`dstBuffer is the whole Buffer.`
		3- the Channel Data TLV is copied into dstBuffer.
		The applet checks the returned value is dstOffset + dstLength = 6.
			1- RECEIVE DATA proactive command is fetched
			TERMINAL RESPONSE with one Channel data TLV element. (6 bytes available = "Hello2")

8	**Compare copied Buffer** `Check dstBuffer.`	The applet checks that dstBuffer contains the channel data from the TERMINAL RESPONSE.	
9	**Check the Channel Data TLV is selected** `Call the ViewHandler.getValueByte(0) method`	The returned byte is the same than the first byte of the Channel data TLV (i.e. "H")	

10	**Successful copyChannelData()**
	`Call ProactiveResponseHandler.copyChannelData()`
	`dstBuffer.length = 8`
	`dstOffset = 2`
	`dstLength = 3`
	`DstBuffer is a part of Buffer.`
		The Channel Data TLV is copied into dstBuffer.
		The applet checks the returned value is dstOffset + dstLength = 5.	

11	**Compare copied Buffer** `Check dstBuffer.`	The applet checks that bytes from 2 to 4 of dstBuffer contain the first 3 bytes of channel data TLV from the TERMINAL RESPONSE.	

12	**Successful copyChannelData()**
	`1- Initialise dstBuffer to [00, 01…]`
	`2- Call ProactiveResponseHandler.copyChannelData()`
	`dstBuffer.length = 8`
	`dstOffset = 2`
	`dstLength = 3`
	`DstBuffer is a part of buffer.`
		2- The Channel Data TLV is copied into dstBuffer.
		The returned value is dstOffset + dstLength = 5.	

13	**Compare copied Buffer** `Check dstBuffer.`	The applet checks that only bytes from 2 to 4 of dstBuffer have been updated with the first 3 bytes of channel data TLV from the TERMINAL RESPONSE.	

14	**Successful copyChannelData(), with 2 TLV**
	`1- call init() method for the RECEIVE DATA proactive command.`
	`Call send() method.`
	`2- call ProactiveResponseHandler.copyChannelData() with dstLength lower than the value field of the available TLV.`
	`dstBuffer.length = 8`
	`dstOffset = 0`
	`dstLength = 6`
		2- the first Channel Data TLV is copied into dstBuffer.
		The returned value is dstOffset+dstLength =0x06
			1- RECEIVE DATA proactive command is fetched
			TERMINAL RESPONSE with two Channel data TLV element
			1st TLV : 6 bytes available = "Hello3"
			2nd TLV : 6 bytes available = "Hello4"

15	**Compare copied Buffer**	Check that dstBuffer contains the	

	Check dstBuffer.	first Channel Data TLV from the TERMINAL RESPONSE.	

6.2.8.22.4 Test Coverage

CRR number	Test case number
N1	7, 10, 12, 14
N2	14
N3	9
N4	8, 11, 13, 15
P1	1
P2	2, 3
P3	4
P4	5
C1	6

6.2.9 Class ToolkitRegistry

6.2.9.1 Method allocateTimer

Test Area Reference: API_2_TKR_ATIM

6.2.9.1.1 Conformance requirement:

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

```
public byte allocateTimer()
                   throws ToolkitException
```

6.2.9.1.1.1 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.

6.2.9.1.1.2 Parameters error

No requirements.

6.2.9.1.1.3 Context errors

- 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.

6.2.9.1.2 Test suite files

Test Script:	API_2_TKR_ATIM_1.scr
Test Applet:	API_2_TKR_ATIM_1.java
	API_2_TKR_ATIM_2.java
	API_2_TKR_ATIM_3.java

- Installation parameters:
- For this test procedure the non-volatile memory of each instance is 200 (Hexa).
- The maximum timer parameter value is as follows for each applet:
  - applet1 (API_2_TKR_ATIM_1): 8 timers
  - applet2 (API_2_TKR_ATIM_2): 4 timers
  - applet3 (API_2_TKR_ATIM_3): 0 timer

Load Script: API_2_TKR_ATIM_1.ldr

- The load script installs the 6 instances.

Cleanup Script: API_2_TKR_ATIM_1.clr

Parameter File: API_2_TKR_ATIM_1.par

6.2.9.1.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Allocates up to 8 timers (applet1)** `8 * allocateTimer().`	No exception shall be thrown. Timer ID returned shall be between 01 and 08 inclusive. It shall be different after each call.	
2	**Allocate timers more than the maximum (applet1)** `The applet1 allocates 1 more timer.`	Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE.	

3	**Check applet is Triggered by ENVELOPE(TIMER_EXPIRATION) command (applet1)**
	`Send ENVELOPE(TIMER EXPIRATION) with all timers id (not in an increase order).`
	`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 (applet2)** `4 * allocateTimer().`	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.	
5	**Allocate timers more than the maximum (applet3)** `The applet3 allocates 1 more timer.`	Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE.	

6.2.9.1.4 Test Coverage

CRR number	Test case number
N1	1, 4
N2	1, 4
N3	3
N4	1
C1	2
C2	5

6.2.9.2 Method changeMenuEntry

Test Area Reference: API_2_TKR_CMETB_BSSBZBS

6.2.9.2.1 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
```

6.2.9.2.1.1 Normal execution

- CRRN1: The SIM Toolkit Framework shall dynamically 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: The default state of the changed menu entry is 'enabled'.

- CRRN3: a call to isEventSet() method on EVENT_MENU_SELECTION shall return true before and after the call.

- CRRN4: if helpSupported was true then a call to isEventSet() method on EVENT_MENU_SELECTION_HELP_REQUEST event shall return true.

- CRRN5: 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.

- CRRN6: if help supported was true, the SIM Toolkit Framework shall issue a SETUP MENU command with command qualifier = '80'

- CRRN7: 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 .

- CRRN8: 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.

- CRRN9: 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.

- CRRN10: 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.

- CRRN11: If Next Action Indicator was different from '00', 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 as defined in 3GPP TS 51.014 [4].

6.2.9.2.1.2 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

6.2.9.2.1.3 Context errors

- CRRC1: Shall throw a ToolkitException with MENU_ENTRY_NOT_FOUND reason if the Menu Identifier 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.

6.2.9.2.2 Test suite files

Additional requirements for the GSM personalization:

- content of EF sume shall be:
  - Title Alpha Identifier: "TOOLKIT TEST"

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:

  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_1.ldr

Cleanup Script: API_2_TKR_CMETB_BBSSBZBS_1.clr

Parameter File: API_2_TKR_CMETB_BBSSBZBS_1.par

6.2.9.2.3 Test procedure

Id	Description	API Expectation	APDU Expectation

1	**Applet changes the entry's title by menuEntry buffer, with a greater length than the initial length**
	1- ChangeMenuEntry()with parameters:
	Id = '02'
	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).
		1- No exception shall be thrown.
		2- shall return true.
		3- shall return false.
			The SIM shall issue a SETUP MENU proactive command which contains the new text for entry ID '02'.

Id	Description	API Expectation	APDU Expectation

2	**Changing the title with part of menuEntry buffer**
	1- changeMenuEntry()with parameters:
	Id = '01'
	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.
			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 '01' and entry '02', with parameters:
	Id = '01'/'02'
	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.
			The SIM shall issue a SETUP MENU proactive command which contains for entry '01'and entry '02', no text part.

4	**Setting a next action indicator != 0**
	1- changeMenuEntry()with parameters:
	Id = '02'
	MenuEntry = "NextActionIndic"
	Offset = 0
	Length = menuEntry.length
	NextAction = '10' (SETUP CALL)
	HelpSupported = false
	IconQualifier = 0
	IconIdentifier = 0
	2- isEventSet(EVENT_MENU_SELECTION).
	3- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST).
	4- changeMenuEntry()with parameters:
	Id = '02'
	MenuEntry = "NextActionIndic"
	Offset = 0
	Length = menuEntry.length
	NextAction = '10' (SETUP CALL)
	HelpSupported = true
	IconQualifier = 0
	IconIdentifier = 0
		1- No exception shall be thrown.
		2- Shall return true.
		3- Shall return false.
			The SIM shall issue a SETUP MENU proactive command which contains an Items Next Action Indicator list and which contains a command qualifier '80'.

Id	Description	API Expectation	APDU Expectation
5	**Checking applet is triggered by a MENU_SELECTION_HELP_REQUEST** `Send ENVELOPE(MENU_SELECTION_HELP_REQUEST) with Item Identifier = '02'`	Applet is trigged by a MENU_SELECTION_HELP_REQUEST and the Item Identifier is 02	

6	**help supported=true**
	`1- changeMenuEntry()with parameters:`
	`Id = '01'`
	`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.
			The SIM shall issue a SETUP MENU proactive command which contains a command qualifier '80'.

7	**Checking applet is triggered by a MENU_SELECTION_HELP_REQUEST** `Send ENVELOPE(MENU_SELECTION_HELP_REQUEST) with Item Identifier = '01'`	Applet is trigged by a MENU_SELECTION_HELP_REQUEST and the Item Identifier is 01	

8	**Setting icons, help supported = false**
	`1- changeMenuEntry() for entries '01','02', with parameters:`
	`Id = '01'/'02'`
	`MenuEntry = "IconQualifier"`
	`Offset = 0`
	`Length = menuEntry.length`
	`NextAction = 0`
	`HelpSupported = false`
	`IconQualifier = '01'`
	`IconIdentifier = '02' / '01'`
	`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.
			The SIM shall issue a SETUP MENU proactive command which contains an Icon Identifier List.

9	**MenuEntry is disabled**
	`1- disableMenuEntry('01').`
	`2- changeMenuEntry()with parameters:`
	`Id = '01'`
	`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.
			The SIM shall issue a SETUP MENU proactive command which contains the entry. Without Icon identifier List Simple TLV

Id	Description	API Expectation	APDU Expectation

10	**MenuEntry is null**
	`changeMenuEntry()with:`
	`MenuEntry = NULL`
		Shall throw java.lang.NullPointerException.	

11	**Offset causes access outside array bounds**
	`Id = '01'`
	`MenuEntry = "Violation"`
	`Offset = menuEntry.length +1`
	`Length = 0`
	`NextAction = 0`
	`HelpSupported = false`
	`IconQualifier = 0`
	`IconIdentifier = 0`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

12	**Big Offset causes access outside array bounds**
	`Id = '01'`
	`MenuEntry = "Violation"`
	`Offset = 255`
	`Length = 1`
	`NextAction = 0`
	`HelpSupported = false`
	`IconQualifier = 0`
	`IconIdentifier = 0`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

13	**Offset < 0 causes access outside array bounds**
	`Id = '01'`
	`MenuEntry = "Violation"`
	`Offset = -1`
	`Length = 1`
	`NextAction = 0`
	`HelpSupported = false`
	`IconQualifier = 0`
	`IconIdentifier = 0`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

14	**Length causes access outside array bounds**
	`Id = '01'`
	`MenuEntry = "Violation"`
	`Offset = 0`
	`Length = MenuEntry.length + 1`
	`NextAction = 0`
	`HelpSupported = false`
	`IconQualifier = 0`
	`IconIdentifier = 0.`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

15	**Length < 0 causes access outside array bounds**
	`Id = '01'`
	`MenuEntry = "Violation"`
	`Offset = 0`
	`Length = -1`
	`NextAction = 0`
	`HelpSupported = false`
	`IconQualifier = 0`
	`IconIdentifier = 0.`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

16	**Both offset and length causes access outside array bounds**
	`Id = '01'`
	`MenuEntry = "Violation"`
	`Offset ∊ [1, MenuEntry.length]`
	`Length = MenuEntry.length`
	`NextAction = 1`
	`HelpSupported = false`
	`IconQualifier = 0`
	`IconIdentifier = 0`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.
		.	

Id	Description	API Expectation	APDU Expectation

17	**Invalid ID used**
	`Id = '00'`
	`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.	

18	**ID isn't allocated to a menu entry of this applet instance**
	`Id = '0A'`
	`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.	

19	**The text is bigger than the allocated space**
	`Id = '02'`
	`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.	

20	**With a smaller text length than the initial length**
	`1. changeMenuEntry()with parameters:`
	`Id = '02'`
	`MenuEntry = "Init"`
	`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)`
		1. No exception shall be thrown.
		2. Shall return true.
		3. Shall return false.
			The SIM shall issue a SETUP MENU proactive command which contains the new text for entry ID '02'.

6.2.9.2.4 Test Coverage

CRR number	Test case number
N1	1, 2, 3, 4, 6, 8, 9, 20
N2	9
N3	1, 2, 3, 4, 6, 8, 9, 20
N4	6
N5	7,5
N6	6
N7	1, 2, 3, 4, 8, 9, 20
N8	Checked in framework tests: FWK_APT_EMSH
N9	8, 9
N10	8
N11	4
P1	10
P2	11, 12, 13
P3	14, 15
P4	16

CRR number	Test case number
C1	17, 18
C2	19

6.2.9.3 Method clearEvent

Test Area Reference: API_2_TKR_CEVTB

6.2.9.3.1 Conformance requirement:

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

```
public void clearEvent(byte event)
                throws ToolkitException,
                       javacard.framework.TransactionException
```

6.2.9.3.1.1 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.

6.2.9.3.1.2 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.

6.2.9.3.1.3 Context errors

- CRRC1: shall throw javacard.framework.TransactionException - if the operation would cause the commit capacity to be exceeded.

6.2.9.3.2 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 3GPP TS 43.019 [7] excepted those that are not allowed or supported by setEvent().

Load Script:	API_2_TKR_CEVTB_1.ldr
Cleanup script:	API_2_TKR_CEVTB_1.clr
Parameter File:	API_2_TKR_CEVTB_1.par

6.2.9.3.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Clear ALLOWED unregistered events** For events ranging from -1, 1 to 24 and 127* excepted those that aren't allowed (7, 8, 11, 19), 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 event (-1, 1 to 24 and 127)* excepted those that aren't allowed (7, 8, 11, 19), the applet calls setEvent() method. 2- For each ALLOWED and SUPPORTED event (-1, 1 to 24 and 127)* excepted those that aren't allowed (7, 8, 11, 19), 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 ToolkitException with reason EVENT_NOT_ALLOWED.	
4	**Checking applet isn't triggered by an ENVELOPE(SMS-PP DOWNLOAD) command** 1 - reset and initialise the card 2 - An ENVELOPE(SMS-PP DOWNLOAD) is sent with a TAR referencing applet.	Applet is not trigged by an ENVELOPE(SMS-PP DOWNLOAD) command	

NOTE: Although the method clearEvent is defined for a range from –128 to 127 only the allowed events are tested here, because the range from -128 to –2 is reserved for propriatary use in TS 43.019 [7] chapter 6.2 and the range from 25 to 126 is omitted for compatibility with future releases of TS 43.019 [7]

6.2.9.3.4 Test Coverage

CRR number	Test case number
N1	1,2
N2	4
N3	Framework
N4	Framework
N5	Framework
N6	Framework
P1	3
P2	3
P3	3
P4	3

C1	not testable

6.2.9.4 Method disableMenuEntry

Test Area Reference: API_2_TKR_DMETB

6.2.9.4.1 Conformance requirement:

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

```
public void disableMenuEntry(byte id)
                      throws ToolkitException
```

6.2.9.4.1.1 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 shall dynamically update the menu stored in the ME .
- 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.

6.2.9.4.1.2 Parameters error

No requirements.

6.2.9.4.1.3 Context errors

- CRRC1: shall throw a ToolkitException with reason = ENTRY_NOT_FOUND if the menu entry doesn't exist for this applet

6.2.9.4.2 Test suite files

Additional requirements for the GSM personalization:

- content of EF sume shall be:
  - Title Alpha Identifier: "TOOLKIT TEST"

Test Script: API_2_TKR_DMETB_1.scr

Test Applet: API_2_TKR_DMETB_1.java

- Installation parameter:

  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_DMETB_1.ldr

Cleanup script: API_2_TKR_DMETB_1.clr

Parameter File: API_2_TKR_DMETB_1.par

6.2.9.4.3 Test procedure

Id	Description	API Expectation	APDU Expectation

1	**Check the menu state before disabling a previously enabled entry not registered to EVENT_MENU_SELECTION_HELP_REQUEST**
	`1- reset and initialise the card`
	`2- isEventSet(EVENT_MENU_SELECTION)`
	`3- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)`
		1- Shall return true
		2- Shall return false
			1- The SIM shall issue a SET UP MENU proactive command with entry '01' and '02'.

2	**Check the menu state after disabling a previously enabled entry not registered to EVENT_MENU_SELECTION_HELP_REQUEST**
	`1- disableMenuEntry('01')`
	`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.
			3- The SIM shall issue a SET UP MENU proactive command with entry '02' only.

3	**Check the menu before disabling a previously enabled entry registered to EVENT_MENU_SELECTION_HELP_REQUEST**
	`1- change Menu Entry '02' to indicate help supported`
	`2- isEventSet(EVENT_MENU_SELECTION)`
	`3- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)`
		2- Shall return true
		3- Shall return true
			3- The SIM shall issue a SET UP MENU proactive command with entry '02', indicating help supported.

4	**Check the menu after disabling a previously enabled entry registered to EVENT_MENU_SELECTION_HELP_REQUEST**
	`1- disableMenuEntry('02')`
	`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.
			3- The SIM shall issue a SET UP MENU proactive command with 1st Item TLV with a length of 0.

5	**Disabling invalid entries** `For ID ranging from '00' to 'FF' except '01' and '02', the applet calls disableMenuEntry(ID) method.`	Each time a Toolkit Exception with MENU_ENTRY_NOT_FOUND reason code shall be thrown.	

6.2.9.4.4 Test Coverage

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

6.2.9.5 Method enableMenuEntry

Test Area Reference: API_2_TKR_EMETB

6.2.9.5.1 Conformance requirement:

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

```
public void enableMenuEntry(byte id)
                     throws ToolkitException
```

6.2.9.5.1.1 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 shall dynamically issue a SETUP MENU proactive command which does contain an ITEM SIMPLE TLV object for this entry.

6.2.9.5.1.2 Parameters error

No requirements.

6.2.9.5.1.3 Context errors

- CRRC1: shall throw a ToolkitException with reason = MENU_ENTRY_NOT_FOUND if the menu entry doesn't exist for this applet

6.2.9.5.2 Test suite files

Additional requirements for the GSM personalization:

- content of EF sume shall be:
  - Title Alpha Identifier: "TOOLKIT TEST"

Test Script:	API_2_TKR_EMETB_1.scr
Test Applet:	API_2_TKR_EMETB_1.java

- Installation parameter:

  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_EMETB_1.ldr
Cleanup script:	API_2_TKR_EMETB_1.clr
Parameter File:	API_2_TKR_EMETB_1.par

6.2.9.5.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Check menu state before enabling a previously disabled entry not registered to EVENT_MENU_SELECTION_HELP_REQUEST** 1- isEventSet(EVENT_MENU_SELECTION) 2- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST) 3- disableMenuEntry('01')	1- Shall return true 2- Shall return false 3- No exception shall be thrown.	3- The SIM shall issue a SET UP MENU proactive command with entry '02' only.
2	**Check menu state after enabling a previously disabled entry not registered to EVENT_MENU_SELECTION_HELP_REQUEST** 1- enableMenuEntry('01') 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.	3- The SIM shall issue a SET UP MENU proactive command with entry '01' and '02'.
3	**Check menu state before enabling a previously enabled entry registered to EVENT_MENU_SELECTION_HELP_REQUEST** 1- change Menu Entry '02' to indicate help supported 2- isEventSet(EVENT_MENU_SELECTION) 3- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST) 4- disableMenuEntry('02')	2- Shall return true 3- Shall return true 4- No exception shall be thrown	4- The SIM shall issue a SET UP MENU proactive command with entry '01'. The help information available flag.is not verified
4	**Check menu state after enabling a previously enabled entry registered to EVENT_MENU_SELECTION_HELP_REQUEST** 1- enableMenuEntry('02'). 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.	3- The SIM shall issue a SET UP MENU proactive command with entries '01' and '02' indicating help supported.
5	**Enabling invalid entries** For ID ranging from '00' to 'FF' except '01' and '02', the applet calls enableMenuEntry(ID) method.	Each time a Toolkit Exception with MENU_ENTRY_NOT_FOUND reason code shall be thrown.	

6.2.9.5.4 Test Coverage

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

6.2.9.6 Method getEntry

Test Area Reference: API_2_TKR_GETY

6.2.9.6.1 Conformance requirement:

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

```
public static ToolkitRegistry getEntry()
                                throws ToolkitException
```

6.2.9.6.1.1 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.

6.2.9.6.1.2 Parameters error

No requirements.

6.2.9.6.1.3 Context errors

No requirements.

6.2.9.6.2 Test suite files

Test Script:	API_2_TKR_GETY_1.scr
Test Applet:	API_2_TKR_GETY_1.java
Load Script:	API_2_TKR_GETY_1.ldr
Cleanup script:	API_2_TKR_GETY_1.clr
Parameter File:	API_2_TKR_GETY_1.par

6.2.9.6.3 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 ToolkitRegistry instance as for test case 1.	

6.2.9.6.4 Test Coverage

CRR number	Test case number
N1	1
N2	2

6.2.9.7 Method getPollInterval

Test Area Reference: API_2_TKR_GPOL

6.2.9.7.1 Conformance requirement:

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

```
public short getPollInterval()
```

6.2.9.7.1.1 Normal execution

- CRRN1: shall return a value between 1 and 15300 if applet is registered to EVENT_STATUS_COMMAND event.
- CRRN2: shall return POLL_NO_DURATION value (0) if the toolkit applet is not registered to EVENT_STATUS_COMMAND event.

6.2.9.7.1.2 Parameters error

No requirements.

6.2.9.7.1.3 Context errors

No requirements.

6.2.9.7.2 Test suite files

Test Script:	API_2_TKR_GPOL_1.scr
Test Applet:	API_2_TKR_GPOL_1.java
Load Script:	API_2_TKR_GPOL_1.ldr
Cleanup script:	API_2_TKR_GPOL_1.clr
Parameter File:	API_2_TKR_GPOL_1.par

6.2.9.7.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Applet isn't registered to EVENT_STATUS_COMMAND** `getPollInterval().`	Shall return 0.	
2	**Requesting max duration** 1- `requestPollInterval(15300)` 2- Reset and initialise the card 3- `getPollInterval()`	1- No exception shall be thrown. 3- Shall return a value between 1 and 15300.	
3	**Requesting System Duration** 1- `requestPollInterval(POLL_SYSTEM_DURATION)` 2- Reset and initialise the card 3- `getPollInterval().`	1- No exception shall be thrown. 3- Shall return a value between 1 and 15300.	
4	**Requesting no Duration** 1- `requestPollInterval(POLL_NO_DURATION)` 2- Reset and initialise the card 3- `getPollInterval().`	1- No exception shall be thrown. 3- Shall return 0.	

6.2.9.7.4 Test Coverage

CRR number	Test case number
N1	2, 3
N2	1, 4

6.2.9.8 Method initMenuEntry

Test Area Reference: API_2_TKR_IMET_BSSBZBS

6.2.9.8.1 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
```

6.2.9.8.1.1 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 = '80'
- 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 '00', 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.
- CRRN10: After the completion of the SETUP MENU command, if an ENVELOPE (MENU_SELECTION) command is received by the SIM for this identifier, then the SIM Toolkit framework shall trigger the applet.

6.2.9.8.1.2 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

6.2.9.8.1.3 Context errors

- 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)

6.2.9.8.2 Test suite files

Additional requirements for the GSM personalization:

- content of EF sume shall be:
  - Title Alpha Identifier: "TOOLKIT TEST"
  - 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:

  Same as default applet but with:

  - Maximum text length for a menu entry: 15
  - Maximum number of menu entries: 6
  - Position / Identifier for each menu entry: '01'/'01', '02'/'02', '03'/'03', '04'/'04', '05'/'05', and '06'/'06'

Load Script: API_2_TKR_IMET_BSSBZBS_1.ldr

Cleanup script: API_2_TKR_IMET_BSSBZBS_1.clr

Parameter File: API_2_TKR_IMET_BSSBZBS_1.par

6.2.9.8.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**NULL as parameter to menuEntry** `MenuEntry = NULL`	Shall throw a java.lang.NullPointerException.	

2	**Offset > menuEntry.length**
	`MenuEntry = "ToolkitTest"`
	`Offset = 12`
	`Length = 0`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

Id	Description	API Expectation	APDU Expectation

3	**Offset < 0**
	`MenuEntry = "ToolkitTest"`
	`Offset = -1`
	`Length = 11`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

4	**Offset = 255**
	`MenuEntry = "ToolkitTest"`
	`Offset = 255`
	`Length = 11`
		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	**Offset + length > menuEntry.length**
	`MenuEntry = "ToolkitTest"`
	`Offset = 11`
	`Length = 1`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

8	**MenuEntry.length > size allocated at loading for each menu entry**
	`MenuEntry = "ToolkitTest impossible"`
	`Offset = 0`
	`Length = 16`
		ALLOWED_LENGTH_EXCEEDED ToolkitException is thrown.	

9	**Successful call,**
	menuEntry is the whole buffer
	`1- initMenuEntry()`
	`MenuEntry = "TOOLKIT TEST 1"`
	`Offset = 0`
	`Length = 14`
	`NextAction = '00'`
	`HelpSupported = false`
	`IconQualifier = '00'`
	`IconIdentifier = 0`
	`2- isEventSet(EVENT_MENU_SELECTION)`
		1- No exception shall be thrown, Shall return ID '01'.
		2- Shall return true.	

10	**Successful call,**
	menuEntry part of a buffer
	`1- initMenuEntry()`
	`MenuEntry = "1234567TOOLKIT TEST 2"`
	`Offset = 7`
	`Length = 14`
	`NextAction = '00'`
	`HelpSupported = false`
	`IconQualifier = '00'`
	`IconIdentifier = 0`
	`2- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)`
		1- No exception shall be thrown,Shall return ID '02'.
		2- Shall return false.	

Id	Description	API Expectation	APDU Expectation

11	**Successful call,
	menuEntry with help supported**
	1- initMenuEntry()
	MenuEntry = "TOOLKIT TEST 3"
	Offset = 0
	Length = 14
	NextAction = '00'
	HelpSupported = true
	IconQualifier = '00'
	IconIdentifier = 0
	2- isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)
		1- No exception shall be thrown, Shall return ID '03'
		2- Shall return true.	

12	**Successful call,
	menuEntry with an Icon**
	MenuEntry = "TOOLKIT TEST 4"
	Offset = 0
	Length = 14
	NextAction = '00'
	HelpSupported = false
	IconQualifier = '01' [icon not self explanatory]
	IconIdentifier = 1
		1- No exception shall be thrown.
		2- Shall return ID '04'	

13	**Successful call,
	menuEntry with a next action indication**
	MenuEntry = "TOOLKIT TEST 5"
	Offset = 0
	Length = 14
	NextAction = '24' [Select Item]
	HelpSupported = false
	IconQualifier = '00'
	IconIdentifier = 0
		1- No exception shall be thrown.
		2- Shall return ID '05'	

14	**Successful call,
	length = 0**
	initMenuEntry()
	MenuEntry = "ToolkitTest"
	Offset = 0
	Length = 0
	NextAction = '00'
	HelpSupported = false
	IconQualifier = '00'
	IconIdentifier = 0
		No exception shall be thrown,Shall return ID '06'.	

15	**Initialise more entry than allocated at loading**
	MenuEntry = "ToolkitTest"
	Offset = 0
	Length = 11
		REGISTRY_ERROR ToolkitException is thrown.	

Id	Description	API Expectation	APDU Expectation
16	**Dynamic update of the menu stored by the** ME 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" [ItemId=6] = "" [ItemsNextAction]=06000000002400
17	**Check Applet is triggered by ENVELOPE(MENU_SELECTION) command** Menu Entry ID = '01'	Applet is trigged by an ENVELOPE(MENU_SELECTION) command & Menu Entry ID = '01'	
18	**Check Applet is triggered by ENVELOPE(MENU_SELECTION) command** Menu Entry ID = '02'	Applet is trigged by an ENVELOPE(MENU_SELECTION) command & Menu Entry ID = '02'	
19	**Check Applet is triggered by ENVELOPE(MENU_SELECTION) command** Menu Entry ID = '03'	Applet is trigged by an ENVELOPE(MENU_SELECTION) command & Menu Entry ID = '03'	
20	**Check Applet is triggered by ENVELOPE(MENU_SELECTION) command** Menu Entry ID = '04'	Applet is trigged by an ENVELOPE(MENU_SELECTION) command & Menu Entry ID = '04'	
21	**Check Applet is triggered by ENVELOPE(MENU_SELECTION) command** Menu Entry ID = '05'	Applet is trigged by an ENVELOPE(MENU_SELECTION) command & Menu Entry ID = '05'	
22	Check Applet is triggered by ENVELOPE (MENU_SELECTION_HELP_REQUEST) command Menu Entry ID = '03'	Applet is trigged by an ENVELOPE(MENU_SELECTION_HELP_REQUEST) command & Menu Entry ID = '03'	
23	**Check Applet is triggered by ENVELOPE(MENU_SELECTION) command** Menu Entry ID = '06'	Applet is trigged by an ENVELOPE(MENU_SELECTION) command & Menu Entry ID = '06'	

6.2.9.8.4 Test Coverage

CRR number	Test case number
N1	16
N2	9
N3	11

CRR number	Test case number
N4	22
N5	11, 16
N6	10
N7	12,16
N8	12,16
N9	13,16
N10	9, 10, 11, 12, 13, 14, 17, 18, 19, 20, 21, 23
P1	1
P2	2, 3, 4
P3	5, 6
P4	7
C1	8
C2	14

6.2.9.9 Method isEventSet

Test Area Reference: API_2_TKR_IEVSB

6.2.9.9.1 Conformance requirement:

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

```
public boolean isEventSet(byte event)
```

6.2.9.9.1.1 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.

6.2.9.9.1.2 Parameters error

No requirements.

6.2.9.9.1.3 Context errors

No requirements.

6.2.9.9.2 Test suite files

Test Script: API_2_TKR_IEVSB_1.scr

Test Applet: API_2_TKR_IEVSB_1.java

API_2_TKR_IEVSB_2.java

- Installation parameter:

  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'
  - Maximum number of timers: 1

Load Script: API_2_TKR_IEVSB_1.ldr

Cleanup script: API_2_TKR_IEVSB_1.clr

Parameter File: API_2_TKR_IEVSB_1.par

6.2.9.9.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Install Applet1 only registered to EVENT FORMATTED_SMS_PP_ENV and EVENT_MENU_SELECTION** **Test that events aren't set** `Applet calls isEventSet() for each event ranging from –1, 1 to 24 and 127* excepted EVENT_FORMATTED_SMS_PP_ENV (2) and EVENT_MENU_SELECTION (7).`	Shall return false each time.	
2	**For EVENT_FORMATTED_SMS_PP_ENV** `isEventSet(EVENT_FORMATTED_SMS_PP_ENV)`	Shall return true.	
3	**For EVENT_MENU_SELECTION** `isEventSet(EVENT_MENU_SELECTION)`	Shall return true	

4	**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.	

5	**Setting events**
	`For all allowed events defined in TS 43.019[7] for method setEvent(): EVENT_PROFILE_DOWNLOAD, EVENT_FORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_PP_UPD, EVENT_FORMATTED_SMS_CB, EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_CB, EVENT_CALL_CONTROL_BY_SIM, EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM, EVENT_EVENT_DOWNLOAD_MT_CALL, EVENT_EVENT_DOWNLOAD_CALL_CONNECTED, EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED, EVENT_EVENT_DOWNLOAD_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_USER_ACTIVITY, EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE , EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS, EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION, EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION, EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS, EVENT_FIRST_COMMAND_AFTER_SELECT, EVENT_UNRECOGNIZED_ENVELOPE`
	`applet calls:`
	`1- setEvent() method`
	`2- isEventSet() method`
		1- No exception shall be thrown.
		2- Shall return true each time.	

6	**For EVENT_MENU_SELECTION_HELP_ REQUEST**
	`1- isEventSet(EVENT_MENU_SELECTION_HELP_ REQUEST)`
	`2- call changeMenuEntry() with help supported`
	`3- isEventSet(EVENT_MENU_SELECTION_HELP_ REQUEST)`
		1- Shall return false.
		Shall return true.	

7	**For EVENT_TIMER_EXPIRATION**	1- Shall return false.	

	1- isEventSet(EVENT_TIMER_EXPIRATION) 2- call allocateTimer() 3- isEventSet(EVENT_TIMER_EXPIRATION)	3- Shall return true.	
8	**For EVENT_STATUS_COMMAND** 1- isEventSet(EVENT_STATUS_COMMAND) 2- call requestPollInterval(POLL_SYSTEM_DURATION) 3- isEventSet(EVENT_STATUS_COMMAND)	1- Shall return false. 3- Shall return true.	
9	**Install Applet2 only registered to EVENT FORMATTED_SMS_PP_ENV** isEventSet(EVENT_MENU_SELECTION)	Shall return false.	

NOTE: Although the method isEventSet() is defined for a range from –128 to 127 only the allowed events are tested, because the range from -128 to –2 is reserved for propriatary use in TS 43.019 [7] chapter 6.2 and the range from 25 to 126 is omitted for compatibility with future releases of TS 43.019 [7]

6.2.9.9.4 Test Coverage

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

6.2.9.10 Method releaseTimer

Test Area Reference: API_2_TKR_RTIMB

6.2.9.10.1 Conformance requirement:

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

```
public void releaseTimer(byte timerIdentifier)
                  throws ToolkitException
```

6.2.9.10.1.1 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: After invocation of the method the indicated timer shall be released and available for reallocation.
- CRRN4: The applet is deregistered of the EVENT_TIMER_EXPIRATION for the indicated Timer Identifier.

6.2.9.10.1.2 Parameters error

- CRRP1: shall throw a ToolkitException with INVALID_TIMER_ID reason if the timer identifier isn't between 1 and 8.

6.2.9.10.1.3 Context errors

- CRRC1: shall throw a ToolkitException with INVALID_TIMER_ID reason if the timer is valid but isn't allocated to this applet.

6.2.9.10.2 Test suite files

Test Script: API_2_TKR_RTIMB_1.scr

Test Applet: API_2_TKR_RTIMB_1.java

- Installation parameter:
  - As Default, except max timer which is set to 8.

Load Script: API_2_TKR_RTIMB_1.ldr

Cleanup script: API_2_TKR_RTIMB_1.clr

Parameter File: API_2_TKR_RTIMB_1.par

6.2.9.10.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Releasing not allocated timers** For each timer ID ranging from '00' to 'FF', applet calls releaseTimer(ID).	Each time, method shall throw a ToolkitException with reason code INVALID_TIMER_ID.	
2	**Releasing allocated timers** 1- 8 * allocateTimer() . 2- 7 * releaseTimer(id). 3- isEventSet(EVENT_TIMER_EXPIRATION)	1- No exception shall be thrown. 2- Each time, no exception shall be thrown. 3- Shall return true	
3	**Releasing invalid timer ID** 1- releaseTimer('FF') 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(last timer allocated) 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 8 * 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 Send ENVELOPE(TIMER_EXPIRATION)	Applet is not trigged by an ENVELOPE(TIMER_EXPIRATION) command	

6.2.9.10.4 Test Coverage

CRR number	Test case number
N1	4
N2	2, 3
N3	5, 6
N4	7
P1	1, 3
C1	Framework

6.2.9.11 Method requestPollInterval

Test Area Reference: API_2_TKR_RPOLS

6.2.9.11.1 Conformance requirement:

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

```
public void requestPollInterval(short duration)
                         throws ToolkitException
```

6.2.9.11.1.1 Normal execution

- CRRN1: If duration is between 1 and 15300 or equal to POLL_SYSTEM_DURATION, the applet registers to EVENT_STATUS_COMMAND.
- CRRN2: If duration is POLL_NO_DURATION, the applet is deregistered from EVENT_STATUS_COMMAND.

6.2.9.11.1.2 Parameters error

- CRRP1: the method should throw a ToolkitException with REGISTRY_ERROR reason if duration is > 15300 or is < -1 (POLL_SYSTEM_DURATION).

6.2.9.11.1.3 Context errors

No requirements.

6.2.9.11.2 Test suite files

Test Script:	API_2_TKR_RPOLS_1.scr
Test Applet:	API_2_TKR_RPOLS_1.java
Load Script:	API_2_TKR_RPOLS_1.ldr
Cleanup script:	API_2_TKR_RPOLS_1.clr
Parameter File:	API_2_TKR_RPOLS_1.par

6.2.9.11.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Requesting a value between 1 and 15300 s** 1- isEventSet(EVENT_STATUS_COMMMAND) 2- requestPollInterval(duration) for boundaries values: 1, 255, 256, 15300. 3- isEventSet(EVENT_STATUS_COMMAND).	1- Shall return false. 2- No exception shall be thrown. 3- Shall return true.	
2	**Check Applet is triggered by a STATUS command** 1- reset and card initialization 2- Send STATUS command	2- Applet is trigged by a STATUS command	
3	**Requesting POLL SYSTEM DURATION** 1- isEventSet(EVENT_STATUS_COMMMAND). 2- RequestPollInterval(POLL_SYSTEM_DURATION).	1- Shall return true. 2- No exception shall be thrown. 3- Shall return true.	

	`3- IsEventSet(EVENT_STATUS_COMMAND).`		

4	**Check Applet is triggered by a STATUS command**
	`1- reset and card initialization`
	`2- Send STATUS command`
		2- Applet is trigged by a STATUS command	

5	**Requesting invalid duration** requestPollInterval(duration) for following values: 15301, 32767, -2, -32768	Each time, a ToolkitException with REGISTRY_ERROR reason code, shall be thrown.	

6	**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.	

7	**Check Applet isn't triggered by an STATUS command.**
	`1- reset and card initialization`
	`2- Send STATUS command`
		2- Applet is not trigged by a STATUS command	

6.2.9.11.4 Test Coverage

CRR number	Test case number
N1	1, 2, 3, 4
N2	6, 7
P1	5

6.2.9.12 Method setEvent

Test Area Reference: API_2_TKR_SEVTB

6.2.9.12.1 Conformance Requirement:

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

```
public void setEvent(byte id)
              throws ToolkitException,
                     javacard.framework.TransactionException
```

6.2.9.12.1.1 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: the method shall accept all the events defined in 3GPP TS 43.019 [7] except: EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION , EVENT_STATUS_COMMAND
- CRRN4: no exception shall be thrown if the applet registers more than once to the same event.
- CRRN5: all updates in the ToolkitRegistry are atomic.

6.2.9.12.1.2 Parameters error

- CRRP1: shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason if event is 0.

- 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.

6.2.9.12.1.3 Context errors

- 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.
- CRRC3: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_PP_ENV and the applet has no TAR defined.
- CRRC4: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_PP_UPD and the applet has no TAR defined.
- CRRC5: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_CB_ENV and the applet has no TAR defined.
- CRRC6: shall throw javacard.framework.TransactionException - if the operation would cause the commit capacity to be exceeded.

6.2.9.12.2 Test suite files

Test Script:	API_2_TKR_SEVTB_1.scr
Test Applet:	API_2_TKR_SEVTB_1.java
	API_2_TKR_SEVTB_2.java
	API_2_TKR_SEVTB_3.java
	API_2_TKR_SEVTB_4.java
Load Script:	API_2_TKR_SEVTB_1.ldr
	The load script installs the 4 instances.
Cleanup script:	API_2_TKR_SEVTB_1.clr
Parameter File:	API_2_TKR_SEVTB_1.par

6.2.9.12.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Applet1 is triggered by ENVELOPE(SMS_ PP_FORMATTED) command.** `Send ENVELOPE(SMS_PP_FORMATTED)`	Applet1 shall be triggered	

2	**Setting ALLOWED and SUPPORTED events**
	`1- For all allowed events (-1, 1 to 24 and 127 excepted 7, 8, 11, 19) defined in TS 43.019 [7]*:`
	`EVENT_PROFILE_DOWNLOAD,`		

Id	Description	API Expectation	APDU Expectation
	EVENT_FORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_PP_UPD, EVENT_FORMATTED_SMS_CB, EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_CB, EVENT_CALL_CONTROL_BY_SIM, EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM, EVENT_EVENT_DOWNLOAD_MT_CALL, EVENT_EVENT_DOWNLOAD_CALL_CONNECTED, EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED, EVENT_EVENT_DOWNLOAD_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_USER_ACTIVITY, EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE, EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS, EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION, EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION, EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS, EVENT_FIRST_COMMAND_AFTER_SELECT, EVENT_UNRECOGNIZED_ENVELOPE 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** Call setEvent(0)	Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason code.	
4	**Setting EVENT_MENU_SELECTION** Call setEvent(EVENT_MENU_SELECTION)	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
5	**Setting EVENT_MENU_SELECTION_HELP_REQUEST** Call setEvent(EVENT_MENU_SELECTION_HELP_REQUEST)	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
6	**Setting EVENT_TIMER_EXPIRATION** Call setEvent(EVENT_TIMER_EXPIRATION)	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
7	**Setting EVENT_STATUS_COMMAND** Call setEvent(EVENT_STATUS_COMMAND)	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
8	**Setting EVENT_CALL_CONTROL_BY_SIM** Call setEvent(EVENT_CALL_CONTROL_BY_SIM)	No Exception shall be thrown	
9	**Setting EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM** Call setEvent(EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM)	No Exception shall be thrown	
10	**Check applet is triggered by an ENVELOPE(CALL_CONTROL_BY_SIM)** Trigger the applet	Applet is trigged by an ENVELOPE(CALL_CONTROL_BY_SIM)	

Id	Description	API Expectation	APDU Expectation
11	**Check applet is triggered by an ENVELOPE(MO_SHORT_MESSAGE_CONTROL_BY_SIM)** Trigger the Applet	Applet is trigged by an ENVELOPE(MO_SHORT_MESSAGE_CONTROL_BY_SIM)	
12	**Applet2 is triggered by ENVELOPE(SMS_ PP_DOWNLOAD) command.** `Trigger the Applet2`	Applet2 is trigged by an ENVELOPE(SMS_ PP_DOWNLOAD) command	
13	**Applet2 registers to CALL_CONTROL_BY_SIM but it is already assigned** `setEvent(EVENT_CALL_CONTROL_BY_SIM)`	Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	
14	**Applet2 registers to MO_MESSAGE_CONTROL_BY SIM but it is already assigned** `setEvent(EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM)`	Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	
15	**Applet3 with no TAR defined registers to EVENT_UNFORMATTED_SMS_CB** 1- send ENVELOPE(CELL_BROADCAST_DATA_ DOWNLOAD) 2- setEvent(FORMATTED_SMS_PP_ENV) 3- setEvent(FORMATTED_SMS_PP_UPD) 4- setEvent(FORMATTED_SMS_CB_ENV)	1- Applet3 shall be triggered 2- ToolkitException with reason code TAR_NOT_DEFINED should be thrown 3- ToolkitException with reason code TAR_NOT_DEFINED should be thrown ToolkitException with reason code TAR_NOT_DEFINED should be thrown	
16	**Applet4 registers multiple to EVENT_FORMATTED_SMS_PP_ENV** 1- send ENVELOPE(EVENT_FORMATTED_ SMS_PP_ENV) 2- setEvent(EVENT_FORMATTED_SMS_PP_ UPD) 3- setEvent(EVENT_FORMATTED_SMS_PP_ UPD) 4- send ENVELOPE(EVENT_FORMATTED_ SMS_PP_UPD)	1- Applet4 shall be triggered 2- no Exception shall be thrown 3- no Exception shall be thrown 4- Applet4 shall be triggered	

NOTE: Although the method setEvent is defined for a range from –128 to 127 only the allowed events are tested, because the range from -128 to –2 is reserved for propriatary use in TS TS 43.019 [7] chapter 6.2 and the range from 25 to 126 is omitted for compatibility with future releases of TS 43.019 [7]

6.2.9.12.4 Test Coverage

CRR number	Test case number
N1	2
N2	1,8,9,10,11,12
N3	2,4,5,6,7
N4	16
N5	not testable
P1	3
P2	4
P3	5
P4	6
P5	7
C1	13

C2	14
C3	15
C4	15
C5	15
C6	not testable

6.2.9.13 Method setEventList

Test Area Reference: API_2_TKR_SEVL_BSS

6.2.9.13.1 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,
                         javacard.framework.TransactionException
```

6.2.9.13.1.1 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 3GPP TS 43.019 [7] except: EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION , EVENT_STATUS_COMMAND.
- CRRN4: all updates on the ToolkitRegistry are atomic
- CRRN5: No exception shall be thrown if the applet registers more than once to the same event.

6.2.9.13.1.2 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.
- 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.

6.2.9.13.1.3 Context errors

- 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.
- CRRC3: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_PP_ENV and the applet has no TAR defined.
- CRRC4: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_PP_UPD and the applet has no TAR defined.
- CRRC5: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_CB_ENV and the applet has no TAR defined.
- CRRC6: shall throw javacard.framework.TransactionException - if the operation would cause the commit capacity to be exceeded.

6.2.9.13.2 Test suite files

Test Script:	API_2_TKR_SEVL_BSS_1.scr
Test Applet:	API_2_TKR_SEVL_BSS_1.java
	API_2_TKR_SEVL_BSS_2.java
	API_2_TKR_SEVL_BSS_3.java
Load Script:	API_2_TKR_SEVL_BSS_1.ldr
	The load script installs the 4 instances.
Cleanup script:	API_2_TKR_SEVL_BSS_1.clr
Parameter File:	API_2_TKR_SEVL_BSS_1.par

6.2.9.13.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Applet1 Registering all eventList buffer** EventList = all allowed events (-1, 1 to 24 and 127 excepted 7, 8, 11, 19) defined in TS 43.019[7]: EVENT_PROFILE_DOWNLOAD, EVENT_FORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_PP_UPD, EVENT_FORMATTED_SMS_CB, EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_CB, EVENT_CALL_CONTROL_BY_SIM, EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM, EVENT_EVENT_DOWNLOAD_MT_CALL, EVENT_EVENT_DOWNLOAD_CALL_CONNECTED, EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED, EVENT_EVENT_DOWNLOAD_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_USER_ACTIVITY, EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE, EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS, EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION, EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION, EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS, EVENT_FIRST_COMMAND_AFTER_SELECT, EVENT_UNRECOGNIZED_ENVELOPE 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** EventList = all allowed events defined in TS 43.019[7] (see test case 1). 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)	1- No exception shall be thrown. 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	

Id	Description	API Expectation	APDU Expectation

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** `Call setEventList(eventList) with eventList indicating event 0`	Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason code.	
12	**EVENT_MENU_SELECTION** `Call setEventList(eventList) with eventList indicating EVENT_MENU_SELECTION`	Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	
13	**EVENT_MENU_SELECTION_HELP_REQUEST** `Call setEventList(eventList) with eventList indicating EVENT_MENU_SELECTION_HELP_REQUEST`	Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	
14	**EVENT_TIMER_EXPIRATION** `Call setEventList(eventList) with eventList indicating EVENT_TIMER_EXPIRATION`	Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	
15	**EVENT_STATUS_COMMAND** `Call setEventList(eventList) with eventList indicating EVENT_STATUS_COMMAND`	Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	
16	**Setting EVENT_CALL_CONTROL_BY_SIM** `setEventList(List, 0, 2) with List containing EVENT_CALL_CONTROL_BY_SIM & EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM`	Shall not throw an exception	

17	**Check applet is triggered by an ENVELOPE(CALL_CONTROL_BY_SIM)**
	`Reset and initialise the card`
	`Trigger the applet`
		Applet is trigged by an ENVELOPE(CALL_CONTROL_BY_SIM)	

18	**Check applet is triggered by an ENVELOPE(MO_SHORT_MESSAGE_CONTROL_BY_SIM)** `Trigger the applet`	Applet is trigged by an ENVELOPE(MO_SHORT_MESSAGE_CONTROL_BY_SIM)	

Id	Description	API Expectation	APDU Expectation
19	**Applet2 registers to CALL_CONTROL_BY_SIM but it is already assigned** `setEventList(MonoEventList,0,1) with MonoEventList containing EVENT_CALL_CONTROL_BY_SIM`	Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	
20	**Applet2 registers to MO_SHORT_MESSAGE_CONTROL_BY_SIM but it is already assigned setEventList(MonoEventList,0,1) with MonoEventList containing EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM**	Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	

21	**Applet3 with no TAR defined registers to EVENT_UNFORMATTED_SMS_CB**
	`1- send ENVELOPE(EVENT_UNFORMATTED_SMS_CB)`
	`2- setEventList(EVENT_FORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_ENV)`
	`3- setEventList(EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_PP_ENV)`
	`4- setEventList(EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_CB_ENV)`
	`5- isEventSet(EVENT_UNFORMATTED_SMS_PP_ENV)`
	`6- isEventSet(EVENT_UNFORMATTED_SMS_PP_UPD)`
	`7- isEventSet(EVENT_FORMATTED_SMS_PP_ENV)`
	`8- isEventSet(EVENT_FORMATTED_SMS_PP_UPD)`
	`9- isEventSet(EVENT_FORMATTED_SMS_CB_ENV)`
		1- Applet3 shall be triggered
		2- ToolkitException with reason code TAR_NOT_DEFINED should be thrown
		3- ToolkitException with reason code TAR_NOT_DEFINED should be thrown
		4- ToolkitException with reason code TAR_NOT_DEFINED should be thrown
		5- method should return FALSE
		6- method should return FALSE
		7- method should return FALSE
		8- method should return FALSE
		9- method should return FALSE	

22	`1- setEventList(EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_ENV)`
	`2- isEventSet(EVENT_UNFORMATTED_SMS_PP_ENV)`
		1- no exception should be thrown
		2- method should return true	

6.2.9.13.4 Test Coverage

CRR number	Test case number
N1	1,2
N2	16,17,18
N3	1,2,11,12,13,14,15
N4	21
N5	22
P1	3
P2	4,5,6
P3	7,8,9
P4	10
P5	11
P6	12

P7	13
P8	14
P9	15
C1	19
C2	20
C3	21
C4	21
C5	21
C6	not testable

6.2.10 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.

6.2.11 Class ToolkitException

6.2.11.1 Exception Constants

Test Area Reference: API_2_TKE_CONS

6.2.11.1.1 Conformance requirement:

There is no API, only constants.

6.2.11.1.1.1 Normal execution

- CRRN1: The Constants of the class ToolkitException shall all have the same name and value defined in the 3GPP TS 43.019 [7].

6.2.11.1.1.2 Parameters error

No requirements.

6.2.11.1.1.3 Context errors

No requirements.

6.2.11.1.2 Test suite files

None.

6.2.11.1.3 Test procedure

The constants in Java are resolved at compilation time, therefore a runtime test is not useful. No test of constants will be performed.

6.2.11.2 Constructor ToolkitException

Test Area Reference: API_2_TKE_COORS

6.2.11.2.1 Conformance requirement:

The constructor with following headershall compliant to its definition in the API.

```
public ToolkitException(short reason)
```

6.2.11.2.1.1 Normal execution

- CRRN1: Construct a ToolkitException instance with the specified reason.

6.2.11.2.1.2 Parameters error

No requirements.

6.2.11.2.1.3 Context errors

No requirements.

6.2.11.2.2 Test suite files

No additional requirements for the GSM personalization:

Test Script:	API_2_TKE_COORS_1.scr
Test Applet:	API_2_TKE_COORS_1.java
Load Script:	API_2_TKE_COORS_1.ldr
Cleanup script:	API_2_TKE_COORS_1.clr
Parameter File:	API_2_TKE_COORS_1.par

6.2.11.2.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**reason = (short) 19**	ToolkitException.getReason() = (short)19	

6.2.11.2.4 Test Coverage

CRR number	Test case number
N1	1

6.2.11.3 Method throwIt

Test Area Reference: API_2_TKE_THITS

6.2.11.3.1 Conformance requirement:

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

```
public static void throwIt(short reason)
        throws  ToolkitException
```

6.2.11.3.1.1 Normal execution

- CRRN1: Throws the JCRE instance of the ToolkitException class with the specified reason.
- CRRN2: extends javacard.framework.CardRuntimeException

6.2.11.3.1.2 Parameters error

No requirements.

6.2.11.3.1.3 Context errors

No requirements.

6.2.11.3.2 Test suite files

No additional requirements for the GSM personalization:

Test Script:	API_2_TKE_THITS_1.scr
Test Applet:	API_2_TKE_THITS_1.java
Load Script:	API_2_TKE_THITS_1.ldr
Cleanup Script:	API_2_TKE_THITS_1.clr
Parameter File:	API_2_TKE_THITS_1.par

6.2.11.3.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Throws the JCRE instance of ToolkitException with the specified reason**	Reason = 0	
2	**Throws the JCRE instance of ToolkitException with the specified reason**	Reason = 1	
3	**Throws the JCRE instance of ToolkitException with the specified reason**	Reason = 15	
4	**ToolkitException extends javacard.framework.CardRuntimeException**	Reason = 0	
5	**ToolkitException extends javacard.framework.CardRuntimeException**	Reason = 1	
6	**ToolkitException extends javacard.framework.CardRuntimeException**	Reason = 15	

6.2.11.3.4 Test Coverage

CRR number	Test case number
N1	1, 2, 3
N2	4, 5, 6

6.3 SIM Toolkit Framework

6.3.1 Minimum Handler Availability

This test area tests the rules that define the minimum requirements for the availability of the system handlers.

6.3.1.1 ProactiveHandler

Test Area Reference: FWK_MHA_PAHD

6.3.1.1.1 Conformance Requirement

6.3.1.1.1.1 Normal Execution

- CRRN1: If a proactive session is not ongoing the ProactiveHandler is available from the invocation to the termination of the processToolkit method for the following events:

  EVENT_FORMATTED_SMS_PP_ENV

  EVENT_UNFORMATTED_SMS_PP_ENV

  EVENT_FORMATTED_SMS_PP_UPD

  EVENT_UNFORMATTED_SMS_PP_UPD

EVENT_FORMATTED_SMS_CB

EVENT_UNFORMATTED_SMS_CB

EVENT_MENU_SELECTION

EVENT_MENU_SELECTION_HELP_REQUEST

EVENT_TIMER_EXPIRATION

EVENT_EVENT_DOWNLOAD_MT_CALL

EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE

EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION

EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION

EVENT_UNRECOGNIZED_ENVELOPE

EVENT_STATUS_COMMAND

EVENT_CALL_CONTROL

EVENT_SMS_MO_CONTROL

EVENT_PROFILE_DOWNLOAD

EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS

6.3.1.1.1.2 Parameters error

No requirements.

6.3.1.1.1.3 Context errors

- CRRC1: The ProactiveHandler and its content are not available for any toolkit applet triggered from the invocation to the termination of their processToolkit method for the following events:

EVENT_FIRST_COMMAND_AFTER_SELECT

6.3.1.1.2 Test Suite Files

Test Script:	FWK_MHA_PAHD_1.scr
Test Applet:	FWK_MHA_PAHD_1.java
	FWK_MHA_PAHD_2.java
Load Script:	FWK_MHA_PAHD_1.ldr
Cleanup Script:	FWK_MHA_PAHD_1.clr
Parameter File:	FWK_MHA_PAHD_1.par

Test Procedure

Id	Description	API /Framework Expectation	APDU Expectation
1	**Applets registration to all events and Proactive Handler availability with EVENT_FIRST_COMMAND_AFTER_SELECT** Applet1 is registered to all events defined in TS 43.019 [7]. Using the methods initMenuEntry() for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION and setEventList() for the rest of the events. Applet2 is registered to all events defined in TS 43.019 [7], EVENT_CALL_CONTROL_BY_SIM and EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM. Using the methods initMenuEntry() for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION and setEventList() for the rest of the events. The priority of applet1 is higher than priority of applet2 1- Select MF 2- Applet1 gets the Proactive Handler. Applet1 is deregistered from EVENT_FIRST_COMMAND_AFTER_SELECT. 3- Applet2 gets the Proactive Handler Applet2 is deregistered to EVENT_FIRST_COMMAND_AFTER_SELECT.	1- Applet1 is triggered by EVENT_FIRST_COMMAND_AFTER_SELECT 2- A Toolkit Exception HANDLER_NOT_AVAILABLE is thrown. Applet1 finalizes Applet2 is triggered by EVENT_FIRST_COMMAND_AFTER_SELECT 3- A Toolkit Exception HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
2	**Proactive Handler availability with EVENT_PROFILE_DOWNLOAD** 1- Terminal Profile command is sent to the SIM without the facility of SET_EVENT_LIST, POLL_INTERVAL,SET UP IDLE MODE TEXT and SET UP MENU. 2- Applet1 gets the Proactive Handler Applet1 is deregistered to EVENT_PROFILE_DOWNLOAD 3- Applet2 gets the Proactive Handler Applet2 is deregistered to EVENT_PROFILE_DOWNLOAD	1- Applet1 is triggered by EVENT_PROFILE_DOWNLOAD 2- No exception is thrown. Applet1 finalizes. Applet2 is triggered by EVENT_PROFILE_DOWNLOAD 3- No exception is thrown	
3	**Proactive Handler availability with EVENT_MENU_SELECTION_HELP_REQUEST** Perform SIM initialization with all the facilities supported 1- Envelope menu selection with help request is sent to the SIM 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes	
4	**Proactive Handler availability with EVENT_MENU_SELECTION** 1- Envelope menu selection is sent to the SIM 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes	
5	**Proactive Handler availability with EVENT_FORMATTED_SMS_PP_ENV** 1- Envelope dataDownLoad formatted is sent to the SIM 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
6	**Proactive Handler availability with EVENT_UNFORMATTED_SMS_PP_ENV** 1- Envelope dataDownLoad unformatted is sent to the SIM 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes Applet2 is triggered 3 No exception is thrown.	
7	**Proactive Handler availability with EVENT_FORMATTED_CELL_BROADCAST** 1- Envelope cell broadcast formatted is sent to the SIM 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2-No exception is thrown Applet1 finalizes	
8	**Proactive Handler availability with EVENT_UNFORMATTED_CELL_BROADCAST** 1- Envelope cell broadcast unformatted is sent to the SIM 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes Applet2 is triggered 3 No exception is thrown	
9	**Proactive Handler availability with EVENT_TIMER_EXPIRATION** 1- Timer Id =1 Envelope Timer Expiration is sent to the SIM 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes	
10	**Proactive Handler availability with EVENT_CALL_CONTROL_BY_SIM** 1- Envelope call control by SIM is sent to the SIM 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown.	

Id	Description	API /Framework Expectation	APDU Expectation
11	**Proactive Handler availability with EVENT_MO_SHORT_MESSAGE_CONTROL** 1- Envelope mo short message control by SIM is sent to the SIM 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown	
12	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_MT_CALL** 1- Envelope event download mt call is sent to the SIM 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes Applet2 is triggered 3-No exception is thrown	
13	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_CALL_CONNECTED** 1- Envelope event download call connected is sent to the SIM 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes Applet2 is triggered 3- No exception is thrown	
14	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED** 1- Envelope event download call disconnected is sent to the SIM 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes Applet2 is triggered 3- No exception is thrown.	

Id	Description	API /Framework Expectation	APDU Expectation
15	**Applets triggering with EVENT_EVENT_LOCATION_STATUS** 1- Envelope event download location status is sent to the SIM 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes Applet2 is triggered 3- No exception is thrown	
16	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_USER_ACTIVITY** 1- Envelope event download user activity is sent to the SIM 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes Applet2 is triggered 3- No exception is thrown	
17	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE** 1- Envelope event download idle screen available is sent to the SIM 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes Applet2 is triggered 3- No exception is thrown	
18	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS** 1- Envelope event download card reader status is sent to the SIM 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes Applet2 is triggered 3- No exception is thrown	

Id	Description	API /Framework Expectation	APDU Expectation
19	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION** 1- Envelope event download language selection is sent to the SIM 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2-No exception is thrown. Applet1 finalizes Applet2 is triggered 3-No exception is thrown	
20	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION** 1- Envelope event download browser termination is sent to the SIM 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2-No exception is thrown. Applet1 finalizes Applet2 is triggered 3-No exception is thrown	
21	**Proactive Handler availability with EVENT_STATUS_COMMAND** 1- Status command is sent to the SIM 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes Applet2 is triggered 3- No exception is thrown.	
22	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE** 1- Applet1 builds a proactive command OPEN CHANNEL proactiveHandler.send() method is called. 2- An Envelope Event Download Data Available is sent to the SIM, with channelId=01. 3- Applet1 gets the Proactive Handler	2-Applet1 is triggered 3-No exception is thrown. Applet1 finalizes	1- OPEN CHANNEL proactive Command is fetched TERMINAL RESPONSE is issued with Channel Id = 01
23	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS** 1- An Envelope Event Download Channel Status is sent to the SIM, with ChannelId=01 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
24	**Proactive Handler availability with UNRECOGNIZED_ENVELOPE** 1- An unrecognized Envelope (BER TLV Tag unrecognized) is sent to the SIM 2- Applet1 gets the Proactive Handler **3-Applet2 gets the Proactive Handler**	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes Applet2 is triggered 3- No exception is thrown	
25	**Proactive Handler availability with EVENT_FORMATTED_SMS_PP_UPD** 1- Update Record EFsms instruction formatted is sent to the SIM 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes	
26	**Proactive Handler availability with EVENT_UNFORMATTED_SMS_PP_UPD** 1- Update Record EFsms instruction unformatted is sent to the SIM 1- Applet1 gets the Proactive Handler 2- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes 3- Applet2 is triggered 4- No exception is thrown.	

6.3.1.1.4 Test Coverage

CRR Number	Test Case Number
CRRN1	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26
CRRC1	1

6.3.1.2 ProactiveResponseHandler

Test Area Reference: FWK_MHA_PRHD

6.3.1.2.1 Conformance Requirement

6.3.1.2.1.1 Normal Execution

- CRRN1: The ProactiveResponseHandler is available after the first call to the ProactiveHandler.send() method to the termination of the processToolkit method for the following events:

  EVENT_FORMATTED_SMS_PP_ENV

  EVENT_UNFORMATTED_SMS_PP_ENV

EVENT_FORMATTED_SMS_PP_UPD

EVENT_UNFORMATTED_SMS_PP_UPD

EVENT_FORMATTED_SMS_CB

EVENT_UNFORMATTED_SMS_CB

EVENT_MENU_SELECTION

EVENT_MENU_SELECTION_HELP_REQUEST

EVENT_TIMER_EXPIRATION

EVENT_EVENT_DOWNLOAD_MT_CALL

EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE

EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION

EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION

EVENT_UNRECOGNIZED_ENVELOPE

EVENT_STATUS_COMMAND

EVENT_CALL_CONTROL

EVENT_SMS_MO_CONTROL

EVENT_PROFILE_DOWNLOAD

EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS

6.3.1.2.1.2 Parameters error

No requirements.

6.3.1.2.1.3 Context errors

- CRRC1: The ProactiveResponseHandler and its content are not available for any toolkit applet triggered from the invocation to the termination of their processToolkit method for the following events:

  EVENT_FIRST_COMMAND_AFTER_SELECT

6.3.1.2.2 Test Suite Files

Test Script:	FWK_MHA_PRHD_1.scr
Test Applet:	FWK_MHA_PRHD_1.java
	FWK_MHA_PRHD_2.java
Load Script:	FWK_MHA_PRHD_1.ldr

Cleanup Script: FWK_MHA_PRHD_1.clr

Parameter File: FWK_MHA_PRHD_1.par

6.3.1.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applets registration to all events and Proactive Response Handler availability with EVENT_PROFILE_DOWNLOAD** Applet1 is registered to all events defined in TS 43.019 [7] except EVENT_FIRST_COMMAND_AFTER_SELECT, Applet2 is registered to all events defined in TS 43.109[7] except EVENT_FIRST_COMMAND_AFTER_SELECT, EVENT_CALL_CONTROL_BY_SIM and EVENT_MO_SMS_CONTROL_BY_SIM. Using the methods initMenuEntry() for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION and setEventList() for the rest of the events. 1-Terminal Profile command is sent to the SIM without the facility of SET_EVENT_LIST, POLL_INTERVAL,SET UP IDLE MODE TEXT and SET UP MENU. 2- Applet1 builds a proactive command DISPLAY TEXT. 3- ProactiveHandler.send() method is called 4- ProactiveResponseHandler.getTheHandler() method is called Applet1 is deregistered to EVENT_PROFILE_DOWNLOAD 5- Applet2 builds a proactive command DISPLAY TEXT. 6- ProactiveHandler.send() method is called 7- ProactiveResponseHandler.getTheHandler() method is called Applet1 is deregistered to EVENT_PROFILE_DOWNLOAD	1-Applet1 is triggered by EVENT_PROFILE_DOWNLOAD No exception is thrown 4- No exception is thrown Applet1 finalizes Applet2 is triggered by EVENT_PROFILE_DOWNLOAD 7- No exception is thrown	3- The proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 6- The proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE

Id	Description	API/Framework Expectation	APDU Expectation
2	**Proactive Response Handler availability with EVENT_MENU_SELECTION_HELP_REQUEST** Perform SIM initialization with all the facilities supported 1-Envelope menu selection with help request is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2- ProactiveHandler.send() method is called 3- ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
3	**Proactive Response Handler availability with EVENT_MENU_SELECTION** 1-Envelope menu selection is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2- ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
4	**Proactive Response Handler availability with EVENT_FORMATTED_SMS_PP_ENV** 1-Envelope dataDownLoad formatted is sent to the SIM Applet builds a proactive command DISPLAY TEXT 2-ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
5	**Proactive Response Handler availability with EVENT_UNFORMATTED_SMS_PP_ENV** 1-Envelope dataDownLoad unformatted is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2- ProactiveHandler.send() method is called 3- ProactiveResponseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY	1- Applet1 is triggered 3- No exception is thrown Applet1 finalizes Applet2 is triggered	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE

Id	Description	API/Framework Expectation	APDU Expectation
	TEXT 4- ProactiveHandler.send() method is called 5- ProactiveResponseHandler.getTheHandler() method is called	5- No exception is thrown	4- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
6	**Proactive Response Handler availability with EVENT_FORMATTED_SMS _CB** 1-Envelope cell broadcast formatted is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2- ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called.	1- Applet1 is triggered 3- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
7	**Proactive Response Handler availability with EVENT_UNFORMATTED_SMS _CB** 1-Envelope call broadcast unformatted is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2- ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called. Applet2 builds a proactive command DISPLAY TEXT 4- ProactiveHandler.send() method is called 5- ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown Applet1 finalizes Applet2 is triggered 5- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 4- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
8	**Proactive Response Handler availability with EVENT_TIMER_EXPIRATION** Timer id=1 1-Envelope Timer Expiration is sent to the SIM Applet builds a proactive command DISPLAY TEXT 2-ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
9	**Proactive Response Handler availability with EVENT_CALL_CONTROL_BY_SIM** 1-Envelope call control by sim is sent to the SIM Applet builds a proactive command DISPLAY TEXT	1- Applet1 is triggered	

Id	Description	API/Framework Expectation	APDU Expectation
	2-ProactiveHandler.send() method is called		2- A proactive command DISPLAY TEXT is fetched
	3-ProactiveResponseHandler.getTheHandler() method is called	3- No exception is thrown	TERMINAL RESPONSE
10	**Proactive Response Handler availability with _ MO_SHORT_MESSAGE_CONTROL_BY_SIM**		
	1-Envelope mo short message control by sim is sent to the SIM	1- Applet1 is triggered	
	Applet builds a proactive command DISPLAY TEXT		
	2-ProactiveHandler.send() method is called		2- A proactive command DISPLAY TEXT is fetched
	3-ProactiveResponseHandler.getTheHandler() method is called	3- No exception is thrown	TERMINAL RESPONSE
11	**Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_MT_CALL**		
	1-Envelope event download mt call is sent to the SIM	1- Applet1 is triggered	
	Applet1 builds a proactive command DISPLAY TEXT		
	2-ProactiveHandler.send() method is called		2- A proactive command DISPLAY TEXT is fetched
	3-ProactiveResponseHandler.getTheHandler() method is called.	3- No exception is thrown	TERMINAL RESPONSE
		Applet1 finalizes Applet2 is triggered	
	Applet2 builds a proactive command DISPLAY TEXT		
	4- ProactiveHandler.send() method is called		4- A proactive command DISPLAY TEXT is fetched
	5-ProactiveResponseHandler.getTheHandler() method is called	5- No exception is thrown	TERMINAL RESPONSE

Id	Description	API/Framework Expectation	APDU Expectation
12	**Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_CALL_CONNECTED** 1-Envelope event download call connected is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2-ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called Applet builds a proactive command DISPLAY TEXT 4- ProactiveHandler.send() method is called 5-ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown Applet1 finalizes Applet2 is triggered 5- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 4- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
13	**Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED** 1-Envelope event download call disconnected is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2-ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4- ProactiveHandler.send() method is called 5-ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown Applet1 finalizes Applet2 is triggered 5- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 4- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
14	**Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_LOCATION_STATUS** 1-Envelope event download location status is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2-ProactiveHandler.send() method is called	1- Applet1 is triggered	2-A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE

Id	Description	API/Framework Expectation	APDU Expectation
	3-ProactiveResponseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4- ProactiveHandler.send() method is called 5- ProactiveResponseHandler.getTheHandler() method is called	3- No exception is thrown Applet1 finalizes Applet2 is triggered 5- No exception is thrown	4- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
15	**Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_USER_ACTIVITY** 1-Envelope event download user activity is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2-ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4- ProactiveHandler.send() method is called 5- ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown Applet1 finalizes Applet2 is triggered 5- No exception is thrown	2-A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 4- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
16	**Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE** 1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2-ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4- ProactiveHandler.send() method is called 5- ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown Applet1 finalizes Applet2 is triggered 5- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 4- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
17	**Proactive Response Handler availability with**		

Id	Description	API/Framework Expectation	APDU Expectation
	EVENT_EVENT_DOWNLOAD_CARD_READER _STATUS 1-Envelope event download card reader status is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2-ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4- ProactiveHandler.send() method is called 5- ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown Applet1 finalizes Applet2 is triggered 5- No exception is thrown	2-A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 4- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
18	**Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_LANGUAGE_ SELECTION** 1-Envelope event download language selection is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2-ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4- ProactiveHandler.send() method is called 5- ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3-No exception is thrown Applet1 finalizes Applet2 is triggered 5-No exception is thrown	2-A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 4-A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE

Id	Description	API/Framework Expectation	APDU Expectation
19	**Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION** 1-Envelope event download Browser termination is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2-ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4- ProactiveHandler.send() method is called 5- ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3-No exception is thrown Applet1 finalizes Applet2 is triggered 5-No exception is thrown	2-A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 4-A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
20	**Proactive Response Handler availability with EVENT_STATUS_COMMAND** 1-Status command is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2-ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4- ProactiveHandler.send() method is called 5- ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown Applet1 finalizes Applet2 is triggered 5- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 4- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
21	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE** 1- Applet1 builds a proactive command OPEN CHANNEL. proactiveHandler.send() method is called 2- An Envelope Event Download Data Available is sent to the SIM, with ChannelId=01. 3-Applet1 builds a proactive command DISPLAY TEXT	1- Applet1 is registered to EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE and EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS 2- Applet1 is triggered	1- OPEN CHANNEL proactive command is fetched TERMINAL RESPONSE is issued with Channel Id = 01

Id	Description	API/Framework Expectation	APDU Expectation
	4- ProactiveHandler.send() method is called 5- ProactiveResponseHandler.getTheHandler() method is called	5- No exception is thrown	4- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE

Id	Description	API/Framework Expectation	APDU Expectation
22	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS** 1-An Envelope Event Download Channel Status is sent to the SIM with ChannelId=01. Applet1 builds a proactive command DISPLAY TEXT 2- ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
23	**Proactive Response Handler availability with UNRECOGNIZED_ENVELOPE** 1-An unrecognized Envelope is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2-ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4- ProactiveHandler.send() method is called 5-ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown Applet1 finalizes Applet2 is triggered 5- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 4- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
24	**Proactive Response Handler availability with EVENT_FORMATTED_SMS_PP_UPD** 1- Update Record EFsms instruction formatted is sent to the SIM Applet builds a proactive command DISPLAY TEXT 2-ProactiveHandler.send() method is called 3-ProactiveResponseHandler.getTheHandler() method is called	1- Applet1 is triggered 3- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
25	**Proactive Response Handler availability with EVENT_UNFORMATTED_SMS_PP_UPD** 1- Update Record EFsms instruction unformatted is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT	1- Applet1 is triggered	

Id	Description	API/Framework Expectation	APDU Expectation
	2- ProactiveHandler.send() method is called 3- ProactiveResponseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4- ProactiveHandler.send() method is called 5- ProactiveResponseHandler.getTheHandler() method is called	3- No exception is thrown Applet1 finalizes Applet2 is triggered 5- No exception is thrown	2- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 4- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE

6.3.1.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25
CRRC1	Not testable

6.3.1.3 EnvelopeHandler

Test Area Reference: FWK_MHA_ENHD

6.3.1.3.1 Conformance Requirement

6.3.1.3.1.1 Normal Execution

- CRRN1: The EnvelopeHandler and its content are available for all toolkit applets triggered from the invocation to the termination of their processToolkit method for the following events:

EVENT_FORMATTED_SMS_PP_ENV

EVENT_UNFORMATTED_SMS_PP_ENV

EVENT_FORMATTED_SMS_PP_UPD

EVENT_UNFORMATTED_SMS_PP_UPD

EVENT_FORMATTED_SMS_CB

EVENT_UNFORMATTED_SMS_CB

EVENT_MENU_SELECTION

EVENT_MENU_SELECTION_HELP_REQUEST

EVENT_TIMER_EXPIRATION

EVENT_EVENT_DOWNLOAD_MT_CALL

EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE

EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION

EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION

EVENT_UNRECOGNIZED_ENVELOPE

EVENT_CALL_CONTROL

EVENT_SMS_MO_CONTROL

EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS

6.3.1.3.1.2 Parameters error

No requirements.

6.3.1.3.1.3 Context Errors

- CRRC1: The EnvelopeHandler and its content are not available for any toolkit applet triggered from the invocation to the termination of their processToolkit method for the following events:

  EVENT_STATUS_COMMAND

  EVENT_PROFILE_DOWNLOAD

  EVENT_FIRST_COMMAND_AFTER_SELECT

6.3.1.3.2 Test Suite Files

Test Script:	FWK_MHA_ENHD_1.scr
Test Applet:	FWK_MHA_ENHD_1.java
	FWK_MHA_ENHD_2.java
Load Script:	FWK_MHA_ENHD_1.ldr
Cleanup Script:	FWK_MHA_ENHD_1.clr
Parameter File:	FWK_MHA_ENHD_1.par

6.3.1.3.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet1 and Applet2 registration and Envelope Handler availability with EVENT_FIRST_COMMAND_AFTER_SELECT** 1.Applet1 is registered to all events defined TS 43.019 [7]. The registration is done using the methods initMenuEntry() for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION and setEventList() for the rest of the events. Applet2 is registered to all events defined TS 43.019 [7] except EVENT_PROFILE_DOWNLOAD, EVENT_CALL_CONTROL_BY_SIM and EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM. The registration is done using the methods initMenuEntry() for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer for EVENT_TIMER_EXPIRATION and setEventList for the rest of the events. 2- Select MF. 3-EnvelopeHandler.getTheHandler() method is called by Applet1 Applet1 is deregistered from EVENT_FIRST_COMMAND_AFTER_SELECT. 4-EnvelopeHandler.getTheHandler() method is called by Applet2 Applet2 is deregistered to EVENT_FIRST_COMMAND_AFTER_SELECT.	1- No exception is thrown 2- Applet1 is triggered by EVENT_FIRST_COMMAND_AFTER_SELECT 3- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 4- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
2	**Handler availability with EVENT_PROFILE_DOWNLOAD** 1- Terminal Profile command is sent to the SIM without the facility of SET_EVENT_LIST, SETUP_IDLE_MODE_TEXT, POLL_INTERVAL and SETUP MENU 2- EnvelopeHandler.getTheHandler() method is called by Applet1 Applet1 is deregistered to EVENT_PROFILE_DOWNLOAD 3-EnvelopeHandler.getTheHandler() method is called by Applet2 Applet2 is deregistered to EVENT_PROFILE_DOWNLOAD	1- Applet1 is triggered by EVENT_PROFILE_DOWNLOAD 2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered by EVENT_PROFILE_DOWNLOAD 3- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	

Id	Description	API/Framework Expectation	APDU Expectation
3	**Envelope Handler availability with EVENT_MENU_SELECTION_HELP_REQUEST** Perform SIM initialization with all the facilities supported Envelope menu selection with help request is sent to the SIM 1-EnvelopeHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- No exception is thrown.	
4	**Envelope Handler availability with EVENT_MENU_SELECTION** 1-Envelope menu selection is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- No exception is thrown.	
5	**Envelope Handler availability with EVENT_FORMATTED_SMS_PP_ENV** 1-A EVENT_FORMATTED_SMS_PP_ENV envelope is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- No exception is thrown.	
6	**Envelope Handler availability with EVENT_UNFORMATTED_SMS_PP_ENV** 1-An unformatted sms pp envelope is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes 3- Applet2 is triggered 4- No exception is thrown.	
7	**Envelope Handler availability with EVENT_FORMATTED_CB** 1-Envelope cell broadcast formatted is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2-No exception is thrown	
8	**Envelope Handler availability with EVENT_UNFORMATTED_CB** 1-Envelope cell broadcast unformatted is		

Id	Description	API/Framework Expectation	APDU Expectation
	sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes 3- Applet2 is triggered 4- No exception is thrown	
9	**Envelope Handler availability with EVENT_TIMER_EXPIRATION** Timer id=1 1-Envelope Timer Expiration is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- No exception is thrown.	
10	**Envelope Handler availability with EVENT_CALL_CONTROL_BY_SIM** 1-Envelope call control by sim is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- No exception is thrown.	
11	**Envelope Handler availability with EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM** 1-Envelope mo short message control by sim is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1.	1- Applet1 is triggered 2- No exception is throw	
12	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_MT_CALL** 1-Envelope event download mt call is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes 3- Applet2 is triggered 4- No exception is thrown.	
13	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_CALL_CONNECTED** 1-Envelope event download call connected is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes 3- Applet2 is triggered	

Id	Description	API/Framework Expectation	APDU Expectation
		4- No exception is thrown.	
14	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_CALL_DISCONECTTED** 1-Envelope event download call disconnected is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered. 2- No exception is thrown. Applet1 finalizes 3- Applet2 is triggered 4- No exception is thrown.	
15	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_LOCATION_STATUS** 1-Envelope event download location status is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes 3- Applet2 is triggered 4- No exception is thrown.	
16	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_USER_ACTIVITY** 1-Envelope event download user activity is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes 3- Applet2 is triggered 4- No exception is thrown	
17	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE** 1-Envelope event download idle screen available is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes 3- Applet2 is triggered 4- No exception is thrown.	
18	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_CARD_READER**		

Id	Description	API/Framework Expectation	APDU Expectation
	_STATUS 1-Envelope event download card reader status is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes 3- Applet2 is triggered 4- No exception is thrown.	
19	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION** 1-Envelope event download language selection is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 Applet1 finalizes. 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2-No exception is thrown. Applet1 finalizes. Applet2 is triggered 3-No exception is thrown.	
20	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION** 1-Envelope event download browser termination is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2-No exception is thrown. Applet1 finalizes. Applet2 is triggered 3-No exception is thrown.	
21	**Envelope Handler availability with EVENT_STATUS_COMMAND** 1-Status command is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes. 3- Applet2 is triggered 4- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
22	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE** 1- Applet1 builds a proactive command OPEN CHANNEL. proactiveHandler.send() method is called	1- Applet1 is registered to EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE and EVENT_EVENT_DOWNLOAD_CH	1- OPEN CHANNEL proactive command is fetched

Id	Description	API/Framework Expectation	APDU Expectation
	2-Envelope event download data available is sent to the SIM with ChannelId=01. 3-EnvelopeHandler.getTheHandler() method is called by Applet1	ANNEL_STATUS 2- Applet1 is triggered 3-No exception is thrown.	TERMINAL RESPONSE is issued with Channel Id = 01
23	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS** 1-Envelope event download channel status is sent to the SIM with ChannelId=01. 2-EnvelopeHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2-No exception is thrown.	
24	**Envelope Handler availability with EVENT_ UNRECOGNIZED_ENVELOPE** 1-An unrecognized Envelope is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes Applet2 is triggered 3- No exception is thrown.	
25	**Envelope Handler availability with EVENT_FORMATTED_SMS_PP_UPD** 1- A formatted Update Record EFsms instruction is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- No exception is thrown.	
26	**Envelope Handler availability with EVENT_UNFORMATTED_SMS_PP_UPD** 1-An unformatted Update Record EFsms instruction is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called by Applet1 3-EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes Applet2 is triggered 3- No exception is thrown.	

6.3.1.3.4 Test Coverage

CRR Number	Test Case Number
CRRN1	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23, 24, 25, 26
CRRC1	1, 2, 21

6.3.1.4 EnvelopeResponseHandler

Test Area Reference: FWK_MHA_ERHD

6.3.1.4.1 Conformance Requirement

6.3.1.4.1.1 Normal Execution

- CRRN1: The handler is available for all triggered toolkit applets from the invocation of the processToolkit method of the toolkit applet until a toolkit applet has posted an envelope response or the first invocation of the ProactiveHandler.send method for the following events:

  EVENT_FORMATTED_SMS_PP_ENV

  EVENT_UNFORMATTED_SMS_PP_ENV

  EVENT_CALL_CONTROL

  EVENT_SMS_MO_CONTROL

  EVENT_UNRECOGNIZED_ENVELOPE

- CRRN2: After a call to the post method the handler is not longer available.

- CRRN3: After a call to the send method the handler is not longer available.

6.3.1.4.1.2 Parameters error

No requirements.

6.3.1.4.1.3 Context Errors

- CRRC1: The handler is not available for the following events:

  EVENT_FORMATTED_SMS_CB

  EVENT_UNFORMATTED_SMS_CB

  EVENT_MENU_SELECTION

  EVENT_MENU_SELECTION_HELP_REQUEST

  EVENT_TIMER_EXPIRATION

  EVENT_EVENT_DOWNLOAD_MT_CALL

  EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

  EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

  EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

  EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

  EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE

  EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

  EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION

  EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION

  EVENT_STATUS_COMMAND

  EVENT_PROFILE_DOWNLOAD

EVENT_FIRST_COMMAND_AFTER_SELECT

EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS

EVENT_FORMATTED_SMS_PP_UPD

EVENT_UNFORMATTED_SMS_PP_UPD

6.3.1.4.2 Test Suite Files

Test Script:	FWK_MHA_ERHD_1.scr
Test Applet:	FWK_MHA_ERHD_1.java
	FWK_MHA_ERHD_2.java
Load Script:	FWK_MHA_ERHD_1.ldr
Cleanup Script:	FWK_MHA_ERHD_1.clr
Parameter File:	FWK_MHA_ERHD_1.par

6.3.1.4.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Toolkit Applet1 and Toolkit Applet2 registration and Envelope Response Handler availability with EVENT_FIRST_COMMAND_AFTER_SELECT** 1- Applet1 is registered to all events defined in TS 43.019 [7]. Using the methods initMenuEntry() for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION and setEventList() for the rest of the events. Applet2 is registered to EVENT_UNFORMATTED_SMS_PP_ENV and EVENT_UNRECOGNIZED_ENVELOPE. 3- Select MF. 3-EnvelopeResponseHandler.getTheHandler() method is called by Applet1 Applet1 is deregistered to EVENT_FIRST_COMMAND_AFTER_SELECT.	1- No exception is thrown 2- Applet1 is triggered by EVENT_FIRST_COMMAND_AFTER_SELECT 3- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
2	**Handler availability with EVENT_PROFILE_DOWNLOAD** 1- Terminal Profile command is sent to the SIM without the facility of SET_EVENT_LIST, SETUP_IDLE_MODE_TEXT, SETUP_MENU and POLL_INTERVAL. 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1 Applet1 is deregistered to EVENT_PROFILE_DOWNLOAD	1- Applet1 Is Triggered By EVENT_PROFILE_DOWNLOAD 2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	

Id	Description	API/Framework Expectation	APDU Expectation
3	**Envelope Response Handler availability with EVENT_MENU_SELECTION_HELP_REQUEST** Perform SIM initialization with all the facilities supported 1-Envelope menu selection with help request is sent to the SIM 2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered. 2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
4	**Envelope Response Handler availability with EVENT_MENU_SELECTION** 1-A envelope menu selection is sent to the SIM 2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
5	**Envelope Response Handler availability with EVENT_FORMATTED_CB** 1-Envelope cell broadcast formatted is sent to the SIM 2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- The applet1 is triggered. 2-A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
6	**Envelope Response Handler availability with EVENT_UNFORMATTED_CB** 1-Envelope cell broadcast unformatted is sent to the SIM 2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered. 2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
7	**Envelope Response Handler availability with EVENT_TIMER_EXPIRATION** 1-Envelope Timer Expiration is sent to the SIM 2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered. 2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
8	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_MT_CALL** 1-Envelope event download mt call is sent to the SIM	1- Applet1 is triggered.	

Id	Description	API/Framework Expectation	APDU Expectation
	`2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`	2 -A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
9	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_CALL_CONNECTED**		
	`1-Envelope event download call connected is sent to the SIM`	1- Applet1 is triggered.	
	`2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`	2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
10	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED**		
	`1-Envelope event download call disconnected is sent to the SIM`	1- Applet1 is triggered.	
	`2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`	2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
11	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_LOCATION_STATUS**		
	`1-Envelope event download location status is sent to the SIM`	1- Applet1 is triggered.	
	`2-Applet1 obtains the Envelope Response Handler`	2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
12	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_USER_ACTIVITY**		
	`1-Envelope event download user activity is sent to the SIM`	1- Applet1 is triggered.	
	`2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`	2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
13	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE**		
	`1-Envelope event download idle screen available is sent to the SIM`	1- Applet1 is triggered.	
	`2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`	2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	

Id	Description	API/Framework Expectation	APDU Expectation

14	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_CARD_READER _STATUS**
	`1-Envelope event download card reader status is sent to the SIM`
	`2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`
		1- Applet1 is triggered
		2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	

15	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_LANGUAGE_ SELECTION**
	`1-Envelope event download language selection is sent to the SIM`
	`2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`
		1- Applet1 is triggered
		2-A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	

Id	Description	API/Framework Expectation	APDU Expectation

16	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_BROWSER_ TERMINATION**
	`1-Envelope event download browser termination is sent to the SIM`
	`2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`
		1- Applet1 is triggered
		2-A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	

17	**Envelope Response Handler availability with EVENT_STATUS_COMMAND**
	`1-Status command is sent to the SIM`
	`2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`
		1- Applet1 is triggered
		2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	

18	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_DATA_AVAILAB LE**
	`1- Applet1 initialises a proactive command OPEN CHANNEL and calls the send() method.`
	`2- Envelope event download data avalaible is sent to the SIM with channelId=01`
	`3-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`
		2- Applet1 is triggered
		3- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown
			1- The OPEN CHANNEL command is fetched.
			TERMINAL RESPONSE IS SENT TO THE SIM with channelId=01

Id	Description	API/Framework Expectation	APDU Expectation
19	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS** 1- Envelope event download channel status is sent to the SIM with channelId=01 2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	

Id	Description	API/Framework Expectation	APDU Expectation

20	**Envelope Response Handler availability with EVENT_FORMATTED_SMS_PP_UPD**
	`1- Update Record EFsms instruction formatted is sent to the SIM`
	`2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`
		1- The applet1 is triggered.
		2-A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	

Id	Description	API/Framework Expectation	APDU Expectation
21	**Envelope Response Handler availability with EVENT_UNFORMATTED_SMS_PP_UPD** 1- Update Record EFsms instruction unformatted is sent to the SIM 2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered. 2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	

Id	Description	API/Framework Expectation	APDU Expectation
22	**Envelope Response Handler availability with EVENT_FORMATTED_SMS_PP_ENV**		
	`1-A formatted sms pp envelope is sent to the SIM`	1- Applet1 is triggered	
	`2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`	2- No exception is thrown.	
	`3-Applet1 builds an additional information for response packet and it calls the post method`		3- The response packet is sent
	`4-Applet1 calls all methods of the Envelope Response Handler (including the inherited method)`	4- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method	
	`5-A EVENT_FORMATTED_SMS_PP_ENV envelope is sent to the SIM`	Applet1 finalizes 5- Applet1 is triggered	
	`6-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`	6- No Exception is thrown	
	`7-Applet1 builds a proactive command and it calls the send() method`		7- The proactive command is sent
	`8-Applet1 calls all methods of the Envelope Response Handler (including the inherited method)`	8- Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method	

Id	Description	API/Framework Expectation	APDU Expectation
23	**Envelope Response Handler availability with EVENT_UNFORMATTED_SMS_PP_ENV** 1-An unformatted sms pp envelope is sent to the SIM 2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1 3-Applet1 builds the envelope response and it calls the post() method 4- Applet1 calls all methods of the Envelope Response Handler (including the inherited method) 5-EnvelopeResponseHandler.getTheHandler() method is called 6-An unformatted sms pp envelope is sent to the SIM 7-EnvelopeResponseHandler.getTheHandler() method is called. 8-Applet1 builds a proactive command and it calls the send() method 9-Applet1 calls all methods of the Envelope Response Handler (including the inherited method) 10-EnvelopeResponseHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2- No exception is thrown. 4- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method Applet1 finalizes 5- Applet2 is triggered. A Toolkit exception HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes 6- Applet1 is triggered. 7- No exception is thrown. 9- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method. Applet1 finalizes 10- Applet2 is triggered. A Toolkit exception HANDLER_NOT_AVAILABLE is thrown.	3- The envelope response is sent 9- The proactive command is fetched and the Terminal response is issued.

Id	Description	API/Framework Expectation	APDU Expectation
24	**Envelope Response Handler availability with EVENT_CALL_CONTROL_BY_SIM** 1-Envelope call control by sim is sent to the SIM 2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1 3-Applet1 builds the envelope response and it calls the postAsBERTLV() method 4-Applet1 calls all methods of the Envelope Response Handler (including the inherited method) 5-Envelope call control by sim is sent to the SIM 6-EnvelopeResponseHandler.getTheHandler() method is called by Applet1 7-Applet1 builds a proactive command and it calls the send() method 8-Applet1 calls all methods of the Envelope Response Handler (including the inherited method)	1- Applet1 is triggered 2- No exception is thrown. 4- Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method Applet1 finalizes 5- Applet1 is triggered 6- No Exception is thrown 8- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method	3- The envelope response is sent 7- The proactive command is fetched and the Terminal response is issued

Id	Description	API/Framework Expectation	APDU Expectation
25	**Envelope Response Handler availability with EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM** 1-Envelope mo short message control by sim is sent to the SIM 2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1 3-Applet1 builds the envelope response and it calls the postAsBERTLV() method 4-Applet1 calls all methods of the Envelope Response Handler (including the inherited method) 5-Envelope mo short message control by sim is sent to the SIM 6-EnvelopeResponseHandler.getTheHandler() method is called by Applet1 7-Applet1 builds a proactive command and it calls the send method 8-Applet1 calls all methods of the Envelope Response Handler (including the inherited method)	1- Applet1 is triggered 2- No exception is thrown. 4- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method Applet1 finalizes 5- Applet1 is triggered 6- No exception is thrown 8- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method	3-The envelope response is sent 7- The proactive command is fetched and the Terminal Response is issued

Id	Description	API/Framework Expectation	APDU Expectation
26	**Envelope Response Handler availability with EVENT_UNRECOGNIZED_ENVELOPE** 1-An unrecognized Envelope is sent to the SIM 2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1 3-Applet1 builds the envelope response and it calls the postAsBERTLV() or post method 4-Applet1 calls all methods of Envelope Response Handler (including the inherited method) 5-EnvelopeResponseHandler.getTheHandler() method is called 6-An unrecognized Envelope is sent to the SIM 7-EnvelopeResponseHandler.getTheHandler() method is called 8-Applet1 builds a proactive command and it calls the send() method 9-Applet1 calls all methods of the Envelope Response Handler (including the inherited method) 10-EnvelopeResponseHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered 2- No exception is thrown. 4- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method Applet1 finalizes 5- Applet2 is triggered. A Toolkit exception HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes 6- Applet1 is triggered. 7- No exception is thrown. 9- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method Applet1 finalizes 10- Applet2 is triggered A Toolkit exception HANDLER_NOT_AVAILABLE is thrown.	3- The envelope response is sent 9- The proactive command is fetched and the Terminal response is issued

Id	Description	API/Framework Expectation	APDU Expectation
27	**The envelope response is sent when a proactive session is ongoing**		
	1-A formatted SMS PP envelope is sent to the SIM.	1- Applet1 is triggered.	
	2-Proactive command DISPLAY TEXT is built and it calls the send() method.		2- 91 XX
	3-A call control by sim envelope is sent to the SIM.	3- Applet1 is triggered	
	4-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	4- No exception is thrown	
	5-Applet1 builds the envelope response and it calls the postAsBERTLV		5-The envelope response is sent 9F YY GET RESPONSE Data 91 XX Fetch DISPLAY TEXT Terminal Response DISPLAY TEXT

Id	Description	API/Framework Expectation	APDU Expectation
28	**Envelope Response Handler availability with EVENT_UNFORMATTED_SMS_PP_ENV in case of multi-triggering** 1-A unformatted sms pp envelope is sent to the SIM 2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1 5-EnvelopeResponseHandler.getTheHandler() method is called by Applet 2 6- Applet2 calls the post( ) method	1- Applet1 is triggered 2- No exception is thrown. 3- Applet1 finalizes 4- Applet2 is triggered. 5- No Exception is thrown Applet2 finalizes	6. The response is checked.

Id	Description	API/Framework Expectation	APDU Expectation
29	**Envelope Response Handler availability with EVENT_UNRECOGNIZED_ENVELOPE in case of multi-triggering**	1- Applet1 is triggered 2- No exception is thrown.	
	`1-An unrecognized Envelope is sent to the SIM`		
	`2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1`	3- Applet1 finalizes	
		4- Applet2 is triggered.	
		5- No Exception is thrown	
	`5-EnvelopeResponseHandler.getTheHandler() method is called by Applet 2`		
		Applet2 finalizes	6- The response is checked
	`6- Applet2 calls the post( ) method`		

6.3.1.4.4 Test Coverage

CRR Number	Test Case Number
CRRN1	20, 21, 22, 23, 24, 25,26,27
CRRN2	20, 21, 22, 23, 24, 25
CRRN3	20, 21, 22, 23, 24, 25
CRRC1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 28, 29

6.3.2 Handler Integrity

6.3.2.1 ProactiveHandler

Test Area Reference: FWK_HIN_PAHD

6.3.2.1.1 Conformance Requirement

6.3.2.1.1.1 Normal Execution

- CRRN1: At the processToolkit invocation the TLV-List is cleared.
- CRRN2: After a call to ProactiveHandler.send method the handler will remain unchanged until the ProactiveHandler.init or appendTLV method are called.

6.3.2.1.1.2 Parameters error

No requirements.

6.3.2.1.1.3 Context Errors

No requirements.

6.3.2.1.2 Test Suite Files:

Test Script:	FWK_HIN_PAHD_1.scr
Test Applet:	FWK_HIN_PAHD_1.java FWK_HIN_PAHD_2.java
Load Script:	FWK_HIN_PAHD_1.ldr
Cleanup Script:	FWK_HIN_PAHD_1.clr
Parameter File:	FWK_HIN_PAHD_1.par

6.3.2.1.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**At the processToolkit invocation the TLV-List is cleared** Applet1 and Applet2 are registered to EVENT_UNFORMATTED_SMS_PP_ENV. 1-An envelope containing an unformatted sms pp is sent to the SIM 2-ProactiveHandler.getLength() method is called by Applet1	1- Applet1 is triggered. 2- The return value is 0	
2	**TLV-List change after the init method invocation** ProactiveHandler.init() method is called by Applet1 1-ProactiveHandler.getLength() method is called by Applet1	1- The return value is 9	
3	**The TLV-List remains unchanged after the send() method invocation** 1-ProactiveHandler.send() method is called by Applet1 2-ProactiveHandler.getLength() method is called by Applet1 It's checked that the content is the same than before the calling to send method using ProactiveHandler.copyValue and Util.arrayCompare methods	2- The return value is 9, and its contents is the same than before the calling to send method	1- The proactive command is fetched and the terminal response is issued.
4	**At the processToolkit invocation the TLV-List is cleared** 1-ProactiveHandler.getLength() method is called by Applet2 2-ProactiveHandler.getValueLength() method is called by Applet2	1- Applet2 is triggered 2- The return value is 0 3- ToolkitException UNAVAILABLE_ELEMENT is thrown	

6.3.2.1.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3, 4
CRRN2	3

6.3.2.2 ProactiveResponseHandler

Test Area Reference: FWK_HIN_PRHD

6.3.2.2.1 Conformance Requirement

6.3.2.2.1.1 Normal Execution

- CRRN1: The ProactiveResponseHandler content is changed after the call to ProactiveHandler.send method and remains unchanged until next call to the ProactiveHandler.send method.

- CRRN2: The ProactiveResponseHandler may not be available before the first call to ProactiveHandler.send method, if available the content is cleared.

6.3.2.2.1.2 Parameters error

No requirements.

6.3.2.2.1.3 Context Errors

No requirements.

6.3.2.2.2 Test Suite Files

Test Script:	FWK_HIN_PRHD_1.scr
Test Applet:	FWK_HIN_PRHD_1.java
Load Script:	FWK_HIN_PRHD_1.ldr
Cleanup Script:	FWK_HIN_PRHD_1.clr
Parameter File:	FWK_HIN_PRHD_1.par

6.3.2.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration and ProactiveResponseHandler obtaining** 1-Applet is registered to all events defined in [7]. Using the methods initMenuEntry for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION and setEventList() for the rest of the events. Terminal Profile command is sent to the SIM without the facilities of SET_EVENT_LIST ,SETUP_IDLE_MODE_TEXT, SETUP_MENU and POLL_INTERVAL. For each event: 2-ProactiveResponseHandler.getTheHandler() is called If handler is available, ProactiveResponseHandler.getLength() is called	1- No exception is thrown 2- Applet is triggered. 3- Behaviour 1: Toolkit Exception HANDLER_NOT_AVAILABLE is thrown. Behaviour 2: No exception is thrown, the return value is 0	
2	**The ProactiveResponseHandler remains unchanged after send method invocation until next send method invocation** 1-Applet builds a proactive command ProactiveHandler.send() method is called 2-ProactiveResponseHandler.getLength() method is called 3-ProactiveHandler.init() method is called 4-ProactiveHandler.send() method is called 5-ProactiveResponseHandler.getLength() method is called	1- The ProactiveResponseHandler contains the terminal response 3- The return value is 12 4- No exception is thrown and the Proactive Response Handler remains unchanged 5- The ProactiveResponseHandler contains the terminal response of the second proactive command 7- The return value is 15	2- A proactive command is fetched The terminal response is sent with length 12 6- A proactive command is fetched The terminal response is sent with length 15

6.3.2.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	1

6.3.2.3 EnvelopeHandler

Test Area Reference: FWK_HIN_ENHD

6.3.2.3.1 Conformance Requirement

6.3.2.3.1.1 Normal Execution

- CRRN1: The EnvelopeHandler and its content are available for all triggered toolkit applets, from the invocation to the termination of their processToolkit method
- CRRN2: The SIM Toolkit Framework guarantees that all triggered toolkit applets receive the data.
- CRRN3: The SIM Toolkit Framework shall convert the Update Record EFsms in the EnvelopeHandler TLV List containing Device Identities TLV, Address TLV and SMS TPDU TLV.
- CRRN4: The getEnvelopeTag() method shall return *BTAG_SMS_PP_DOWNLOAD*.
- CRRN5: The getLength() method shall return the Simple TLV list length.
- CRRN6 The Device Identity Simple TLV is used to store the information about the absolute record number in the EFsms file and the value of the EFsms record status byte.

6.3.2.3.1.2 Parameters error

No requirements.

6.3.2.3.1.3 Context Errors

No requirements.

6.3.2.3.2 Test Suite Files

Test Script:	FWK_HIN_ENHD_1.scr
Test Applet:	FWK_HIN_ENHD_1.java
Load Script:	FWK_HIN_ENHD_1.ldr
Cleanup Script:	FWK_HIN_ENHD_1.clr
Parameter File:	FWK_HIN_ENHD_1.par

6.3.2.3.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet initialization and Envelope Handler integrity checks with EVENT_MENU_SELECTION_HELP_REQUEST** 1- Applet is registered to all events defined in TS 43.019 [7] except EVENT_PROFILE_DOWNLOAD and EVENT_STATUS_COMMAND. Using the methods initMenuEntry() for EVENT_MENU_SELECTION, allocateTimer()for EVENT_TIMER_EXPIRATION, and setEventList() for the rest of the events. Perform SIM initialization with all the facilities supported 2-Envelope menu selection with help request is sent to the SIM 3-EnvelopeHandler.getTheHandler() method is called 4-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_HELP_REQUEST 5-A proactive command DISPLAY TEXT is sent 6-Envelope call control by sim is sent to SIM EnvelopeHandler.getTheHandler() method is called 7- It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES Call Control execution is finished. Check that the TAG_HELP_REQUEST is the TLV selected 8-The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	1-No exception is thrown 2- Applet is triggered 3- No exception is thrown. 4- No exception is thrown 6- Applet is triggered 7- No exception is thrown and the handler contains the envelope call control by SIM 8- The contents of the envelope handler shall be the same as stored in buffer 1	5- 91 xx. A proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
2	**Envelope Handler integrity checks with EVENT_MENU_SELECTION** 1-An envelope menu selection is sent to SIM 2-EnvelopeHandler.getTheHandler() method is called	1- Applet is triggered 2- No exception is thrown.	

Id	Description	API/Framework Expectation	APDU Expectation
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_ITEM_IDENTIFIER		
	4-A proactive command DISPLAY TEXT is sent		
	5-Envelope call control by sim is sent to SIM		4- 91 XX
		5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6- It"s checked the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
	It"s checked that the TAG_ITEM_IDENTIFIER is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()		
		7- The contents of the envelope handler shall be the same as stored in buffer 1	

Id	Description	API/Framework Expectation	APDU Expectation
3	**Envelope Handler integrity checks with EVENT_FORMATTED_SMS_PP_ENV** 1-A formatted sms pp envelope is sent to SIM 2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy( ) The EnvelopeHandler.findTLV() method is called with TAG_SMS_TPDU 4-A proactive command DISPLAY TEXT is sent 5-Envelope call control by sim is sent to SIM EnvelopeHandler.getTheHandler() method is called 6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy( ) and Util.arrayCompare methods The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES Call Control execution is finished. It"s checked that the TAG_SMS_TPDU is the TLV selected 7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	1- Applet is triggered 2- No exception is thrown. 3- No exception is thrown. 5- Applet is triggered 6- No exception is thrown and the handler contains the envelope call control by SIM 7- The contents of the envelope handler shall be the same as stored in buffer 1	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM

Id	Description	API/Framework Expectation	APDU Expectation
4	**Envelope Handler integrity checks with EVENT_UNFORMATTED_SMS_PP_ENV**		
	1-A unformatted sms pp envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy( )	3- No exception is thrown.	
	The EnvelopeHandler.findTLV method is called with TAG_DEVICE_IDENTITIES		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It"s checked that the TAG_DEVICE_IDENTITIES is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1.	

Id	Description	API/Framework Expectation	APDU Expectation
5	**Envelope Handler integrity checks with EVENT_UNFORMATTED_SMS_CB**		
	1-A unformatted cellbroadcast envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_CELLBROADCAST_PAGE		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It"s checked that the TAG_CELLBROADCAST_PAGE is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1.	
6	**Envelope Handler integrity checks with EVENT_TIMER_EXPIRATION**		
	1-A timer expiration envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_TIMER_ID		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	

Id	Description	API/Framework Expectation	APDU Expectation
	EnvelopeHandler.getTheHandler() method is called		
	6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
	It"s checked that the TAG_TIMER_ID is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	

Id	Description	API/Framework Expectation	APDU Expectation
7	**Envelope Handler integrity checks with EVENT_CALL_CONTROL_BY_SIM**		
	1-A call control envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_ADDRESS		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
	It"s checked that the TAG_ADDRESS is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	
8	**Envelope Handler integrity checks with EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM**		
	1-A mo short message control by sim envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_ADDRESS		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		

Id	Description	API/Framework Expectation	APDU Expectation
	6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES Call Control execution is finished. It's checked that the TAG_ADDRESS is the TLV selected 7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	6- No exception is thrown and the handler contains the envelope call control by SIM 7- The contents of the envelope handler shall be the same as stored in buffer 1.	Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
9	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_MT_CALL** 1-A event download mt call envelope is sent to SIM 2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_ADDRESS 4-A proactive command DISPLAY TEXT is sent 5-Envelope call control by sim is sent to SIM EnvelopeHandler.getTheHandler() method is called 6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES Call Control execution is finished. It"s checked that the TAG_ADDRESS is the TLV selected 7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	1- Applet is triggered 2- No exception is thrown. 3- No exception is thrown. 5- Applet is triggered 6- No exception is thrown and the handler contains the envelope call control by SIM 7- The contents of the envelope handler shall be the same as stored in buffer 1	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM

Id	Description	API/Framework Expectation	APDU Expectation
10	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_CALL_CONNECTED**		
	1-A event download call connected envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_ADDRESS		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It"s checked that the TAG_ADDRESS is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1.	

Id	Description	API/Framework Expectation	APDU Expectation
11	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_CALL_DISCONNECTED**		
	1-A event download call disconnected envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_ADDRESS		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
	It"s checked that the TAG_ADDRESS is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1.	
12	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_LOCATION_STATUS**		
	1-A event download location status envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_LOCATION_STATUS		
	4-A proactive command DISPLAY TEXT is sent		4-91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		

Id	Description	API/Framework Expectation	APDU Expectation
	6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It"s checked that the TAG_LOCATION_STATUS is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	
13	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_USER_ACTIVITY**		
	1-A event download user activity envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It's checked that the TAG_DEVICE_IDENTITIES is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	

Id	Description	API/Framework Expectation	APDU Expectation
14	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE**		
	1-A event download idle screen available envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
	It"s checked that the TAG_DEVICE_IDENTITIES is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	

Id	Description	API/Framework Expectation	APDU Expectation
15	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_CARD_READER_STATUS**		
	1-A event download card reader status envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_CARD_READER_STATUS		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It"s checked that the TAG_CARD_READER_STATUS is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	

Id	Description	API/Framework Expectation	APDU Expectation
16	**Envelope Handler integrity checks with UNRECOGNIZED_ENVELOPE**		
	1-A unrecognized envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called The EnvelopeHandler.getValueLength() is called		
	6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	

Id	Description	API/Framework Expectation	APDU Expectation
17	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION**		
	1-A event download language selection envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2-No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3-No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_EVENT_LIST		
	4-A proactive command DISPLAY TEXT is sent		
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	4-91 XX
	EnvelopeHandler.getTheHandler() method is called		
	6-It's checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It's checked that the TAG_EVENT_LIST is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	

Id	Description	API/Framework Expectation	APDU Expectation
18	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION**		
	1-A event download browser termination envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2-No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3-No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_EVENT_LIST		
	4-A proactive command DISPLAY TEXT is sent		
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	4-91 XX
	EnvelopeHandler.getTheHandler() method is called		
	6-It's checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It's checked that the TAG_EVENT_LIST is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	

Id	Description	API/Framework Expectation	APDU Expectation
19	**Envelope Handler integrity checks with EVENT_FORMATTED_SMS_PP_UPD** 1-Update Record EFsms instruction single and formatted is sent to the SIM 2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy( ) The EnvelopeHandler.findTLV() method is called with TAG_SMS_TPDU 4-A proactive command DISPLAY TEXT is sent 5-Envelope call control by sim is sent to SIM EnvelopeHandler.getTheHandler() method is called 6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy( ) and Util.arrayCompare methods The EnvelopeHandler.findTLV() method is called with TAG_SMS_TPDU Call Control execution is finished. It"s checked that the TAG_SMS_TPDU is the TLV selected 7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	1- Applet is triggered 2- No exception is thrown. 3- No exception is thrown. 5- Applet is triggered 6- No exception is thrown and the handler contains the envelope call control by SIM 7- The contents of the envelope handler shall be the same as stored in buffer 1	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM

Id	Description	API/Framework Expectation	APDU Expectation
20	**Envelope Handler integrity checks with EVENT_UNFORMATTED_SMS_PP_UPD**		
	1- Update Record EFsms instruction single and unformatted is sent to the SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy( )	3- No exception is thrown.	
	The EnvelopeHandler.findTLV method is called with TAG_SMS_TPDU		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6-It"s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It"s checked that the TAG_DEVICE_IDENTITIES is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1.	

Id	Description	API/Framework Expectation	APDU Expectation
21	**Check the TLV list conversion for EVENT_FORMATTED_SMS_PP_UPD** 1- An EVENT_FORMATTED_SMS_PP_UPD is sent to the SIM. 2- The findTLV(tag == device identities Tag) is called. 3- The getValueByte(offset == 0) is called. 4- The getValueByte(offset == 1) is called. 5- The findTLV(tag == address Tag) is called. 6- Check the content 7- The findTLV(tag == SMS TPDU Tag) is called. 8- Check the content	1- Applet is triggered 2- No exception is thrown. 3- return the absolute record. 4- return the record status 5- No exception is thrown. 7- No exception is thrown.	

Id	Description	API/Framework Expectation	APDU Expectation
22	**Check TLV list conversion for EVENT_FORMATTED_SMS_PP_UPD** `1- The getLength() method is called`	1. return the Simple TLV list length	

Id	Description	API/Framework Expectation	APDU Expectation
23	**Check TLV list conversion for EVENT_FORMATTED_SMS_PP_UPD** `1- The getEnvelopeTag() method is called`	1- return *BTAG_SMS_PP_DOWNLOAD*	

Id	Description	API/Framework Expectation	APDU Expectation
24	**Check the TLV list conversion for EVENT_UNFORMATTED_SMS_PP_UPD** 1- An EVENT_UNFORMATTED_SMS_PP_UPD is sent to the SIM. 2- The findTLV(tag == device identities Tag) is called. 3- The getValueByte(offset == 0) is called. 4- The getValueByte(offset == 1) is called. 5- The findTLV(tag == address Tag) is called. 6- Check the content 7- The findTLV(tag == SMS TPDU Tag) is called. 8- Check the content	1- Applet is triggered 2- No exception is thrown. 3- return the absolute record. 4- return the record status 5- No exception is thrown. 7- No exception is thrown.	

Id	Description	API/Framework Expectation	APDU Expectation
25	**Check TLV list conversion for EVENT_UNFORMATTED_SMS_PP_UPD** `1- The getLength() method is called`	1. return the Simple TLV list length	

Id	Description	API/Framework Expectation	APDU Expectation
26	**Check TLV list conversion for EVENT_UNFORMATTED_SMS_PP_UPD** 1- The getEnvelopeTag() method is called	1- return *BTAG_SMS_PP_DOWNLOAD*	

6.3.2.3.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
CRRN2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
CRRN3	21, 24
CRRN4	22, 25
CRRN5	23, 26
CRRN6	21, 24

6.3.2.4 EnvelopeResponseHandler

Test Area Reference: FWK_HIN_ERHD

6.3.2.4.1 Conformance Requirement

6.3.2.4.1.1 Normal Execution

- CRRN1: At the processToolkit invocation the TLV-List is cleared.

6.3.2.4.2 Test Suite Files:

Test Script:	FWK_HIN_ERHD_1.scr
Test Applet:	FWK_HIN_ERHD_1.java
Load Script:	FWK_HIN_ERHD_1.ldr
Cleanup Script:	FWK_HIN_ERHD_1.clr
Parameter File:	FWK_HIN_ERHD_1.par

6.3.2.4.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	Applet1 is registered to EVENT_UNRECOGNIZED_ENVELOPE. 1-An unrecognised envelope is sent to the SIM 2- EnvelopeResponseHandler.getTheHandler()is called by the Applet1. 3- EnvelopeResponseHandler.getLength() method is called by Applet1	1- Applet 1 is triggered. 2- The return value shall be 0.	

6.3.2.4.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1

6.3.3 Applet Triggering

6.3.3.1 EVENT_PROFILE_DOWNLOAD

Test Area Reference: FWK_APT_EPDW

6.3.3.1.1 Conformance Requirement

6.3.3.1.1.1 Normal Execution

- CRRN1: Upon the reception of Terminal Profile command by the SIM, the STF stores the ME Profile and then triggers the registered toolkit applets.
- CRRN2: The applet is not triggered by the EVENT_PROFILE_DOWNLOAD once it has deregistered from this event.
- CRRN3: The STF shall not reply busy to a Terminal Profile command

6.3.3.1.1.2 Parameters error

No requirements.

6.3.3.1.1.3 Context Errors

No requirements.

6.3.3.1.2 Test Suite Files

Test Script:	FWK_APT_EPDW_1.scr
Test Applet:	FWK_APT_EPDW_1.java
	FWK_APT_EPDW_2.java
	FWK_APT_EPDW_3.java
Load Script:	FWK_APT_EPDW_1.ldr
Cleanup Script:	FWK_APT_EPDW_1.clr
Parameter File:	FWK_APT_EPDW_1.par

6.3.3.1.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applets registration to EVENT_PROFILE_DOWNLOAD and triggering** Applet1 is registered to the EVENT_PROFILE_DOWNLOAD Applet2 is registered to the EVENT_PROFILE_DOWNLOAD Applet3 is not registered to the EVENT_PROFILE_DOWNLOAD and is registered to EVENT_FORMATTED_SMS_PP_ENV. 1-Terminal Profile command is sent to SIM	1- Applet1 is triggered Applet1 finalizes 2- Applet2 is triggered	

Id	Description	API/Framework Expectation	APDU Expectation
		Applet2 finalizes 3- Applet3 is not triggered	
2	**The STF shall not reply busy to a Terminal Profile command** 1-Formatted sms pp envelope is sent to SIM Applet3 builds a REFRESH proactive command in sim initialization mode 2-ProactiveHandler.send() method is called by applet3 3-Terminal Profile command is sent to SIM Applet1 calls Toolkit Registry.clearEvent(EVENT_PROFILE_DOWNLOAD) 4-Applet2 calls Toolkit Registry.clearEvent(EVENT_PROFILE_DOWNLOAD) ToolkitRegistry.setEvent(EVENT_PROFILE_DOWNLOAD) method is called	1- Applet3 is triggered by the EVENT_FORMATTED_SMS_PP_ENV Applet3 is suspended until the terminal response 3- Applet1 is triggered by EVENT_PROFILE_DOWNLOAD Applet1 finalizes 4- Applet2 is triggered by EVENT_PROFILE_DOWNLOAD Applet2 finalizes Applet3 finalizes	2- A proactive command is sent The terminal Response of the proactive command is sent
3	**Deregistered applets are not triggered** Terminal Profile command is sent to SIM	Applet3 is triggered (Applet1 and Applet2 are not triggered)	

6.3.3.1.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1
CRRN2	3
CRRN3	2

6.3.3.2 EVENT_MENU_SELECTION

Test Area Reference: FWK_APT_EMSE

6.3.3.2.1 Conformance Requirement

6.3.3.2.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_MENU_SELECTION when an Envelope Menu Selection is received with the item identifier of a menu entry of this applet if no proactive session is ongoing.

6.3.3.2.1.2 Parameters error

No requirements.

6.3.3.2.1.3 Context Errors

No requirements.

6.3.3.2.2 Test Suite Files

Test Script:	FWK_APT_EMSE_1.scr
Test Applet:	FWK_APT_EMSE_1.java
	FWK_APT_EMSE_2.java
Load Script:	FWK_APT_EMSE_1.ldr
Cleanup Script:	FWK_APT_EMSE_1.clr
Parameter File:	FWK_APT_EMSE_1.par

6.3.3.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation

1	**Applet registration to EVENT_MENU_SELECTION and triggering**
	ToolkitRegistry.initMenuEntry() method is called in the constructor of applet1 and Applet2.
	For applet1:
	MenuEntry="Applet1"
	Offset=0
	Length=menuEntry.length
	HelpSupported=false
	IconQualifier=0
	IconIdentifier=0
	For applet2:
	MenuEntry="Applet2"
	Offset=0
	Length=menuEntry.length
	HelpSupported=false
	IconQualifier=0
	IconIdentifier=0
	event= EVENT_MENU_SELECTION
	1-ToolkitRegistry.isEventSet() is called in constructor.
	Perform SIM initialization the facility SET UP MENU and without the facilities SET EVENT LIST and POLL INTEVAL features
	2-Item Identifier = 1
	Event Menu Selection envelope is sent to the SIM with the item identifier of a menu entry of applet
	3-Item Identifier = 2
	Event Menu Selection envelope is sent to the SIM with the item identifier of a menu entry of applet
		1- The method must return true.
		2- Applet1 is triggered and applet2 is not triggered
		Applet1 finalizes
		3- Applet2 is triggered and applet1 is not triggered	

6.3.3.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1

6.3.3.3 EVENT_MENU_SELECTION_HELP_REQUEST

Test Area Reference: FWK_APT_EMSH

6.3.3.3.1 Conformance Requirement

6.3.3.3.1.1 Normal Execution

- CRRN1: If an ENVELOPE (MENU_SELECTION_HELP_SUPPORTED) command is received for one entry supporting help, then STF shall trigger the corresponding applet.

- CCRN2: A toolkit applet shall be triggered by the EVENT_MENU_SELECTION_HELP_REQUEST event only if the Menu Id corresponding to the Envelope Menu Selection Help Request received by the SIM Toolkit framework was registered with the helpSupported value set to true.

- CCRN3: If at least one menuId of a Toolkit Applet registers to EVENT_MENU_SELECTION_HELP_REQUEST, the SET UP MENU proactive command sent by the SIM Toolkit Framework shall indicate to the ME that help information is available unless all the menus entries that support help are disabled.

6.3.3.3.1.2 Parameters error

No requirements.

6.3.3.3.1.3 Context Errors

No requirements.

6.3.3.3.2 Test Suite Files

Test Script:	FWK_APT_EMSH_1.scr
Test Applet:	FWK_APT_EMSH_1.java
	FWK_APT_EMSH_2.java
	FWK_APT_EMSH_3.java
Load Script:	FWK_APT_EMSH_1.ldr
Cleanup Script:	FWK_APT_EMSH_1.clr
Parameter File:	FWK_APT_EMSH_1.par

6.3.3.3.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation

1	**Applet registration to EVENT_MENU_SELECTION_HELP_REQUEST and triggering**
	Applet1 and Applet2 are installed
	ToolkitRegistry.InitMenuEntry() method is called in the constructor of Applet1 and Applet2.
	For Applet1 (item id 1):
	MenuEntry="Applet1A"
	Offset=0
	Length=menuEntry.length
	HelpSupported=true
	IconQualifier=0
	IconIdentifier=0
	For Applet1 (item id 2):
	MenuEntry="Applet1B"
	Offset=0
	Length=menuEntry.length
	HelpSupported=false
	IconQualifier=0
	IconIdentifier=0
	event= EVENT_MENU_SELECTION_HELP_REQUEST
	1- ToolkitRegistry.isEventSet() is called in constructor.
		1- The command shall return true.	

	For Applet2 (item id 3):
	MenuEntry="Applet2A"
	Offset=0
	Length=menuEntry.length
	HelpSupported=true
	IconQualifier=0
	IconIdentifier=0
	For Applet2 (item id 4):
	MenuEntry="Applet2B"
	Offset=0
	Length=menuEntry.length
	HelpSupported=false
	IconQualifier=0
	IconIdentifier=0
	event= EVENT_MENU_SELECTION_HELP_REQUEST
	2- ToolkitRegistry.isEventSet() is called in constructor.
		2- The command shall return true.	

	Perform SIM initialization with the facility SET UP MENU and without the facilities SET EVENT LIST and POLL INTERVAL		
	3-Item identifier = 1 Menu Selection Help Request envelope is sent to the SIM with item identifier 1 belonging to applet1	3- Applet1 is triggered and Applet2 is not triggered	
	4-Item identifier = 2 Menu Selection Help Request envelope is sent to the SIM with item identifier 2 belonging to applet1	4 Applet1 and Applet2 are not triggered	

Id	Description	API/Framework Expectation	APDU Expectation
	5-Item identifier = 3 Menu Selection Help Request envelope is sent to the SIM with item identifier 3 belonging to applet2 6-Item identifier = 4 Menu Selection Help Request envelope is sent to the SIM with item identifier 4 belonging to applet2	5- Applet2 is triggered and Applet1 is not triggered 6- Applet2 and Applet1 are not triggered	

Id	Description	API/Framework Expectation	APDU Expectation

2	**Applet deregistration to EVENT_MENU_SELECTION_HELP_REQUEST**
	Applet1 and Applet2 are deleted
	Applet3 is installed
	ToolkitRegistry.InitMenuEntry() method is called in the constructor of Applet3.
	For Applet3 (item id 5):
	MenuEntry="Applet3A"
	Offset=0
	Length=menuEntry.length
	HelpSupported=true
	IconQualifier=0
	IconIdentifier=0
	For Applet3 (item id 6):
	MenuEntry="Applet3B"
	Offset=0
	Length=menuEntry.length
	HelpSupported=true
	IconQualifier=0
	IconIdentifier=0
	For Applet3 (item id 7):
	MenuEntry="Applet3C"
	Offset=0
	Length=menuEntry.length
	HelpSupported=false
	IconQualifier=0
	IconIdentifier=0
	1. Perform SIM initialization with the facility SET UP MENU and without the facilities SET EVENT LIST and POLL INTERVAL
	2. Menu Selection Help Request envelope is sent to the SIM with item identifier 5 belonging to applet3
	3. ToolkitRegistry.disableMenuEntry() method for item id 5 is called by the Menu Selection Help Request Envelope.
	4. Menu Selection Help Request envelope is sent to the SIM with item identifier 6 belonging to applet3
	5. ToolkitRegistry.disableMenuEntry() method for item id 6 is called by the Menu Selection Help Request Envelope.
		2. Applet3 is triggered by EVENT_MENU_SELECTION_HELP_REQUEST
		4. Applet3 is triggered by EVENT_MENU_SELECTION_HELP_REQUEST
			1. The SIM shall issue a SET UP MENU proactive command with Menu Entry ID entry '05', '06' and "07", and Help supported set to true.
			3. The SIM shall issue a SET UP MENU proactive command with Menu Entry ID entry "06" and "07", and Help supported set to true.
			5. The SIM shall issue a SET UP MENU proactive command with Menu Entry ID entry "07", and Help supported set to false.

6.3.3.3.4 Test Coverage

CRR Number	Test Case Number

CRR Number	Test Case Number
CRRN1	1
CRRN2	1
CRRN3	2

6.3.3.4 EVENT_FORMATTED_SMS_PP_ENV

Test Area Reference: FWK_APT_EFSE

6.3.3.4.1 Conformance Requirement

6.3.3.4.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_FORMATTED_SMS_PP_ENV once:
  - it has been registered to this event;
  - a Short Message Point to Point (Single or Concatenated) is received by Envelope APDU(s) and is formatted according to 3GPP TS 23.048 [8];
  - the toolkit applet to be triggered is registered with the corresponding TAR in the SMS TPDU;
  - the security is verified.
- CRRN2: The applet is not triggered by the EVENT_FORMATTED_SMS_PP_ENV once it has deregistered from this event.

6.3.3.4.1.2 Parameters error

No requirements.

6.3.3.4.1.3 Context Errors

No requirements.

6.3.3.4.2 Test Suite Files

Test Script:	FWK_APT_EFSE_1.scr
Test Applet:	FWK_APT_EFSE_1.java
Load Script:	FWK_APT_EFSE_1.ldr
Cleanup Script:	FWK_APT_EFSE_1.clr
Parameter File:	FWK_APT_EFSE_1.par

6.3.3.4.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT FORMATTED_SMS_PP_ENV and triggering** Applet is registered to EVENT_FORMATTED_SMS_PP_ENV and EVENT_UNRECOGNIZED_ENVELOPE 1- A Single Short Message SMS-PP Formatted Data Download is sent to the SIM. 2- A Concatenated Short Message SMS-PP		

Id	Description	API/Framework Expectation	APDU Expectation
	Formatted Data Download is sent to the SIM (composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the second 70)	1- Applet is triggered 2- Applet is triggered	
2	**Applet deregistration** ToolkitRegistry.clearEvent() method is called for EVENT_FORMATTED_SMS_PP_ENV 1- A Single Short Message SMS-PP Data Download is sent to the SIM.. 2- A Concatenated Short Messages SMS-PP Data Download is sent to the SIM (composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the second 70). An unrecognized envelope is sent to the sim ToolkitRegistry.setEvent() method is called for EVENT_FORMATTED_SMS_PP_ENV 3- A Single Short Messages SMS-PP Data Download is sent to the SIM. 4- A Concatenated Short Messages SMS-PP Data Download is sent to the SIM (composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the second 70).	1- Applet is not triggered 2- Applet is not triggered 3- Applet is triggered 4- Applet is triggered	

6.3.3.4.4 Test Coverage

CRR Number	Test Case Number
CRRN1 (See note)	1, 2
CRRN2	2

NOTE: The security checks are not relevant to the test designed in this test area; they will be checked in the "Framework Security Management" section.

6.3.3.5 EVENT_UNFORMATTED_SMS_PP_ENV

Test Area Reference: FWK_APT_EUSE

6.3.3.5.1 Conformance Requirement

6.3.3.5.1.1 Normal Execution

- CRRN1: The applets registers are triggered by the EVENT_UNFORMATTED_SMS_PP_ENV once a Short Message Point to Point (Single or Concatenated) is received by Envelope APDU(s) and is unformatted.

- CRRN2: The applet is not triggered by the EVENT_UNFORMATTED_SMS_PP_ENV once it has deregistered from this event.

6.3.3.5.1.2 Parameters error

No requirements.

6.3.3.5.1.3 Context Errors

No requirements.

6.3.3.5.2 Test Suite Files

Test Script:	FWK_APT_EUSE_1.scr
Test Applet:	FWK_APT_EUSE_1.java
Load Script:	FWK_APT_EUSE_1.ldr
Cleanup Script:	FWK_APT_EUSE_1.clr
Parameter File:	FWK_APT_EUSE_1.par

6.3.3.5.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_UNFORMATTED_SMS_PP_ENV and triggering** Applet is registered to the EVENT_UNFORMATTED_SMS_PP_ENV and EVENT_FORMATTED_SMS_PP_ENV. 1-Toolkit Registry.isEventSet() method is called for EVENT_UNFORMATTED_SMS_PP_ENV 2- A Single and Unformatted SMS-PP Data Download Envelope is sent to the SIM. 3- A Concatenated and Unformatted SMS-PP Data Download Envelope is sent to the SIM (composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the second 70)	1- The method returns true 2- Applet is triggered 3- Applet is triggered	

Id	Description	API/Framework Expectation	APDU Expectation

2	**Applet deregistration**
	`Toolkit Registry.clearEvent()method is called for EVENT_UNFORMATTED_SMS_PP_ENV`
	`1- A Single and Unformatted SMS-PP Data Download Envelope is sent to the SIM.`
	`2- A Concatenated and Unformatted SMS-PP Data Download Envelope is sent to the SIM (composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the second 70)`
	`Applet is triggered by a EVENT_FORMATTED_SMS_PP_ENV`
	`Toolkit Registry.setEvent() method is called for EVENT_UNFORMATTED_SMS_PP_ENV`
	`3- A Single and Unformatted SMS-PP Data Download Envelope is sent to the SIM.`
	`4- A Concatenated and Unformatted SMS-PP Data Download Envelope is sent to the SIM (composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the second 70)`
		1- Applet isn't triggered
		2- Applet isn't triggered
		3- Applet is triggered
		4- Applet is triggered	

6.3.3.5.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

6.3.3.6 EVENT_CALL_CONTROL_BY_SIM

Test Area Reference: FWK_APT_ECCN

6.3.3.6.1 Conformance Requirement

6.3.3.6.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_CALL_CONTROL_BY_SIM once it has registered to this event and an Envelope Call Control is received.
- CRRN2: The applet is not triggered by the EVENT_CALL_CONTROL_BY_SIM once it has deregistered from this event.

6.3.3.6.1.2 Parameters error

No requirements.

6.3.3.6.1.3 Context Errors

No requirements.

6.3.3.6.2 Test Suite Files

Test Script:	FWK_APT_ECCN_1.scr
Test Applet:	FWK_APT_ECCN_1.java
Load Script:	FWK_APT_ECCN_1.ldr
Cleanup Script:	FWK_APT_ECCN_1.clr
Parameter File:	FWK_APT_ECCN_1.par

6.3.3.6.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applets registration to EVENT_CALL_CONTROL_BY_SIM and triggering**		
	`Applet1 is registered to EVENT_CALL_CONTROL_BY_SIM.`		
	`Applet2 is registered to EVENT_FORMATTED_SMS_PP_ENV`		
	`1-An Envelope Call control by SIM is sent to SIM`	1- Applet1 is triggered	
2	**Applet deregistration and registration of the third applet to EVENT_CALL-CONTROL_BY_SIM.**		
	`1-An Envelope Formatted SMS PP envelope is sent to SIM`	1-Applet2 is triggered by EVENT_FORMATTED_SMS_PP_ENV.	
	`Applet2 contructs a DISPLAY TEXT proactive command.`		
	`2-ProactiveHandler.send() method is called`		2- A proactive command DISPLAY TEXT is sent and applet is suspended until the terminal response
	`3-An Envelope Call control by SIM envelope is sent to SIM`	3- Applet1 is triggered	
	`ToolkitRegistry.clearEvent() is called for EVENT_CALL_CONTROL_BY_SIM.`		
		Applet1 finalizes.	TERMINAL RESPONSE of DISPLAY TEXT is sent to the SIM
	`ToolkitRegistry.setEvent() method is called for EVENT_CALL_CONTROL_BY_SIM.`		
		Applet2 finalizes	
3	**Applet triggering**		
	`An Envelope Call control by SIM envelope is sent to SIM`	Applet2 is triggered. (Applet1 is not triggered)	

6.3.3.6.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3
CRRN2	3

6.3.3.7 EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM

Test Area Reference: FWK_APT_EMCN

6.3.3.7.1 Conformance Requirement

6.3.3.7.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM once it has registered to this event and an Envelope MO Short Message Control.
- CRRN2: The applet is not triggered by the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM once it has deregistered from this event.

6.3.3.7.1.2 Parameters error

No requirements.

6.3.3.7.1.3 Context Errors

No requirements.

6.3.3.7.2 Test Suite Files

Test Script:	FWK_APT_EMCN_1.scr
Test Applet:	FWK_APT_EMCN_1.java
	FWK_APT_EMCN_2.java
Load Script:	FWK_APT_EMCN_1.ldr
Cleanup Script:	FWK_APT_EMCN_1.clr
Parameter File:	FWK_APT_EMCN_1.par

6.3.3.7.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM and triggering** Applet1 is reggistered to EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM. Applet2 is registered to EVENT_FORMATTED_SMS_PP_ENV. 1-An Envelope MO short message envelope is sent to SIM	1- Applet1 is triggered.	

Id	Description	API/Framework Expectation	APDU Expectation
2	**Applet deregistration and registration of the third applet to EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM.** **The STF shall not reply busy to a call control envelope** 1-An Envelope formatted SMS PP envelope is sent to SIM. Applet2 builds a DISPLAY TEXT proactive command. 2-ProactiveHandler.send() method is called. 3-An Envelope MO Short message envelope is sent to SIM ToolkitRegistry.clearEvent() for EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM. ToolkitRegistry.setEvent() method is called for EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM.	1- Applet2 is triggered. 3- Applet1 is triggered. Applet1 finalizes. Applet2 finalizes.	2- A Proactive command DISPLAY TEXT is sent and applet is suspended until the terminal response TERMINAL RESPONSE of DISPLAY TEXT is sent to the SIM
3	**Applet3 triggering** An Envelope MO SMS control by SIM envelope is sent to SIM	Applet2 is triggered. (Applet1 is not triggered)	

6.3.3.7.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3
CRRN2	3

6.3.3.8 EVENT_TIMER_EXPIRATION

Test Area Reference: FWK_APT_ETEX

6.3.3.8.1 Conformance Requirement

6.3.3.8.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_TIMER_EXPIRATION once it has been registered to this event and an Envelope Timer Expiration with a Timer Identifier of the applet is received if no proactive session is ongoing.

- CRRN2: The applet is not triggered by the EVENT_TIMER_EXPIRATION once it has been deregistered from this event.

6.3.3.8.1.2 Parameters error

No requirements.

6.3.3.8.1.3 Context Errors

No requirements.

6.3.3.8.2 Test Suite Files

Test Script:	FWK_APT_ETEX_1.scr
Test Applet:	FWK_APT_ETEX_1.java
Load Script:	FWK_APT_ETEX_1.ldr
Cleanup Script:	FWK_APT_ETEX_1.clr
Parameter File:	FWK_APT_ETEX_1.par

6.3.3.8.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_TIMER_EXPIRATION and triggering** Applet is registered to the EVENT_TIMER_EXPIRATION using the allocateTimer() method and to EVENT_FORMATTED_SMS_PP_ENV. event= EVENT_TIMER_EXPIRATION 1-Toolkit Registry.isEventSet() method is called. 2-An Envelope TIMER_EXPIRATION is sent to the SIM.	1- The method returns true 2- Applet is triggered.	
2	**Applet deregistration** Timer id=1 Toolkit Registry.ReleaseTimer() method is called 1-An Envelope timer expiration is sent to the SIM. An Envelope formated sms pp envelope is sent to the sim Toolkit Registry.AllocateTimer() method is called 2-An Envelope TIMER_EXPIRATION is sent to the SIM.	1- Applet isn't triggered 2- Applet is triggered	

6.3.3.8.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

6.3.3.9 EVENT_UNFORMATTED_SMS_CB

Test Area Reference: FWK_APT_EUCB

6.3.3.9.1 Conformance Requirement

6.3.3.9.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_UNFORMATTED_SMS_CB once it has registered to this event and an Envelope Cell Broadcast DownLoad is received.
- CRRN2: The applet is not triggered by the EVENT_UNFORMATTED_SMS_CB once it has deregistered from this event.

6.3.3.9.1.2 Parameters error

No requirements.

6.3.3.9.1.3 Context Errors

No requirements.

6.3.3.9.2 Test Suite Files

Test Script:	FWK_APT_EUCB_1.scr
Test Applet:	FWK_APT_EUCB_1.java
Load Script:	FWK_APT_EUCB_1.ldr
Cleanup Script:	FWK_APT_EUCB_1.clr
Parameter File:	FWK_APT_EUCB_1.par

6.3.3.9.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Applet registration to EVENT_UNFORMATTED_SMS_CB and triggering** Applet is registered to the EVENT_UNFORMATTED_SMS_CB and EVENT_FORMATTED_SMS_PP_ENV. event= EVENT_UNFORMATTED_SMS_CB 1-Toolkit Registry.isEventSet() method is called. 2-An Envelope UNFORMATTED_SMS_CB is sent to the SIM.	1- Method returns true. 2- Applet is triggered	

Id	Description	API Expectation	APDU Expectation

2	**Applet deregistration**
	`Toolkit Registry.ClearEvent()method is called for EVENT_UNFORMATTED_SMS_CB`
	`1-An Envelope UNFORMATTED_SMS_CB is sent to the SIM.`
	`An Envelope formatted sms pp envelope is sent to the sim`
	`event= EVENT_UNFORMATTED_SMS_CB`
	`Toolkit Registry.setEvent() method is called for EVENT_UNFORMATTED_SMS_CB`
	`2-An Envelope UNFORMATTED_SMS_CB is sent to the SIM.`
		1- Applet isn't triggered
		2- Applet is triggered	

6.3.3.9.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

6.3.3.10 EVENT_EVENT_DOWNLOAD_MT_CALL

Test Area Reference: FWK_APT_EDMC

6.3.3.10.1 Conformance Requirement

6.3.3.10.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_MT_CALL once it has registered to this event and an Envelope Event DownLoad MT Call is received.
- CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_MT_CALL once it has deregistered from this event.

6.3.3.10.1.2 Parameters error

No requirements.

6.3.3.10.1.3 Context Errors

No requirements.

6.3.3.10.2 Test Suite Files

Test Script:	FWK_APT_EMSE_1.scr
Test Applet:	FWK_APT_EMSE_1.java
Load Script:	FWK_APT_EMSE_1.ldr
Cleanup Script:	FWK_APT_EMSE_1.clr

Parameter File: FWK_APT_EMSE_1.par

6.3.3.10.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_MT_CALL and triggering** Applet is registered to the EVENT_EVENT_DOWNLOAD_MT_CALL and to EVENT_FORMATTED_SMS_PP_ENV. event= EVENT_EVENT_DOWNLOAD_MT_CALL 1-Toolkit Registry.isEventSet() method is called. 2-An Envelope EVENT_DOWNLOAD_MT_CALL is sent to the SIM.	1- The method returns true 2- Applet is triggered	
2	**Applet deregistration** event= EVENT_EVENT_DOWNLOAD_MT_CALL Toolkit Registry.clearEvent()method is called Perform SIM initialization with all the facilities supported 1-An Envelope EVENT_DOWNLOAD_MT_CALL is sent to the SIM. An Envelope formatted sms pp envelope is sent to the sim event= EVENT_EVENT_DOWNLOAD_MT_CALL Toolkit Registry.setEvent() method is called Perform SIM initialization with all the facilities supported 2-An Envelope EVENT_DOWNLOAD_MT_CALL is sent to the SIM.	1- Applet isn't triggered 2- Applet is triggered	

6.3.3.10.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

6.3.3.11 EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

Test Area Reference: FWK_APT_EDCC

6.3.3.11.1 Conformance Requirement

6.3.3.11.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_CALL_CONNECTED once it has registered to this event and an Envelope Event DownLoad Call Connected is received.

- CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_CALL_CONNECTED once it has deregistered from this event.

6.3.3.11.1.2 Parameters error

No requirements.

6.3.3.11.1.3 Context Errors

No requirements.

6.3.3.11.2 Test Suite Files

Test Script:	FWK_APT_EDCC_1.scr
Test Applet:	FWK_APT_EDCC_1.java
Load Script:	FWK_APT_EDCC_1.ldr
Clean-up Script:	FWK_APT_EDCC_1.clr
Parameter File:	FWK_APT_EDCC _1.par

6.3.3.11.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_CALL_CONNECTED and triggering** Applet is registered to the EVENT_EVENT_DOWNLOAD_CALL_CONNECTED and to EVENT_FORMATTED_SMS_PP_ENV. event= EVENT_EVENT_DOWNLOAD_CALL_CONNECTED 1-Toolkit Registry.isEventSet() method is called. 2-An Envelope EVENT_DOWNLOAD_CALL_CONNECTED is sent to the SIM.	1- Method returns true 2- Applet is triggered.	
2	**Applet deregistration** event=EVENT_EVENT_DOWNLOAD_CALL_CONNECTED Toolkit Registry.clearEvent()method is called Perform SIM initialization with all the facilities supported 1-A call connected event dowload is sent to the SIM. An Envelope formatted sms pp envelope is sent to the sim Event= EVENT_EVENT_DOWNLOAD_CALL_CONNECTED Toolkit Registry.setEvent() method is called Perform SIM initialization with all the facilities supported 2-An Envelope EVENT_DOWNLOAD_CALL_CONNECTED is sent to the SIM.	1- Applet isn't triggered 2- Applet is triggered	

6.3.3.11.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

6.3.3.12 EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

Test Area Reference: FWK_APT_EDCD

6.3.3.12.1 Conformance Requirement

6.3.3.12.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED once it has registered to this event and an Envelope Event DownLoad Call Disconnected is received.
- CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED once it has deregistered from this event.

6.3.3.12.1.2 Parameters error

No requirements.

6.3.3.12.1.3 Context Errors

No requirements.

6.3.3.12.2 Test Suite Files

Test Script:	FWK_APT_EDCD_1.scr
Test Applet:	FWK_APT_EDCD_1.java
Load Script:	FWK_APT_EDCD_1.ldr
Cleanup Script:	FWK_APT_EDCD_1.clr
Parameter File:	FWK_APT_EDCD_1.par

6.3.3.12.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED and triggering** Applet is registered to the EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED and to EVENT_FORMATTED_SMS_PP_ENV. Event=EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED 1-Toolkit Registry.isEventSet() method is called. 2-An Envelope EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED is sent to the SIM.	1- Method returns true 2- Applet is triggered.	

Id	Description	API/Framework Expectation	APDU Expectation

2	**Applet deregistration**
	`Event= EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED Toolkit Registry.clearEvent()method is called`
	`Perform SIM initialization with all the facilities supported`
	`1-An Envelope EVENT_DOWNLOAD_CALL_DISCONNECTED is sent to the SIM.`
	`a formatted sms pp envelope is sent to the sim.`
	`Event= EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED Toolkit Registry.setEvent() method is called`
	`Perform SIM initialization with all the facilities supported`
	`2-An Envelope EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED is sent to the SIM.`
		1- Applet isn't triggered
		2- Applet is triggered	

6.3.3.12.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

6.3.3.13 EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

Test Area Reference: FWK_APT_EDLS

6.3.3.13.1 Conformance Requirement

6.3.3.13.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_LOCATION_STATUS once it has registered to this event and an Envelope Event DownLoad Location Status is received.
- CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_LOCATION_STATUS once it has deregistered from this event.

6.3.3.13.1.2 Parameters error

No requirements.

6.3.3.13.1.3 Context Errors

No requirements.

6.3.3.13.2 Test Suite Files

Test Script:	FWK_APT_EDLS_1.scr
Test Applet:	FWK_APT_EDLS_1.java
Load Script:	FWK_APT_EDLS_1.ldr

Cleanup Script: FWK_APT_EDLS_1.clr

Parameter File: FWK_APT_EDLS_1.par

6.3.3.13.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_LOACTION_STATUS and triggering** Applet is registered to the EVENT_EVENT_DOWNLOAD_LOCATION_STATUS and to EVENT_FORMATTED_SMS_PP_ENV. Event=EVENT_EVENT_DOWNLOAD_LOCATION_STATUS 1-Toolkit Registry.isEventSet() method is called. 2-An Envelope EVENT_EVENT_DOWNLOAD_LOCATION_STATUS is sent to the SIM.	1- Method returns true 2- Applet is triggered.	
2	**Applet deregistration** Event=EVENT_EVENT_DOWNLOAD_LOCATION_STATUS Toolkit Registry.clearEvent()method is called Perform SIM initialization with all the facilities supported 1-An Envelope EVENT_DOWNLOAD_LOCATION_STATUS is sent to the SIM. a formatted sms pp envelope is sent to the sim Event= EVENT_EVENT_DOWNLOAD_LOCATION_STATUS Toolkit Registry.setEvent() method is called Perform SIM initialization with all the facilities supported 2-An Envelope EVENT_DOWNLOAD_LOCATION_STATUS is sent to the SIM.	1- Applet isn't triggered 2- Applet is triggered	

6.3.3.13.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

6.3.3.14 EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

Test Area Reference: FWK_APT_EDUA

6.3.3.14.1 Conformance Requirement

6.3.3.14.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY once it has registered to this event and an Envelope Event DownLoad User Activity is received.
- CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY once it has deregistered from this event.

6.3.3.14.1.2 Parameters error

No requirements.

6.3.3.14.1.3 Context Errors

No requirements.

6.3.3.14.2 Test Suite Files

Test Script:	FWK_APT_EDUA_1.scr
Test Applet:	FWK_APT_EDUA_1.java
Load Script:	FWK_APT_EDUA_1.ldr
Cleanup Script:	FWK_APT_EDUA_1.clr
Parameter File:	FWK_APT_EDUA_1.par

6.3.3.14.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation

1	**Applet registration to EVENT_EVENT_DOWNLOAD_USER_ACTIVITY and triggering**
	`Applet is registered to the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY and to EVENT_FORMATTED_SMS_PP_ENV.`
	`Event= EVENT_EVENT_DOWNLOAD_USER_ACTIVITY`
	`1-Toolkit Registry.isEventSet() method is called.`
	`2-An Envelope EVENT_DOWNLOAD_USER_ACTIVITY is sent to the SIM.`
		1- Method returns true
		2- Applet is triggered	

Id	Description	API/Framework Expectation	APDU Expectation

2	**Applet deregistration**
	`Event= EVENT_EVENT_DOWNLOAD_USER_ACTIVITY Toolkit Registry.clearEvent()method is called`
	`Perform SIM initialization with all the facilities supported`
	`1-An Envelope EVENT_DOWNLOAD_USER_ACTIVITY is sent to the SIM.`
	`a formatted sms pp envelope is sent to the sim`
	`Event= EVENT_EVENT_DOWNLOAD_USER_ACTIVITY Toolkit Registry.setEvent() method is called`
	`Perform SIM initialization with all the facilities supported`
	`2-An Envelope EVENT_DOWNLOAD_USER_ACTIVITY is sent to the SIM.`
		1- Applet isn't triggered
		2- Applet is triggered	

6.3.3.14.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

6.3.3.15 EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE

Test Area Reference: FWK_APT_EDIS

6.3.3.15.1 Conformance Requirement

6.3.3.15.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE once it has registered to this event and an Envelope Event DownLoad Idle Screen Available is received.
- CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE once it has deregistered from this event.

6.3.3.15.1.2 Parameters error

No requirements.

6.3.3.15.1.3 Context Errors

No requirements.

6.3.3.15.2 Test Suite Files

Test Script:	FWK_APT_EDIS_1.scr
Test Applet:	FWK_APT_EDIS_1.java
Load Script:	FWK_APT_EDIS_1.ldr

Cleanup Script: FWK_APT_EDIS_1.clr

Parameter File: FWK_APT_EDIS_1.par

6.3.3.15.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE and triggering** Applet is registered to the EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE and to EVENT_FORMATTED_SMS_PP_ENV Event= EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE 1-Toolkit Registry.isEventSet() method is called. 2-An Envelope EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE is sent to the SIM.	1- Method returns true 2- Applet is triggered	
2	**Applet deregistration** Event=EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE Toolkit Registry.clearEvent()method is called Perform SIM initialization with all the facilities supported 1-An Envelope EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE is sent to the SIM. a formatted sms pp envelope is sent to the sim Event= EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE Toolkit Registry.setEvent() method is called Perform SIM initialization with all the facilities supported 2-An Envelope EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE is sent to the SIM.	1- Applet isn't triggered 2- Applet is triggered	

6.3.3.15.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

6.3.3.16 EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

Test Area Reference: FWK_APT_EDCR

6.3.3.16.1 Conformance Requirement

6.3.3.16.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS once it has registered to this event and Envelope Event DownLoad Card Reader Status is received.
- CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS once it has deregistered from this event.

6.3.3.16.1.2 Parameters error

No requirements.

6.3.3.16.1.3 Context Errors

No requirements.

6.3.3.16.2 Test Suite Files

Test Script:	FWK_APT_EDCR_1.scr
Test Applet:	FWK_APT_EDCR_1.java
Load Script:	FWK_APT_EDCR_1.ldr
Cleanup Script:	FWK_APT_EDCR_1.clr
Parameter File:	FWK_APT_EDCR_1.par

6.3.3.16.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS and triggering** Applet is registered to the EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS and to EVENT_FORMATTED_SMS_PP_ENV Event=EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS 1-Toolkit Registry.isEventSet() method is called. 2-An Envelope EVENT_DOWNLOAD_CARD_READER_STATUS is sent to the SIM.	1- Method returns true 2- Applet is triggered	

Id	Description	API/Framework Expectation	APDU Expectation
2	**Applet deregistration** Event= EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS Toolkit Registry.clearEvent()method is called Perform SIM initialization with all the facilities supported 1-An Envelope EVENT_DOWNLOAD_CARD_READER_STATUS is sent to the SIM. An Envelope formatted sms pp envelope is sent to the sim Event= EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS Toolkit Registry.setEvent() method is called Perform SIM initialization with all the facilities supported 2-An Envelope EVENT_DOWNLOAD_CARD_READER_STATUS is sent to the SIM.	1- Applet isn't triggered 2- Applet is triggered	

6.3.3.16.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

6.3.3.17 EVENT_UNRECOGNIZED_ENVELOPE

Test Area Reference: FWK_APT_EUEV

6.3.3.17.1 Conformance Requirement

6.3.3.17.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_UNRECOGNIZED_ENVELOPE once it has registered to this event and an Unrecognized Envelope is received.
- CRRN2: The applet is not triggered by the EVENT_UNRECOGNIZED_ENVELOPE once it has deregistered from this event.

6.3.3.17.1.2 Parameters error

No requirements.

6.3.3.17.1.3 Context Errors

No requirements.

6.3.3.17.2 Test Suite Files

Test Script:	FWK_APT_EUEN_1.scr
Test Applet:	FWK_APT_EUEN_1.java
Load Script:	FWK_APT_EUEN_1.ldr
Cleanup Script:	FWK_APT_EUEN_1.clr
Parameter File:	FWK_APT_EUEN_1.par

6.3.3.17.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_UNRECOGNIZED_ENVELOPE and triggering** Applet is registered to the EVENT_UNRECOGNIZED_ENVELOPE and to EVENT_FORMMATTED_SMS_PP_ENV Event= EVENT_UNRECOGNIZED_ENVELOPE 1-Toolkit Registry.isEventSet() method is called. 2-An Envelope UNRECOGNIZED_ENVELOPE is sent to the SIM.	1- Method returns true 2- Applet is triggered	
2	**Applet deregistration** Event= EVENT_UNRECOGNIZED_ENVELOPE Toolkit Registry.clearEvent()method is called 1-An Envelope UNRECOGNIZED_ENVELOPE is sent to the SIM. a formatted sms pp envelope is sent to the sim Event= EVENT_UNRECOGNIZED_ENVELOPE Toolkit Registry.setEvent() method is called 2-An Envelope UNRECOGNIZED_ENVELOPE is sent to the SIM.	1- Applet isn't triggered 2- Applet is triggered	

6.3.3.17.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

6.3.3.18 EVENT_STATUS_COMMAND

Test Area Reference: FWK_APT_ESTC

6.3.3.18.1 Conformance Requirement

6.3.3.18.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_STATUS_COMMAND once it has registered to this event and a Status Command is received.
- CRRN2: The applet is not triggered by the EVENT_STATUS_COMMAND once it has deregistered from this event.

6.3.3.18.1.2 Parameters error

No requirements.

6.3.3.18.1.3 Context Errors

No requirements.

6.3.3.18.2 Test Suite Files

Test Script:	FWK_APT_ESTC_1.scr
Test Applet:	FWK_APT_ESTC_1.java
	FWK_APT_ESTC_2.java
	FWK_APT_ESTC_3.java
Load Script:	FWK_APT_ESTC_1.ldr
Cleanup Script:	FWK_APT_ESTC_1.clr
Parameter File:	FWK_APT_ESTC_1.par

6.3.3.18.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applets registration to EVENT_STATUS_COMMAND and triggering** Applet1 is registered to EVENT_STATUS_COMMAND using the requestPollInterval() command. Applet2 is registered to EVENT_STATUS_COMMAND using the RequestPollInterval() command. Applet3 is registered to EVENT_FORMATTED_SMS_PP_ENV. 1-A status command is sent to SIM	1- Applet1 is triggered. Applet1 finalizes 2- Applet2 is triggered. Applet2 finalizes 3- Applet3 is not triggered	

Id	Description	API/Framework Expectation	APDU Expectation
2	**Applet deregistration and registration of the third applet to EVENT_STATUS_COMMAND. The STF shall not reply busy to a call control envelope** 1-A formatted sms pp envelope is sent to SIM Applet3 builds a DISPLAY TEXT. 2- ProactiveHandler.send() is called 3-A status command is sent to SIM. requestPollInteval with POLL_NO_DURATION is called requestPollInteval with POLL_NO_DURATION is called requestPollInterval() method is called.	1- Applet3 is triggered. 3- Applet1 is triggered. Applet1 finalizes 4- Applet2 is triggered. Applet2 finalizes Applet3 finalizes	2- A proactive command DISPLAY TEXT is sent and applet is suspended until the terminal response 5- TERMINAL RESPONSE of DISPLAY TEXT is sent to the SIM
3	**Applet3 triggering** Perform SIM initialization with all the facilities supported Status command is sent to SIM.	Applet3 is triggered. (Applet1 and Applet2 are not triggered)	

6.3.3.18.4 Test Coverage

CR Number	Test Case Number
CRRN1	1, 2, 3
CRRN2	3

6.3.3.19 EVENT_FORMATTED_SMS_CB

Test Area Reference: FWK_APT_EFCB

6.3.3.19.1 Conformance Requirement

6.3.3.19.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_FORMATTED_SMS_CB once:
  - it has been registered to this event;
  - an envelope APDU carrying a Cell Broadcast Page, formatted according to 3GPP TS 23.048 [8], is received;

- the toolkit applet to be triggered is registered with the corresponding TAR in the CB page;
- the security is verified.

- CRRN2: The applet is not triggered by the EVENT_FORMATTED_SMS_CB once it has deregistered from this event.

6.3.3.19.1.2 Parameters error

No requirements.

6.3.3.19.1.3 Context Errors

No requirements.

6.3.3.19.2 Test Suite Files

Test Script:	FWK_APT_EFCB_1.scr
Test Applet:	FWK_APT_EFCB_1.java
Load Script:	FWK_APT_EFCB_1.ldr
Cleanup Script:	FWK_APT_EFCB_1.clr
Parameter File:	FWK_APT_EFCB_1.par

6.3.3.19.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_FORMATTED_SMS_CB and triggering** Applet is registered to EVENT_FORMATTED_SMS_CB and EVENT_FORMATTED_SMS_PP_ENV 1-An Envelope EVENT_FORMATTED_SMS_CB is sent to the SIM.	1-Applet is triggered	
2	**Applet deregistration** ToolkitRegistry.clearEvent() method is called for EVENT_FORMATTED_SMS_CB 1-A formatted SMS CB envelope is sent to the SIM. 2-An envelope SMS-PP formatted is sent to the SIM ToolkitRegistry.setEvent() method is called for EVENT_FORMATTED_SMS_CB 3-An Envelope FORMATTED_SMS_CB is sent to the SIM	1- Applet is not triggered 2- Applet is triggered 3- Applet is triggered	

6.3.3.19.4 Test Coverage

CR Number	Test Case Number
CRRN1 (See note)	1, 2
CRRN2	2
NOTE: The security checks are not relevant to the test designed in this test area; they will be checked in subclause 6.3.6.	

6.3.3.20 EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION

Test Area Reference: FWK_APT_EDLG

6.3.3.20.1 Conformance Requirement

6.3.3.20.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION once it has registered to this event and an Envelope Event DownLoad Language Selection is received.
- CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION once it has deregistered from this event.

6.3.3.20.1.2 Parameters error

No requirements.

6.3.3.20.1.3 Context Errors

No requirements.

6.3.3.20.2 Test Suite Files

Test Script:	FWK_APT_EDLG_1.scr
Test Applet:	FWK_APT_EDLG_1.java
Load Script:	FWK_APT_EDLG_1.ldr
Cleanup Script:	FWK_APT_EDLG_1.clr
Parameter File:	FWK_APT_EDLG_1.par

6.3.3.20.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION and triggering** Applet is registered to the EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION and to EVENT_FORMATTED_SMS_PP_ENV. Event= EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION 1-Toolkit Registry.isEventSet() method is called. 2-An Envelope EVENT_DOWNLOAD_LANGUAGE_SELECTION is sent to the SIM.	1-Method returns true 2- Applet is triggered	

Id	Description	API/Framework Expectation	APDU Expectation

2	**Applet deregistration**
	`Event= EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION Toolkit Registry.clearEvent()method is called`
	`Perform SIM initialization with Profile Download, SMS PP Data Download, Command Result and Language Selection facilities.`
	`1-An Envelope EVENT_DOWNLOAD_LANGUAGE_SELECTION is sent to the SIM.`
	`a formatted sms pp envelope is sent to the sim`
	`Event= EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION Toolkit Registry.setEvent() method is called`
	`Perform SIM initialization with Profile Download, SMS PP Data Download, Command Result and Language Selection facilities.`
	`2-An Envelope EVENT_DOWNLOAD_LANGUAGE_SELECTION is sent to the SIM.`
		1- Applet isn't triggered
		2- Applet is triggered	

6.3.3.20.4 Test Coverage

CR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

6.3.3.21 EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION

Test Area Reference: FWK_APT_EDBT

6.3.3.21.1 Conformance Requirement

6.3.3.21.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION once it has registered to this event and an Envelope Event DownLoad Browser Termination is received.
- CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION once it has deregistered from this event.

6.3.3.21.1.2 Parameters error

No requirements.

6.3.3.21.1.3 Context Errors

No requirements.

6.3.3.21.2 Test Suite Files

Test Script:	FWK_APT_EDBT_1.scr
Test Applet:	FWK_APT_EDBT_1.java
Load Script:	FWK_APT_EDBT_1.ldr
Cleanup Script:	FWK_APT_EDBT_1.clr
Parameter File:	FWK_APT_EDBT_1.par

6.3.3.21.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_ BROWSER_TERMINATION and triggering** Applet is registered to the EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION and to EVENT_FORMATTED_SMS_PP_ENV Event= EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION 1-Toolkit Registry.isEventSet() method is called. 2-An Envelope EVENT_DOWNLOAD_BROWSER_TERMINATION is sent to the SIM.	1-Method returns true 2- Applet is triggered	
2	**Applet deregistration** Event= EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION Toolkit Registry.clearEvent()method is called Perform SIM initialization with Profile Download, SMS PP Data Download, Command Result and Browser Termination facilities. 1-An Envelope EVENT_DOWNLOAD_BROWSER_TERMINATION is sent to the SIM. a formatted sms pp envelope is sent to the sim Event= EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION Toolkit Registry.setEvent() method is called Perform SIM initialization with Profile Download, SMS PP Data Download, Command Result and Browser Termination facilities. 2-An Envelope EVENT_DOWNLOAD_BROWSER_TERMINATION is sent to the SIM.	1- Applet isn't triggered 2- Applet is triggered	

6.3.3.21.4 Test Coverage

CR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

6.3.3.22 EVENT_FIRST_COMMAND_AFTER_SELECT

Test Area Reference: FWK_APT_EFCA

6.3.3.22.1 Conformance Requirement

6.3.3.22.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_FIRST_COMMAND_AFTER_SELECT once it has registered to this event; Upon reception of the first command received by the GSM application after it has been selected, or after the ATR if it is the default application, and before the Status Word of the processed command has been sent back by the GSM application, the toolkit framework shall trigger all the toolkit applets registered to this event.
- CRRN2: The applet is not triggered by the EVENT_FIRST_COMMAND_AFTER_SELECT once it has deregistered from this event.
- CRRN3: If the first command received by the GSM application is a toolkit applet triggering command (e.g. TERMINAL PROFILE), the toolkit applets registered on the EVENT_FIRST_COMMAND_AFTER_SELECT event shall be triggered first.

6.3.3.22.2 Test Suite Files

Test Script:	FWK_APT_EFCA_1.scr
Test Applet:	FWK_APT_EFCA_1.java
	FWK_APT_EFCA_2.java
	FWK_APT_EFCA_3.java
	FWK_APT_EFCA_4.java
	FWK_APT_EFCA_5.java
Load Script:	FWK_APT_EFCA_1.ldr
Cleanup Script:	FWK_APT_EFCA_1.clr
Parameter File:	FWK_APT_EFCA_1.par

6.3.3.22.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applets registration to EVENT_FIRST_COMMAND_AFTER_SELECT and triggering** Applet1 is registered to the EVENT_FIRST_COMMAND_AFTER_SELECT Applet2 is registered to the EVENT_PROFILE_DOWNLOAD. Applet3 is registered to EVENT_FORMATTED_SMS_PP_ENV. 1-Terminal Profile command is sent to the SIM. Applet1 deregisters from EVENT_FIRST_COMMAND_AFTER_SELECT. 2- Applet2 deregisters from EVENT_PROFILE_DOWNLOAD. 3-Envelope(SMS-PP-DOWNLOAD) formatted is sent to the SIM 4-Applet3 calls setEvent() on event EVENT_FIRST_COMMAND_AFTER_SELECT.	1- Applet1 is triggered by EVENT_FIRST_COMMAND_AFTER_SELECT Applet1 finalizes Applet2 is triggered by EVENT_PROFILE_DOWNLOAD Applet2 finalizes Applet3 is not triggered 3-Applet3 is triggered.	
2	**Deregistered applets are not triggered** 1-Reset then Terminal Profile command is sent to the SIM 2-Applet3 calls setEvent() on EVENT_PROFILE_DOWNLOAD.	1-Applet3 is triggered. Applet1 and Applet2 are not triggered. 2-Applet3 finalizes.	
3	**Install a 4th applet registered to EVENT_FIRST_COMMAND_AFTER_SELECT and EVENT_PROFILE_DOWNLOAD** Applet4 is installed, with the same priority level as Applet3. 1-Reset then Terminal Profile command is sent to the SIM Delete all applets.	1- Applet4 is triggered by EVENT_FIRST_COMMAND_AFTER_SELECT. Applet3 is triggered by EVENT_FIRST_COMMAND_AFTER_SELECT. Applet4 is triggered by EVENT_PROFILE DOWNLOAD. Applet3 is triggered by EVENT_PROFILE_DOWNLOAD.	
4	**Check that the applet is triggered before the first SW is sent.** 1-Install Applet 5. Applet 5 is registered with two entries in the menu entries list. Applet5 is also registered to EVENT_FIRST_COMMAND_AFTER_SELECT. 2-Reset and TERMINAL PROFILE. 3-Applet disables a menu entry.	2- Applet 5 is triggered	3-The SETUP MENU proactive command is fetched. There is only one item for Applet5.

NOTE: Testing the triggering of an applet upon the first command after select is not possible.

6.3.3.22.4 Test Coverage

CR Number	Test Case Number
CRRN1	1,2,3, 4
CRRN2	3
CRRN3	1, 4

6.3.3.23 EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

Test Area Reference: FWK_APT_EDDA

6.3.3.23.1 Conformance Requirement

6.3.3.23.1.1 Normal Execution

- CRRN1: For EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE, the framework shall only trigger the applet registered to this event with the appropriate channel identifier.
- CRRN2: The registration to the EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE is effective once the toolkit applet has issued a successful OPEN CHANNEL proactive command, and valid till the first successful CLOSE CHANNEL or the end of card session.
- CRRN3: When a Toolkit Applet has sent an OPEN CHANNEL proactive command and received a successful TERMINAL RESPONSE, the framework shall register the received channel identifier for the calling Toolkit Applet.
- CRRN4: When a Toolkit Applet has sent a CLOSE CHANNEL proactive command and received a successful TERMINAL RESPONSE, the framework shall release the channel identifier contained in the command. A successful TERMINAL RESPONSE means that the result of the proactive command execution belongs to command performed category (i.e. General Result ='0x').

6.3.3.23.2 Test Suite Files

Test Script:	FWK_APT_EDDA_1.scr
Test Applet:	FWK_APT_EDDA_1.java
Load Script:	FWK_APT_EDDA_1.ldr
Cleanup Script:	FWK_APT_EDDA_1.clr
Parameter File:	FWK_APT_EDDA_1.par

6.3.3.23.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE** Applet1 is registered to Unformatted SMS PP Envelope. 1- Unformatted SMS PP envelope is sent to the SIM. 2- Applet calls setEvent() with the event EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE. 3- An envelope Event Download Data Available is sent to the SIM Channel Status = 81 00 4- Unformatted SMS PP envelope is sent to the SIM. 5- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init() method. 6- send() method is called to register to this event. 8- An envelope Event Download Data Available is sent to the SIM with Channel Status = 01 00. 9- Unformatted SMS PP envelope is sent to the SIM. 10- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init() method. 11- send() method is called to register to this event.	1- Applet1 is triggered by Unformatted SMS PP envelope. 2- Applet1 finalizes. 3- Applet1 is not triggered. 4- Applet1 is triggered by Unformatted SMS PP envelope. 7- Applet1 finalizes. 8- Applet1 is not triggered. 9- Applet1 is triggered by EVENT_UNFORMATTED_SMS_PP_ENV. 12- Applet1 finalizes.	6- OPEN CHANNEL proactive command is fetched. Unsuccessful TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM. 11- OPEN CHANNEL proactive command is fetched. Successful TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM with Channel Id = 01.
2	**Applet triggering to EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE** 1- An envelope Event Download Data Available is sent to the SIM Channel Status = 81 00.	1- Applet1 is triggered.	
3	**Applet deregistration to EVENT_EVENT_ DOWNLOAD_DATA_ AVAILABLE** 0- Unformatted SMS PP envelope is sent to the SIM. 1- Applet1 initialises and sends an OPEN CHANNEL proactive command. 2- Applet1 builds a CLOSE CHANNEL	0- Applet1 is triggered.	1- OPEN CHANNEL proactive command is fetched. Successful terminal response is sent, with channelId=02.

Id	Description	API/Framework Expectation	APDU Expectation
	Proactive Command calling ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods. 3- An envelope Event Download Data Available is sent to the SIM. Channel Status = 82 00 4- Applet1 builds a CLOSE CHANNEL Proactive Command calling ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods.	3- Applet1 is triggered. 5- Applet1 finalizes.	2- CLOSE CHANNEL proactive command is fetched. Unsuccessful TERMINAL RESPONSE of CLOSE CHANNEL is sent to the SIM. 4- CLOSE CHANNEL proactive command is fetched. Successful TERMINAL RESPONSE of CLOSE CHANNEL is sent to the SIM with Channel Id = 02.
4	**Applet triggering to EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE** 1- An envelope Event Download Data Available is sent to the SIM Channel Status = 82 00.	1- Applet1 is not triggered.	
5	**Applet1 not triggered after a reset** 0- Applet1 is triggered by an unformatted SMS PP Envelope 1- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init() method. 2- send() method is called to register to this event. 3- isEventSet() method is called. 4- Reset the card. 5- An envelope Event Download Data Available is sent to the SIM Channel Status = 82 00.	3- returns true. 5- Applet1 is not triggered.	1- OPEN CHANNEL proactive command is fetched. 2- Successful TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM with Channel Id = 02.

6.3.3.23.4 Test Coverage

CR Number	Test Case Number
CRRN1	2
CRRN2	1, 4, 5
CRRN3	1
CRRN4	3

6.3.3.24 EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS

Test Area Reference: FWK_APT_EDCS

6.3.3.24.1 Conformance Requirement

6.3.3.24.1.1 Normal Execution

- CRRN1: For EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS, the framework shall only trigger the applet registered to this event with the appropriate channel identifier.

- CRRN2: The registration to the EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS is effective once the toolkit applet has issued a successful OPEN CHANNEL proactive command, and valid till the first successful CLOSE CHANNEL or the end of the card session.
- CRRN3: When a Toolkit Applet has sent an OPEN CHANNEL proactive command and received a successful TERMINAL RESPONSE, the framework shall register the received channel identifier for the calling Toolkit Applet.
- CRRN4: When a Toolkit Applet has sent a CLOSE CHANNEL proactive command and received a successful TERMINAL RESPONSE, the framework shall release the channel identifier contained in the command. A successful TERMINAL RESPONSE means that the result of the proactive command execution belongs to command performed category (i.e. General Result ='0x').

6.3.3.24.2 Test Suite Files

Test Script:	FWK_APT_EDCS_1.scr
Test Applet:	FWK_APT_EDCS_1.java
Load Script:	FWK_APT_EDCS_1.ldr
Cleanup Script:	FWK_APT_EDCS_1.clr
Parameter File:	FWK_APT_EDCS_1.par

6.3.3.24.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS** Applet1 is registered to Unformatted SMS PP Envelope. 1-Unformatted SMS PP envelope is sent to the SIM. 2-The applet calls setEvent() with EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS. 3- An envelope Event Download Channel Status is sent to the SIM. Channel Status = 81 00 4-Unformatted SMS PP envelope is sent to the SIM. 5- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init() method. 6- send() method is called to register to this event. 8- An envelope Event Download Data Available is sent to the SIM with Channel Status = 01 00. 9- Unformatted SMS PP envelope is sent to the SIM. 10- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init() method. 11- send() method is called to register to this event a second time.	1- Applet1 is triggered by Unformatted SMS PP envelope 2- Applet1 finalizes. 3- Applet1 is not triggered. 4- Applet1 is triggered by Unformatted SMS PP envelope. 7- Applet finalizes. 8- Applet1 is not triggered. 9- Applet1 is triggered by EVENT_UNFORMATTED_SMS_PP_ENV. 12- Applet1 finalizes.	6- OPEN CHANNEL proactive command is fetched. Unsuccessful TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM. 11- OPEN CHANNEL proactive command is fetched. Successful TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM with Channel Id = 01.
2	**Applet triggering to EVENT_EVENT_DOWNLOAD_CHANNEL STATUS** 1- An envelope Event Download Channel Status is sent to the SIM. Channel Status = 81 00	1- Applet1 is triggered.	
3	**Applet deregistration to EVENT_EVENT_ DOWNLOAD_CHANNEL STATUS** 0- Unformatted SMS PP envelope is sent to the SIM. 1-Applet1 initialises and sends an OPEN CHANNEL proactive command. 2- Applet1 builds a CLOSE CHANNEL Proactive Command calling ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods.	0- Applet1 is triggered. 3- The applet is triggered.	OPEN CHANNEL proactive command is fetched. Successful terminal response is sent, with channelId=02. 2-CLOSE CHANNEL proactive command is fetched. Unsuccessful TERMINAL

Id	Description	API/Framework Expectation	APDU Expectation
	3-An envelope Event Download Channel Status is sent to the SIM. Channel Status = 82 00 4- Applet1 builds a Close Channel Proactive Command calling ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods.	5- Applet1 finalizes.	RESPONSE of CLOSE CHANNEL is sent to the SIM. 4- CLOSE CHANNEL proactive command is fetched. Successful TERMINAL RESPONSE of CLOSE CHANNEL is sent to the SIM with Channel Id = 02.
4	**Applet triggering to EVENT_EVENT_DOWNLOAD_CHANNEL STATUS** 1- An envelope Event Download Channel Status is sent to the SIM. Channel Status = 82 00	Applet1 is not triggered.	
5	**Applet1 not triggered after a reset** 0- Applet1 is triggered by an unformatted SMS PP Envelope. 1- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init() method. 2- send() method is called to register to this event. 3- isEventSet() method is called. 4- Reset the card. 5- An envelope Event Download Data Available is sent to the SIM Channel Status = 82 00.	3- returns true. 5- Applet1 is not triggered.	1- OPEN CHANNEL proactive command is fetched. 2- Successful TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM with Channel Id = 02.

6.3.3.24.4 Test Coverage

CR Number	Test Case Number
CRRN1	2
CRRN2	1, 4, 5
CRRN3	1
CRRN4	3

6.3.3.25 EVENT_FORMATTED_SMS_PP_UPD

Test Area Reference: FWK_APT_EFSU

6.3.3.25.1 Conformance Requirement

6.3.3.25.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_FORMATTED_SMS_PP_UPD once:
  - it has been registered to this event,
  - a Short Message Point to Point (Single or Concatenated) is received by Update Record EFsms APDU(s) and is formatted according to TS 23.048 [8],
  - the toolkit applet to be triggered is registered with the corresponding TAR in the SMS TPDU,

- CRRN2: The applets are not triggered by the EVENT_FORMATTED_SMS_PP_UPD once it has deregistered from this event.

6.3.3.25.2 Test Suite Files

Test Script:	FWK_APT_EFSU_1.scr
Test Applet:	FWK_APT_EFSU_1.java
Load Script:	FWK_APT_EFSU_1.ldr
Cleanup Script:	FWK_APT_EFSU_1.clr
Parameter File:	FWK_APT_EFSU_1.par

6.3.3.25.3 Test Procedure

6.3.3.25.4 Test Coverage

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT FORMATTED_SMS_PP_UPD and triggering** Applet is registered to EVENT_FORMATTED_SMS_PP_UPD and EVENT_UNRECOGNIZED_ENVELOPE 1. Toolkit Registry.isEventSet() method is called for EVENT_FORMATTED_SMS_PP_UPD 2. Short Message Point to Point Single and Formatted is received by Update Record EFsms APDU. 3. Short Message Point to Point Concatenated Formatted is received by Update Record EFsms APDU(s) (The Concatenated Message is composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the second 70).	1- The method returns true. 2- Applet is triggered. 3- Applet is triggered on reception of the last concatenated SMS	
2	**Applet deregistration** ToolkitRegistry.clearEvent() method is called for EVENT_FORMATTED_SMS_PP_UPD 1. Short Message Point to Point Single and Formatted is received by Update Record EFsms APDU. 2. Short Message Point to Point Concatenated and Formatted is received by Update Record EFsms APDU(s). (The Concatenated Message is composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the second 70). An unrecognized envelope is sent to the sim ToolkitRegistry.setEvent() method is called for EVENT_FORMATTED_SMS_PP_UPD 3. Short Message Point to Point Single and Formatted is received by Update Record EFsms APDU. 4. Short Message Point to Point Concatenated Formatted is received by Update Record EFsms APDU(s). (The Concatenated Message is composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the second 70).	1- Applet is not triggered 2- Applet is not triggered 3- Applet is triggered 4- Applet is triggered on reception of the last concatenated SMS.	

CRR Number	Test Case Number
CRRN1 (See note)	1,2
CRRN2	2

NOTE: The security checks are not relevant to the test designed in this test area; they will be checked in the "Framework Security Management" section.

6.3.3.26 EVENT_UNFORMATTED_SMS_PP_UPD

Test Area Reference: FWK_APT_EUSU

6.3.3.26.1 Conformance Requirement

6.3.3.26.1.1 Normal Execution

- CRRN1: The applets registers are triggered by the EVENT_UNFORMATTED_SMS_PP_UPD once a Short Message Point to Point (Single or Concatenated) is received by Update Record EFsms APDU(s) and is unformatted.

- CRRN2: The applets are not triggered by the EVENT_UNFORMATTED_SMS_PP_UPD once it has deregistered from this event.

6.3.3.26.2 Test Suite Files

Test Script:	FWK_APT_EUSU_1.scr
Test Applet:	FWK_APT_EUSU_1.java
Load Script:	FWK_APT_EUSU_1.ldr
Cleanup Script:	FWK_APT_EUSU_1.clr
Parameter File:	FWK_APT_EUSU_1.par

6.3.3.26.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT UNFORMATTED_SMS_PP_UPD and triggering** Applet is registered to EVENT_UNFORMATTED_SMS_PP_UPD and EVENT_UNRECOGNIZED_ENVELOPE 1. Toolkit Registry.isEventSet() method is called for EVENT_UNFORMATTED_SMS_PP_UPD 2. Short Message Point to Point Single and Unformatted is received by Update Record EFsms APDU 3. Short Message Point to Point Concatenated and Unformatted is received by Update Record EFsms APDU (The Concatenated Message is composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the second 70).	1- Applet is not triggered 2- Applet is triggered. 3- Applet is triggered on reception of the last concatenated SMS.	
2	**Applet deregistration** ToolkitRegistry.clearEvent() method is called for EVENT_UNFORMATTED_SMS_PP_UPD 1. Short Message Point to Point Single and Unformatted is received by Update Record EFsms APDU 2. Short Message Point to Point Concatenated and Unformatted is received by Update Record EFsms APDU(s) (The Concatenated Message is composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the second 70). An unrecognized envelope is sent to the sim ToolkitRegistry.setEvent() method is called for EVENT_UNFORMATTED_SMS_PP_UPD 3. Short Message Point to Point Single and Unformatted is received by Update Record EFsms APDU 4. Short Message Point to Point Concatenated and Unformatted is received by Update Record EFsms APDU(s) (The Concatenated Message is composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the second 70).	1- Applet is not triggered 2- Applet is not triggered. 3- Applet is triggered 4- Applet is triggered on reception of the last concatenated SMS	

6.3.3.26.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1,2
CRRN2	2

6.3.4 Proactive Command Sending by the STF

6.3.4.1 System Proactive Commands

Test Area Reference: FWK_PCS_SPCO

6.3.4.1.1 Conformance Requirements

6.3.4.1.1.1 Normal Execution

- CRRN1: When a toolkit applet changes a menu entry of its registry object, the SIM Toolkit Framework shall dynamically* update the menu stored in the ME during the current card session
- CRRN2: The STF shall use the data of the EFsume file when issuing the SET UP MENU proactive command.
- CRRN3: For all EVENT_EVENT_DOWNLOAD_*: When a toolkit applet changes one or more of these requested events of its registry object, the STF shall dynamically* update the event list stored in the ME during the current card session by SET UP EVENT LIST proactive command.

NOTE: *The STF shall send its system proactive command as soon as no proactive session is pending and all the applets registered to the current events have been triggered and have returned from the processToolkit method invocation.

6.3.4.1.1.2 Parameters error

No requirements.

6.3.4.1.1.3 Context Errors

No requirements.

6.3.4.1.2 Test Suite Files

Test Script: FWK_PCS_SPCO_1.scr

Test Applet: FWK_PCS_SPCO_1.java

Load Script: FWK_PCS_SPCO_1.ldr

Cleanup Script: FWK_PCS_SPCO_1.clr

Parameter File: FWK_PCS_SPCO_1.par

6.3.4.1.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Install Applet 1, Registered to the EVENT_EVENT_DOWNLOAD_MT_CALL and EVENT_EVENT_DOWNLOAD_ LOCATION_STATUS** `Perform SIM initialization with EVENT DOWNLOAD facilities supported`		setEventList proactive command **[Event list]= '19020003'** or **'99020003'**

Id	Description	API/Framework Expectation	APDU Expectation
2	**Trigger the applet by ENVELOPE (SMS_FORMATTED_PP) command** `Clear the events and build a display text command`		1. DISPLAY TEXT Proactive command 2. SET UP EVENT LIST Proactive command [CommandQualifier]= 00h

6.3.4.1.4 Test Coverage

CRR number	Test case number
N1	see: subclause6.2.9.2, CRRN1, subclause 6.2.9.4, CRRN3, subclause 6.2.9.5 CRRN4, subclause 6.2.9.8 CRRN1
N2	see: subclause 6.2.9.2 CRRN1, subclause 6.2.9.8 CRRN1
N3	1,2

6.3.4.2 Interaction with GSM commands

Test Area Reference: FWK_PCS_IGCO

6.3.4.2.1 Conformance Requirements

6.3.4.2.1.1 Normal Execution

- CRRN1: The STF shall process a GSM command even when a proactive command is pending (before and after the FETCH command until the terminal response). The STF shall answer with the SW1 and SW2 described in 3GPP TS 51.011 [3] and 3GPP TS 51.014 [4].

6.3.4.2.1.2 Parameters error

No requirements.

6.3.4.2.1.3 Context Errors

No requirements.

6.3.4.2.2 Test Suite Files

Test Script:	FWK_PCS_IGCO_1.scr
Test Applet:	FWK_PCS_IGCO_1.java
Load Script:	FWK_PCS_IGCO_1.ldr
Cleanup Script:	FWK_PCS_IGCO_1.clr
Parameter File:	FWK_PCS_IGCO_1.par

6.3.4.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation

1	**Interaction with GSM Commands after TERMINAL PROFILE in connection with FETCH and TERMINAL RESPONSE**
	`Applet is registered to Menu Selection`
	`RST`
	`TERMINAL PROFILE`
	`(Profile: supports all facilities except: SET UP EVENT LIST, POLL INTERVAL and POLLING OFF)`
	`1- System issues a proactive command SETUP_MENU`
	`2- SELECT MF`
	`3- GET RESPONSE (6 Bytes)`
	`4- Failed SELECT File`
	`5- FETCH`
	`6- SELECT MF`
	`7- GET RESPONSE (6 Bytes)`
	`8- TERMINAL RESPONSE`
			1- 91xx
			2- 9Fxx
			3- 91xx
			4- 9404
			5- Proactive Command: SETUP MENU
			6- 9Fxx
			7- 9000
			8- 9000

2	**Interaction with GSM Commands after ENVELOPE (MENU SELECTION) in connection with FETCH and TERMINAL RESPONSE**
	`Menu Entry ID = 0x01`
	`1- SELECT MF`
	`2- GET RESPONSE (6 Bytes)`
	`3- Failed SELECT File`
	`4- FETCH`
	`5- SELECT MF`
	`6- GET RESPONSE (6 Bytes)`
	`7- TERMINAL RESPONSE`
			1- 9FXX
			2- 91XX
			3- 9404
			4- Proactive Command: DISPLAY TEXT
			5- 9FXX
			6- 9000
			7- 9000

3	**Interaction with GSM Commands after TERMINAL RESPONSE in proactive command session in connection with FETCH and TERMINAL RESPONSE**
	`Menu Entry ID = 0x02`
	`1- SELECT MF`
	`2- GET RESPONSE (6 Bytes)`
	`3- FETCH`
	`4- SELECT MF`
	`5- GET RESPONSE (6 Bytes)`
	`6- Failed SELECT File`
	`7- TERMINAL RESPONSE`
	`8- SELECT MF`
	`9- GET RESPONSE (6 Bytes)`
	`10-Failed SELECT File`
	`11-FETCH`
	`12-SELECT MF`
	`13-GET RESPONSE (6 Bytes)`
	`14-TERMINAL RESPONSE`
			1- 9FXX
			2- 91XX
			3- Proactive Command: DISPLAY TEXT
			4- 9FXX
			5- 9000
			6- 9404
			7- 9000
			8- 9FXX
			9- 91XX
			10-9404
			11-Proactive Command: DISPLAY TEXT
			12-9FXX
			13-9000
			14-9000

6.3.4.2.4 Test Coverage

CRR number	Test case number
N1	1, 2, 3

6.3.4.3 Proactive Command Control

Test Area Reference: FWK_PCS_PCCO

6.3.4.1.1 Conformance Requirements

6.3.4.1.1.1 Normal Execution

- CRRN1: The SIM Toolkit Framework shall prevent the toolkit applet to issue the following proactive commands: SET UP MENU, SET UP EVENT LIST, POLL INTERVAL, POLLING OFF. If an applet attempts to issue such a command, the SIM Toolkit Framework shall throw an exception.
- CRRN2: The SIM Toolkit Framework shall prevent a toolkit applet to issue a TIMER MANAGEMENT proactive command using a timer identifier, which is not allocated to it. If an applet attempts to issue such a command, the SIM Toolkit Framework shall throw an exception.
- CRRN3: The SIM Toolkit Framework shall prevent a toolkit applet to issue a SEND DATA, RECEIVE DATA and CLOSE CHANNEL proactive commands using a channel identifier, which is not allocated to it. If an applet attempts to issue such a command the SIM Toolkit Framework shall throw an exception.
- CRRN4: The SIM Toolkit Framework shall prevent a toolkit applet to issue an OPEN CHANNEL proactive command if it exceeds the maximum number of channel allocated to this applet. If an applet attempts to issue such a command the SIM Toolkit Framework shall throw an exception.
- CRRN5: The proactive command is sent to the ME as defined and constructed by the toolkit applet without any check of the SIM Toolkit Framework.
- CRRN6: The SIM Toolkit Framework cannot guarantee that if the SET UP IDLE MODE TEXT proactive command is used by a toolkit applet, another toolkit applet will not overwrite this text at a later stage.

6.3.4.1.2 Test Suite Files

Test Script:	FWK_PCS_PCCO_1.scr
Test Applet:	FWK_PCS_PCCO_1.java
	FWK_PCS_PCCO_2.java
	FWK_PCS_PCCO_3.java
Load Script:	FWK_PCS_PCCO_1.ldr
Cleanup Script:	FWK_PCS_PCCO_1.clr
Parameter File:	FWK_PCS_PCCO_1.par

6.3.4.1.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
0	**Applets installation** Applet1 is installed with 4 timers maximum, 0 channel maximum and 1 menu. Applet2 is installed with 8 timers maximum, 3 channels maximum.		

Id	Description	API/Framework Expectation	APDU Expectation
	Applet3 is installed with 1 channel maximum.		
1	**STK Proactive Commands** 1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a SET UP MENU proactive command 3- Applet1 builds and sends a SET UP EVENT LIST proactive command 4- Applet1 builds and sends a POLL INTERVAL proactive command 5- Applet1 builds and sends a POLLING OFF proactive command	1- Applet1 is triggered 2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown	1- 90 00 (no proactive command is sent)
2	**TIMER MANAGEMENT Proactive command** 1- Send a formatted envelope with the TAR of Applet2 2- Applet2 allocates 8 timers by calling allocateTimer() method and release the 3 timers from id 1 to 3. 3- Send a formatted envelope with the TAR of Applet1 4- Applet1 allocates 3 timers (Id 1 to 3) by calling allocateTimer() method 3 times 5- Send a formatted envelope with the TAR of Applet2 6- Applet2 releases timers of Id 4 to 7 7- Send a formatted envelope with the TAR of Applet1 8- For each of the 3 timers allocated by Applet1 (Id 1to 3) a TIMER MANAGEMENT proactive session is performed 9- For other timers (Id 4 to 8), Applet1 builds and sends a TIMER MANAGEMENT proactive command	1- Applet2 is triggered 2- No exception is thrown 3- Applet1 is triggered 4- No exception is thrown 5- Applet2 is triggered 6- No exception is thrown 7- Applet1 is triggered 8- No exception is thrown 9- COMMAND_NOT_ALLOWED toolkit exception is thrown	8- 3 TIMER MANAGEMENT proactive commands are fetched 9- The Status word of the last previous Terminal Response is 90 00 (no more proactive command is sent)
3	**No Channel allowed** 1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command	1- Applet1 is triggered 2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown	1- 90 00 (no proactive command is sent)
4	**4 Channels allowed** 1- Send a formatted envelope with the TAR of Applet3 2- Applet3 builds and sends a CSD OPEN CHANNEL proactive command 3- Send a Fetch and Terminal Response OK on channel 7	1- Applet3 is triggered 2- No exception is thrown	2- 91 1C 3- OPEN CHANNEL proactive

Id	Description	API/Framework Expectation	APDU Expectation
	4- Send a formatted envelope with the TAR of Applet2	4- Applet2 is triggered	
	5- Applet2 builds and sends a CSD OPEN CHANNEL proactive command	5- No exception is thrown	5- 91 1C
	6- Send a Fetch and Terminal Response OK on channel 1		6- OPEN CHANNEL proactive command is fetched
	7- Applet2 builds and sends a GPRS OPEN CHANNEL proactive command	7- No exception is thrown	7- 91 17
	8- Send Fetch and Terminal Response OK on channel 2		8- OPEN CHANNEL proactive command is fetched, SW = 91 1C on the Terminal Response
	9- For each channel id from 3 to 7, Applet2 builds and sends a SEND DATA proactive command	9- COMMAND_NOT_ALLOWED toolkit exception is thrown	
	10- For each channel id from 3 to 7, Applet2 builds and sends a RECEIVE DATA proactive command	10- COMMAND_NOT_ALLOWED toolkit exception is thrown	
	11- For each channel id from 3 to 7, Applet2 builds and sends a CLOSE CHANNEL proactive command	11- COMMAND_NOT_ALLOWED toolkit exception is thrown	
	12- Applet2 builds and sends a CSD OPEN CHANNEL proactive command	12- No exception is thrown	
	13- Fetch and Terminal Response OK on channel 3		13- OPEN CHANNEL proactive command is fetched
	14- Applet2 builds and sends an OPEN CHANNEL proactive command	14- COMMAND_NOT_ALLOWED toolkit exception is thrown	14- 90 00 expected to the previous Terminal Response (no proactive command is sent)
5	**Unknown proactive command**		
	1- Send an envelope menu selection with the item id of Applet1	1- Applet1 is triggered	
	2- Applet1 build an unknown proactive command of 8 null bytes and send it		2- 91 15
	3- Fetch and terminal response OK		3- Command details TLV, Device Identities TLV and unknown TLV including 8 null bytes are fetched.

6.3.4.1.4 Test Coverage

CRR number	Test case number
N1	1
N2	2
N3	3,4
N4	3,4
N5	5
N6	Not testable

6.3.5 Exception Handling

6.3.5.1 Hide Exceptions from the ME

Test Area Reference: FWK_EXH_HEME

6.3.5.1.1 Conformance Requirements

6.3.5.1.1.1 Normal Execution

- CRRN1: A toolkit applet may throw an exception, but this error will not be sent to the ME.

NOTE: Because the behaviour of the SIM is not exactly defined for the above CRRN, there are no tests defined here yet.

6.3.5.1.1.2 Parameters error

No requirements.

6.3.5.1.1.3 Context Errors

No requirements.

6.3.5.2 Interaction with Multiple Triggering

Test Area Reference: FWK_EXH_IMTG

6.3.5.2.1 Conformance Requirements

6.3.5.2.1.1 Normal Execution:

- CRRN1: An exception thrown by a toolkit applet, will not influence toolkit applets registered to the same event.

6.3.5.2.1.2 Parameters error

No requirements.

6.3.5.2.1.3 Context Errors

No requirements.

6.3.5.2.2 Test Suite Files

Test Script:	FWK_EXH_IMTG_1.scr
Test Applet:	FWK_EXH_IMTG_1.java
Load Script:	FWK_EXH_IMTG_1.ldr
Cleanup Script:	FWK_EXH_IMTG_1.clr
Parameter File:	FWK_EXH_IMTG_1.par

6.3.5.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
0	**Load/install 2 toolkit applets registered to EVENT_STATUS_COMMAND, EVENT_PROFILE_DOWNLOAD, EVENT_UNRECOGNIZED_ENVELOPE, EVENT_EVENT_DOWNLOAD_MT_CALL, EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_CB** applet1: Priority= 0x01, applet2: Priority= 0x02, (i.e. applet1 is triggered before applet2)		
1	**Status_Command is sent**	1- Applet1 is triggered : 2- NullPointerException is thrown 3- Applet2 is triggered	
2	**Profile_Download is sent**		

Id	Description	API/Framework Expectation	APDU Expectation
		1- Applet1 is triggered : 2- NullPointerException is thrown 3- Applet2 is triggered	
3	**UNRECOGNIZED_Envelope is sent**	1- Applet1 is triggered : 2- NullPointerException is thrown 3- Applet2 is triggered	
4	**Event_Download_MT_Call is sent**	1- Applet1 is triggered : 2- NullPointerException is thrown 3- Applet2 is triggered	
5	**Unformatted_SMS_PP_Env is sent**	1- Applet1 is triggered : 2- NullPointerException is thrown 3- Applet2 is triggered	
6	**Unformatted_SMS_PP_Upd is sent**	1- Applet1 is triggered : 2- NullPointerException is thrown 3- Applet2 is triggered	
7	**Unformatted_SMS_CB is sent**	1- Applet1 is triggered : 2- NullPointerException is thrown 3- Applet2 is triggered	

6.3.5.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3, 4, 5, 6, 7

6.3.6 Framework Security Management

Security Parameters

The table that follows contains the security parameters that shall be used when the 3GPP TS 23.048 [8] security is required in the test cases developed in the current subclause.

Parameter	Value in hexadecimal
KIC	Value as described in the TS 23.048[8] (recommended value: 15)
KID	Value as described in the TS 23.048[8] (recommended value: 15)
CNTR	00 00 00 00 01
Key for ciphering	Corresponding to KIC (recommended value: 01 41 42 7F DA E8 91 A7 02 41 42 7F DA E8 91 A7)
Key for RC/CC/DS	Corresponding to KID (recommended value: 01 23 45 67 89 AB CD EF EF CD AB 89 67 45 23 01)

If a parameter is not listed explicitly in the above table, the default values of subclause 4.7.3.1 apply.

6.3.6.1 Input Data

Test Area Reference: FWK_FWS_INDA

6.3.6.1.1 Conformance Requirements

6.3.6.1.1.1 Normal Execution

- CRRN1: If the SIM receives an envelope APDU containing an SMS_PP_DATADOWNLOAD BER TLV formatted according to 3GPP TS 23.048 [8], the SIM Toolkit Framework shall verify the security of the SMS TPDU.
- CRRN2: The toolkit applet will only be triggered if the TAR is known and the security verified.
- CRRN3: If the SIM receives an envelope APDU containing an SMS_CB_DATADOWNLOAD formatted according to 3GPP TS 23.048 [8], the SIM Toolkit Framework shall verify the security of the cell broadcast page.
- CRRN4: If the SIM receives an Update Record EFsms instruction formatted according to TS 23.048[8], the SIM Toolkit Framework shall verify the security of the SMS.
- CRRN5: The STF shall provide the input data deciphered.

6.3.6.1.1.2 Parameters error

No requirements.

6.3.6.1.1.3 Context Errors

No requirements.

6.3.6.1.2 Test Area Files

Test Script:	FWK_FWS_INDA_1.scr
Test Applet:	FWK_FWS_INDA_1.java
	FWK_FWS_INDA_2.java
	FWK_FWS_INDA_3.java
	FWK_FWS_INDA_4.java
	FWK_FWS_INDA_5.java
	FWK_FWS_INDA_6.java
Load Script:	FWK_FWS_INDA_1.ldr
Cleanup Script:	FWK_FWS_INDA_1.clr
Parameter File:	FWK_FWS_INDA_1.par

6.3.6.1.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation

Id	Description	API/Framework Expectation	APDU Expectation

1	**Framework checks the Cryptographic checksum and deciphers the data**
	Applet1 is loaded and installed
	1-Envelope(SMS-PP) single and formatted is sent to the SIM with this features:
	Ciphering;
	Cryptographic checksum;
	No proof of receipt;
	TAR of Applet1;
	Data = 01
	2- Short Message concatenated and formatted is sent to the SIM by an Envelope (SMS PP)with these features:
	Ciphering;
	Cryptographic checksum;
	No proof of receipt;
	TAR of Applet1;
	Data length is 150.
		1- Applet1 is triggered and the value integrity is checked.
		2- Applet1 is triggered and the value integrity is checked
			1- The SIM answers to the Envelope with status words 9000
			2- The SIM answers to the Envelope with status words 9000

Id	Description	API/Framework Expectation	APDU Expectation
2	Triggering two different applets with different security Applet2 is installed 1-Envelope(SMS-PP) single and formatted is sent to the SIM with this features: Ciphering; Cryptographic checksum; No proof of receipt; TAR of Applet1 Data = 03 2- Short Message concatenated and formatted is sent to the SIM by an Envelope (SMS PP)with these features: Ciphering; Cryptographic checksum; No proof of receipt; TAR of Applet1 Data length = 150 3-Envelope(SMS-PP) single and formatted is sent to the SIM with this features: No ciphering; No cryptographic checksum; No proof of receipt; TAR of Applet2 Data = 05 4- Short Message concatenated and formatted is sent to the SIM by an Envelope (SMS PP)with these features: No ciphering; No cryptographic checksum; No proof of receipt; TAR of Applet2 Data length = 150.	1- Applet1 is triggered and the value integrity is checked 2- Applet1 is triggered and the value integrity is checked 3- Applet2 is triggered and the value integrity is checked 4- Applet2 is triggered and the value integrity is checked	1- The SIM answers to the Envelope with status words 9000 2- The SIM answers to the Envelope with status words 9000 3- The SIM answers to the Envelope with status words 9000 4- The SIM answers to the Envelope with status words 9000
3	**Envelope(SMS-PP) formatted with wrong cryptographic checksum** 1-Envelope 03.48 single and formatted is sent to the SIM with this features: No ciphering; Wrong cryptographic checksum; No proof of receipt; TAR of Applet1 Data = 07 2- Short Message concatenated and formatted is sent to the SIM by an Envelope (SMS PP)with these features: No ciphering; Wrong cryptographic checksum; No proof of receipt; TAR of Applet1 Data length = 150	1- No applet is triggered. 2- No applet is triggered.	1- The SIM answers to the Envelope with status words 9000

Id	Description	API/Framework Expectation	APDU Expectation
4	**Framework checks the Cryptographic checksum and deciphers the data** Applet3 is loaded and installed 1-Envelope(SMS-CB) formatted is sent to the SIM with this features: Ciphering; Cryptographic checksum; No proof of receipt; Data = 01	1- Applet3 is triggered and the value integrity is checked	1- The SIM answers to the Envelope with status words 9000
5	**Triggering two different applets with different security on Envelope(SMS-CB) formatted** Applet4 is installed 1-Envelope(SMS-CB) formatted is sent to the SIM with this features: Ciphering; Cryptographic checksum; No proof of receipt; TAR of Applet3 Data = 02 2-Envelope(SMS-CB) formatted is sent to the SIM with this features: No ciphering; No cryptographic checksum; No proof of receipt; TAR of Applet4 Data = 03	1- Applet3 is triggered and the value integrity is checked 2- Applet4 is triggered and the value integrity is checked	1- The SIM answers to the Envelope with status words 9000 2- The SIM answers to the Envelope with status words 9000

6	**Envelope(SMS-CB) formatted with wrong cryptographic checksum**
	No ciphering;
	Wrong Cryptographic checksum;
	No proof of receipt;
	TAR of Applet3
	Data = 04
		No applet is triggered	1- The SIM answers to the Envelope with status words 9000

Id	Description	API/Framework Expectation	APDU Expectation

7	**Framework checks the Cryptographic checksum and deciphers the data**
	Applet5 is installed
	1- Short Message single and formatted is sent to the SIM by Update Record EFsms instruction with these features:
	Ciphering;
	Cryptographic checksum;
	No proof of receipt;
	TAR of Applet5;
	Data = 01
	2- Short Message concatenated and formatted is sent to the SIM by Update Record EFsms instruction with these features:
	Ciphering;
	Cryptographic checksum;
	No proof of receipt;
	TAR of Applet5;
	Data length = 150.
		1- Applet5 is triggered and the value integrity is checked.
		2- Applet5 is triggered and the value integrity is checked
			1- The SIM answers to the Update Record EFsms instruction with status words 9000
			2- The SIM answers to the Update Record EFsms instruction with status words 9000

8	**Triggering two different applets with different security** Applet6 is installed 1- Short Message single and formatted is sent to the SIM by Update Record EFsms instruction with these features: Ciphering; Cryptographic checksum; No proof of receipt; TAR of Applet5 Data = 03 2- Short Message concatenated and formatted is sent to the SIM by Update Record EFsms instruction with these features: Ciphering; Cryptographic checksum; No proof of receipt; TAR of Applet5 Data length = 150. 3- Short Message single and formatted is sent to the SIM by Update Record EFsms instruction with these features: No ciphering; No cryptographic checksum; No proof of receipt; TAR of Applet6; Data = 05 4- Short Message concatenated and formatted is sent to the SIM by Update Record EFsms instruction with these features: No ciphering; No cryptographic checksum; No proof of receipt; TAR of Applet6; Data length = 150.	1- Applet5 is triggered and the value integrity is checked. 2- Applet5 is triggered and the value integrity is checked. 3- Applet6 is triggered and the value integrity is checked. 4- Applet6 is triggered and the value integrity is checked.	1- The SIM answers to the Update Record EFsms instruction with status words 9000 2- The SIM answers to the Update Record EFsms instruction with status words 9000 3- The SIM answers to the Update Record EFsms instruction with status words 9000 4- The SIM answers to the Update Record EFsms instruction with status words 9000
9	**Update Record EFsms instruction formatted with wrong cryptographic checksum** 1- Short Message single and formatted is sent to the SIM by Update Record EFsms		

Id	Description	API/Framework Expectation	APDU Expectation
	instruction with these features:No ciphering; Wrong Cryptographic checksum; No proof of receipt; TAR of Applet5 Data = 07 2- Short Message concatenated and formatted is sent to the SIM by Update Record EFsms instruction with these features: No ciphering; Wrong Cryptographic checksum; No proof of receipt; TAR of Applet5 Data length = 150	1- No applet is triggered. 2- No applet is triggered.	1- The SIM answers to the Update Record EFsms instruction with status words 9000 2- The SIM answers to the Update Record EFsms instruction with status words 9000

6.3.6.1.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3
CRRN2	3,6,9
CRRN3	4, 5, 6
CRRN4	7,8,9
CRRN5	1,2,4,5,7,8

6.3.6.2 Output Data

Test Area Reference: FWK_FWS_OUDA

6.3.6.2.1 Conformance Requirements

6.3.6.2.1.1 Normal Execution

- CRRN1: The SIM Toolkit Framework shall secure and send the response packet.

6.3.6.2.1.2 Parameters error

No requirements.

6.3.6.2.1.3 Context Errors

No requirements.

6.3.6.2.2 Test Area Files

Test Script:	FWK_FWS_OUDA_1.scr
Test Applet:	FWK_FWS_OUDA_1.java
Load Script:	FWK_FWS_OUDA_1.ldr
Cleanup Script:	FWK_FWS_OUDA_1.clr
Parameter File:	FWK_FWS_OUDA_1.par

6.3.6.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation

Id	Description	API/Framework Expectation	APDU Expectation
1	**Envelope(SMS-PP) formatted** Ciphering; Cryptographic checksum; proof of receipt response shall be sent using SMS-Deliver-Report; no security applied to proof of receipt Data in plain text = "APPLET1"	The applet is triggered and sends a "Display Text" proactive command with the data received in the Envelope.	The SIM answers to the Envelope with status words 9Fxx and a PoR is retrieved with a GetResponse command. The PoR has no application data. The SIM answers to the Get Response command with status words 91xx to issue a Display Text "APPLET1".
2	**Envelope(SMS-PP) formatted** Ciphering; Cryptographic checksum; proof of receipt response shall be sent using SMS-Deliver-Report; no security applied to proof of receipt Data in plain text = "APPLET1"	The applet posts application data. It does not call the ProactiveHandler.send() method	The SIM answers to the Envelope with status words 9Fxx and a PoR is retrieved with a GetResponse command. The PoR has the application data posted by the application. The SIM answers to the Get Response command with status words 9000.
3	**Envelope(SMS-PP) formatted** Ciphering; Cryptographic checksum; proof of receipt response shall be sent using SMS-Deliver-Report; no security applied to proof of receipt Data in plain text = "TEST"	The applet posts application data and calls the ProactiveHandler.send() method to send a "Display Text" proactive command with the data received in the Envelope.	The SIM answers to the Envelope with status words 9Fxx and a PoR is retrieved with a GetResponse command. The PoR has the application data posted by the application. The SIM answers to the Get Response command with status words 91xx to issue the Display Text "TEST".
4	**Envelope(SMS-PP) formatted** Ciphering; Cryptographic checksum; proof of receipt response shall be sent using SMS-Deliver-Report; proof of receipt shall be ciphered Data in plain text = "TEST"	The applet posts application data and calls the ProactiveHandler.send() method to send a "Display Text" proactive command with the data received in the Envelope.	The SIM answers to the Envelope with status words 9Fxx and a PoR is retrieved with a GetResponse command. The PoR has the application data posted by the application. The SIM answers to the Get Response command with status words 91xx to issue the Display Text "TEST".
5	**Envelope(SMS-PP) formatted** The Terminal Profile command shall be issued with the facility "'9EXX' response code for SIM data download error" enabled The Envelope(SMS-PP) formatted has to be issued with the following features: No ciphering; Wrong Cryptographic checksum; proof of receipt response shall be sent using SMS-Deliver-Report; no security applied to proof of receiptData in plain text = "TEST"	No applet is triggered	The SIM answers to the Envelope with status words 9Exx and a PoR is retrieved with a GetResponse command. The Response Status Code Octet shall be '01'.

6.3.6.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3, 4, 5

6.3.7 Envelope Response Posting

6.3.7.1 EVENT_CALL_CONTROL_BY_SIM

Test Area Reference: FWK_ERP_ECCN

6.3.7.1.1 Conformance Requirements

6.3.7.1.1.1 Normal Execution

- CRRN1: The SIM Toolkit Framework can't reply busy when an Envelope(Call Control) is sent to the SIM.

6.3.7.1.1.2 Parameters error

No requirements.

6.3.7.1.1.3 Context Errors

No requirements.

6.3.7.1.2 Test Area Files

Test Script:	FWK_ERP_ECCN_1.scr
Test Applet:	FWK_ERP_ECCN_1.java
	FWK_ERP_ECCN_2.java
	FWK_ERP_ECCN_3.java
Load Script:	FWK_ERP_ECCN_1.ldr
Cleanup Script:	FWK_ERP_ECCN_1.clr
Parameter File:	FWK_ERP_ECCN_1.par

6.3.7.1.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet1 is registered on the EVENT_CALL_CONTROL_BY_SIM, Applet2 is registered and triggered on the EVENT_MENU_SELECTION.** 1-Applet2 invokes the method send()and no fetch is performed 2-Envelope(Call Control) is sent to the SIM 3-Applet1 calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming dialling number into +11 22 33 44. 4-A Fetch command is sent to the SIM 5-A Terminal Response command is sent to the SIM 6-Delete Applet1 & Applet2 7-Install Applet3	Applet2 is suspended Applet1 is triggered. Applet2's execution shall continue.	The SIM answer 9Fxx to the Envelope(Call Control) The dialling number is retrieved with a GetResponse command. The SIM answers to the Get Response command with status words 91xx.
2	**Applet3 is registered on both the events EVENT_CALL_CONTROL_BY_SIM and EVENT_MENU_SELECTION.** 1-Envelope Menu Selection is sent to the SIM. 2-Applet3 invokes the method send()and no fetch is performed) 3-Envelope(Call Control) is sent to the SIM 4-Applet3 calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming dialling number into +11 22 33 44. 5-A Fetch command is sent to the SIM 6-A Terminal Response command is sent to the SIM	Applet3 is triggered on the EVENT_MENU_SELECTION Applet3 is suspended on the send() method Applet3 is triggered on the EVENT_CALL_CONTROL_BY_SIM. The Applet3's execution shall continue.	The SIM answer 9Fxx to the Envelope(Call Control) The dialling number is retrieved with a GetResponse command. The SIM answers to the Get Response command with status words 91xx.

6.3.7.1.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2

6.3.7.2 EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM

Test Area Reference: FWK_ERP_EMCN

6.3.7.2.1 Conformance Requirements

6.3.7.2.1.1 Normal Execution

- CRRN1: The SIM Toolkit Framework can't reply busy when an Envelope(MO-Short Message Control) is sent to the SIM.

6.3.6.2.1.2 Parameters error

No requirements.

6.3.6.2.1.3 Context Errors

No requirements.

6.3.7.2.2 Test Area Files

Test Script:	FWK_ERP_EMCN_1.scr
Test Applet:	FWK_ERP_EMCN_1.java
	FWK_ERP_EMCN_2.java
	FWK_ERP_EMCN_3.java
Load Script:	FWK_ERP_EMCN_1.ldr
Cleanup Script:	FWK_ERP_EMCN_1.clr
Parameter File:	FWK_ERP_EMCN_1.par

6.3.7.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet1 is registered on the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM; Applet2 is registered and triggered on the EVENT_MENU_SELECTION.** 1-Applet2 invokes the method send()and no fetch is performed) 2-Envelope(MO-SM control) is sent to the SIM 3-Applet1 calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming TP_Destination_Address and any RP_Destination_Address of the Service Center into +11 22 33 44 4-A Fetch command is sent to the SIM 5-A Terminal Response command is sent to the SIM 6-Delete Applet1 & Applet2 7-Install Applet3	Applet2 is suspended Applet1 is triggered. The Applet's execution shall continue.	The SIM answers 9Fxx to the Envelope(MO-Short Message Control) The TP_Destination_Address is retrieved with a GetResponse command. The SIM answers to the Get Response command with status words 91xx.
2	**Applet3 is registered on both the events EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM and EVENT_MENU_SELECTION.** 1-Applet3 invokes the method send()and no fetch is performed) 2-Envelope(MO-SM control) is sent to the SIM 3-Applet3 calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming TP_Destination_Address and any RP_Destination_Address of the Service Center into +11 22 33 44. 4-A Fetch command is sent to the SIM 5-A Terminal Response command is sent to the SIM	Applet3 is suspended on the send() method Applet3 is triggered on the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM. The Applet3's execution shall continue.	The SIM answers 9Fxx to the Envelope(MO-Short Message Control) The TP_Destination_Address is retrieved with a GetResponse command. The SIM answers to the Get Response command with status words 91xx.

6.3.7.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2

6.3.7.3 EVENT_UNRECOGNIZED_ENVELOPE

Test Area Reference: FWK_ERP_EUEN

6.3.7.3.1 Conformance Requirements

6.3.7.3.1.1 Normal Execution

- CRRN1: The EnvelopeResponseHandler is available for the EVENT_UNRECOGNIZED_ENVELOPE.

6.3.7.3.1.2 Parameters error

No requirements.

6.3.7.3.1.3 Context Errors

No requirements.

6.3.7.3.2 Test Area Files

Test Script:	FWK_ERP_EUEN_1.scr
Test Applet:	FWK_ERP_EUEN_1.java
Load Script:	FWK_ERP_EUEN_1.ldr
Cleanup Script:	FWK_ERP_EUEN_1.clr
Parameter File:	FWK_ERP_EUEN_1.par

6.3.7.3.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**An applet triggered on the EVENT_UNRECOGNIZED_ENVELOPE calls the EnvelopeResponseHandler.post() method**	The post() method returns no exception	The SIM answers to the Envelope with status words 9Fxx. The data retrieved with the GetResponse command are the ones posted by the applet.

6.3.7.3.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1

6.3.7.4 EVENT_FORMATTED_SMS_PP_ENV

Test Area Reference: FWK_ERP_EFSE

6.3.7.4.1 Conformance Requirement

6.3.7.4.1.1 Normal Execution

- CRRN1: If PoR is required a SMS-DELIVER REPORT is sent by the SIM, when the post() or the postAsBERTLV() method is invoked and if bit 6 of the second octet of SPI is set to 0.
- CRRN2: If PoR is required a SMS-SUBMIT is sent by the SIM, when the post() or the postAsBERTLV() method is invoked and if bit 6 of the second octet of SPI is set to 1. In this case the statusType method parameter is meaningless. The SIM Toolkit Framework shall build and issue a Send Short Message proactive command as defined in TS 11.14 [4].

6.3.7.4.2 Test Suite Files

Test Script:	FWK_ERP_EFSE_1.scr
Test Applet:	FWK_ERP_EFSE_1.java
	FWK_ERP_EFSE _2.java
Load Script:	FWK_ERP_EFSE _1.ldr
Cleanup Script:	FWK_ERP_EFSE _1.clr
Parameter File:	FWK_ERP_EFSE _1.par

6.3.7.4.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**SMS DELIVER REPORT** 1- A formatted sms pp envelope with SMS Deliver Report required is sent to the SIM with bit 6 of SPI2 set to 0. 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1 3- Applet1 builds the answer and calls the post() method with StatusType=SW1_RP_ACK 4- A formatted sms pp envelope with SMS Deliver Report required is sent to the SIM with bit 6 of SPI2 set to 0. 5- EnvelopeResponseHandler.getTheHandler() method is called by Applet1 6- Applet1 builds the answer and calls the postAsBERTLV() method with StatusType=SW1_RP_ACK	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes 4- Applet1 is triggered 5- No exception is thrown. Applet1 finalizes	3- ME receives 9FXX and checks the response 5- ME receives 9FXX and checks the response
2	**SMS-SUBMIT** 1- A formatted sms pp envelope with SMS Submit required is sent to the SIM with bit 6 of SPI2 set to 1. 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1 3-Applet1 builds the answer and calls the post() method with StatusType=SW1_RP_ACK 4- A formatted sms pp envelope with SMS Submit required is sent to the SIM with bit 6 of SPI2 set to 1. 5- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- No exception is thrown. Applet1 finalizes 4- Applet1 is triggered 5- No exception is thrown	3- ME receives a Send Short Message proactive command.

Id	Description	API/Framework Expectation	APDU Expectation
	6- Applet1 builds the answer and calls the post() method with StatusType=SW1_RP_ERROR	Applet1 finalizes	6- ME receives a Send Short Message proactive command.
	7- A formatted sms pp envelope with SMS Submit required is sent to the SIM with bit 6 of SPI2 set to 1.	7- Applet1 is triggered	
	8- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	8- No exception is thrown.	
	9.-Applet1 builds the answer and calls the postAsBERTLV() method with StatusType=SW1_RP_ACK	Applet1 finalizes	9- ME receives a Send Short Message proactive command.
	10- A formatted sms pp envelope with SMS Submit required is sent to the SIM with bit 6 of SPI2 set to 1.	10- Applet1 is triggered	
	11- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	11- No exception is thrown.	
	12- Applet1 builds the answer and calls the postAsBERTLV () method with StatusType=SW1_RP_ERROR		12- ME receives a Send Short Message proactive command.

6.3.7.4.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1
CRRN2	2

6.3.8 Toolkit Installation

6.3.8.1 Timers Allocation

Test Area Reference: FWK_TIN_TMAL

6.3.8.1.1 Conformance Requirements

6.3.8.1.1.1 Normal execution

- CRRN1: One toolkit applet can register to several timers, but a timer can only be allocated to one toolkit applet.

6.3.8.1.1.2 Parameters error

No requirements.

6.3.8.1.1.3 Context errors

- CRRC1: Allocated timers shall not exceed the maximum number of timers allowed for this applet instance defined during installation.
- CRRC2: The total number of timers allocated for all the applets shall not exceed 8. If the maximum number of timers required is greater than '08' (maximum numbers of timers specified in TS 11.14 [4], the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

6.3.8.1.2 Test suite files

Test Script:	FWK_TIN_TMAL_1.scr
Test Applet:	FWK_TIN_TMAL_1.java
	FWK_TIN_TMAL_2.java
	FWK_TIN_TMAL_3.java
Load Script:	FWK_TIN_TMAL_1.ldr
Cleanup Script:	FWK_TIN_TMAL_1.clr
Parameter File:	FWK_TIN_TMAL_1.par

6.3.8.1.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**More than 8 timers at the instantiation of applet1: check that applet1 is not installed.** `Install for install of applet1 with maximum 9 timers allocated, requesting a PoR to be sent via SMS-DELIVER-REPORT.`		**The SIM answers to the Envelope with status words 9Fxx** **A GET RESPONSE is sent and the additional data in the PoR is checked. It must be 01 6A 80.**
	Reset the card		
2	Good installation of applet2 `Install for install of applet2 (maximum 4 timers allocated).`		The SIM answers to the Envelope with status words 90 00
3	**Allocate 4 timers** **Applet2**	No exception shall be thrown.	
4	**Allocate one more timer** **Applet2**	Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE	
5	Good installation of applet3		

Id	Description	API/Framework Expectation	APDU Expectation
	`Install for install of applet3 (maximum 8 timers allocated).`		The SIM answers to the Envelope with status words 90 00
6	**Allocate 4 timers** **Applet3**	No exception shall be thrown.	
7	**Allocate one more timer** **Applet3**	Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE	
8	**Check that each timerId (allocated by applet2 and applet3) is between 1 and 8 and is different from each other**		

6.3.8.1.4 Test Coverage

CRR number	Test case number
N1	2, 3, 8
C1	1, 7
C2	4, 5, 6

6.3.8.2 Item Identifier

Test Area Reference: FWK_TIN_ITID

6.3.8.2.1 Conformance Requirements

6.3.8.2.1.1 Normal execution

- CRRN1: If the requested item identifier in the range [1-127] is not already allocated, then this item identifier shall be allocated to the current applet.
- CRRN2: If the requested item identifier is '00', the card shall take the first free value in the range [128,255].

6.3.8.2.1.2 Parameters error

- CRRP1: If the requested item identifier is in the range [128,255], then the card shall reject the install command.

6.3.8.2.1.3 Context errors

- CRRC1: If the requested item identifier in the range [1-127] is already allocated, then the card shall reject the install command.

6.3.8.2.2 Test suite files

Test Script:	FWK_TIN_ITID_1.scr
Test Applet:	FWK_TIN_ITID_1.java
	FWK_TIN_ITID_2.java
	FWK_TIN_ITID_3.java
Load Script:	FWK_TIN_ITID_1.ldr
Cleanup Script:	FWK_TIN_ITID_1.clr
Parameter File:	FWK_TIN_ITID_1.par

6.3.8.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation

1	**Bad installation of applet1**
	`Install for install of applet1.The following parameters item Id equal to 128`
	`applet1 is selected`
			applet1 is not found, status word 6X XX

2	**Good installation of applet1**
	`Install for install of applet1. item Id = 1 for the first menu and 127 for the second one`
	`A Terminal Profile is sent to the card with only PROFILE_DOWNLOAD, SMS_PP_DOWNLOAD, MENU_SELECTION, SET_UP_MENU and COMMAND_RESULT facilities.`
			The SIM answers to the Envelope with status words 91xx to send back to the ME the 2 new menus.
			The menus are (position/itemId/text)
			01/01/menu11
			02/127/menu12

3	**Bad installation of applet2**
	`Item identifier already allocated`
	`Install for install of applet2.`
	`item Id = 127`
	`applet2 is selected`
			applet2 is not found, status word 6X XX

4	**Good installation of applet2**
	`Install for install of applet2.`
	`item Id = 0`
			The SIM answers to the Envelope with status words 91xx to send back to the ME the 3 menus.
			The menus are
			01/01/menu11
			02/127/menu12
			03/128/menu21

5	**Good installation of applet3**
	`Install for install of applet3.`
	`item Id = 0`
			The SIM answers to the Envelope with status words 91xx to send back to the ME the 4 menus.
			The menus are
			01/01/menu11
			02/127/menu12
			03/128/menu21
			04/129/menu31

Id	Description	API/Framework Expectation	APDU Expectation

6	**Good delete and installation of applet2**
	`Delete instance of applet2`
	`Perform a RESET and a Terminal Profile with the facilities of PROFILE_DOWNLOAD, SMS-PP_DATA_DOWNLOAD, MENU_SELECTION, COMMAND_RESULT and SET_UP_MENU`
	`Install for install of applet2.`
	`item Id = 0`
			The SIM answers to the Terminal Profile with status words 91xx to send back to the ME the 3 menus.
			The menus are
			01/01/menu11
			02/127/menu12
			03/129/menu31
			The SIM answers to the Envelope with status words 91xx to send back to the ME the 4 menus.
			The menus are
			01/01/menu11
			02/127/menu12
			03/128/menu21
			04/129/menu31

6.3.8.2.4 Test Coverage

CRR number	Test case number
N1	2
N2	4, 5, 6
P1	1
C1	3

6.3.8.3 Item Position

Test Area Reference: FWK_TIN_ITPO

6.3.8.3.1 Conformance Requirements

6.3.8.3.1.1 Normal execution

- CRRN1: The position of the new menu entries is an absolute position among the existing ones.
- CRRN2: If the position identifier is 00h, the menu shall have the last position.

6.3.8.3.1.2 Parameters error

No requirements.

6.3.8.3.1.3 Context errors

No requirements.

6.3.8.3.2 Test suite files

Test Script: FWK_TIN_ITPO_1.scr

Test Applet: FWK_TIN_ITPO_1.java

FWK_TIN_ITPO_2.java

FWK_TIN_ITPO_3.java

Load Script: FWK_TIN_ITPO_1.ldr

Cleanup Script: FWK_TIN_ITPO_1.clr

Parameter File: FWK_TIN_ITPO_1.par

6.3.8.3.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation

1	**Installation of applet1**
	`Perform Install for install of applet1.Position/ItemId`
	`01/01`
	`02/02`
	`A Terminal Profile is sent to the card`
			The menus are (position/itemId/text)
			01/01/menu11
			02/02/menu12

2	**Installation of applet2**
	`Perform Install for install of applet2. Position/ItemId`
	`03/03`
	`04/04`
			The SIM answers to the Envelope with status words 91xx to send back to the ME the 4 menus.
			The menus are (position/itemId/text)
			01/01/menu11
			02/02/menu12
			03/03/menu21
			04/04/menu22

3	**Installation of applet3**
	`Perform Install for install of applet3. Position/ItemId`
	`00/05`
			The SIM answers to the Envelope with status words 91xx to send back to the ME the 5 menus.
			The menus are (position/itemId/text)
			01/01/menu11
			02/02/menu12
			03/03/menu21
			04/04/menu22
			05/05/menu31

6.3.8.3.4 Test Coverage

NOTE: As Item Position management is not fully specified in the 3GPP TS 43.019 [7] or 3GPP TS 23.048 [8] all possible tests cannot be performed.

CRR number	Test case number
N1	1, 2
N2	3

6.3.8.4 Maximum Text Length for a menu entry

Test Area Reference: FWK_TIN_MLME

6.3.8.4.1 Conformance Requirements

6.3.8.4.1.1 Normal execution

- CRRN1: The maximum length of item text string is defined at the installation of the toolkit applet.

6.3.8.4.1.2 Parameters errors

- CRRP1: If initMenuEntry length parameter is greater than the allocated space (Maximum Text Length for a menu entry), then a ToolkitException ALLOWED_LENGTH_EXCEEDED is thrown.
- CRRP2: If changeMenuEntry length parameter is greater than the allocated space (Maximum Text Length for a menu entry), then a ToolkitException ALLOWED_LENGTH_EXCEEDED is thrown.

6.3.8.4.1.3 Context errors

No requirements.

6.3.8.4.2 Test suite files

Test Script:	FWK_TIN_MLME_1.scr
Test Applet:	FWK_TIN_MLME_1.java
Load Script:	FWK_TIN_MLME_1.ldr
Cleanup Script:	FWK_TIN_MLME_1.clr
Parameter File:	FWK_TIN_MLME_1.par

6.3.8.4.3 Test Procedure

Id	Description	API / Framework Expectation	APDU Expectation

1	**Installation of applet with 2 menus not exceeding the maximum text length**
	Install one applet with 2 menu entries allowed and max. text length equal to 10.
	initMenuEntry defined at the install (install) command
	MenuEntry = "MenuEntry1", "MenuEntry2"
	Offset = 0
	Length = 10
	NextAction = '00'
	HelpSupported = false
	IconQualifier = '00'
	IconIdentifier = 0		

2	**initMenuEntry with a too large length**
	initMenuEntry with length equal to 11
	MenuEntry = " MenuEntry03"
	Offset = 0
	Length = 11
	NextAction = '00'
	HelpSupported = false
	IconQualifier = '00'
	IconIdentifier = 0
		ToolkitException ALLOWED_LENGTH_EXCEEDED is thrown	

3	**initMenuEntry with a right length**
	initMenuEntry with length parameter equal to 10
	MenuEntry = " MenuEntry3"
	Offset = 0
	Length = 10
	NextAction = '00'
	HelpSupported = false
	IconQualifier = '00'
	IconIdentifier = 0
			a SET UP MENU (2 items) is issued with TLV item length equal to 11 (Identifier + Text string of item)

Id	Description	API / Framework Expectation	APDU Expectation

4	**changeMenuEntry with a right length**
	`Applet1 is triggered by a EVENT_MENU_SELECTION.`
	`changeMenuEntry of menu 1, with length parameter equal to 10`
	`Id = '01'`
	`MenuEntry = "MenuEntry4"`
	`Offset = 0`
	`Length = menuEntry.length`
	`NextAction = 0`
	`HelpSupported = false`
	`IconQualifier = 0`
	`IconIdentifier = 0`
	`Return from processToolkit`
			a SET UP MENU (2 items) is issued with TLV item length equal to 11 (Identifier + Text string of item)

5	**changeMenuEntry with a too large length**
	`Applet1 is triggered by a EVENT_MENU_SELECTION.`
	`ChangeMenuEntry of menu 1, with length parameter equal to 11`
	`Id = '02'`
	`MenuEntry = "MenuEntry05"`
	`Offset = 0`
	`Length = menuEntry.length`
	`NextAction = 0`
	`HelpSupported = false`
	`IconQualifier = 0`
	`IconIdentifier = 0`
	`Return from processToolkit`
		ToolkitException ALLOWED_LENGTH_EXCEEDED is thrown	Shall not receive a SET UP MENU different from the previous one

6.3.8.4.4 Test Coverage

CRR number	Test case number
CRRN1	1, 3, 4
CRRP1	2
CRRP2	5

6.3.8.5 Maximum number of menu entries

Test Area Reference: FWK_TIN_NBME

6.3.8.5.1 Conformance Requirements

6.3.8.5.1.1 Normal execution

- CRRN1: The maximum number of menu entries is defined at the installation of the toolkit applet and can be the maximum number of successful invocations of the method initMenuEntry .

6.3.8.5.1.2 Parameters errors

- CRRP1: If the menu entry cannot be initialised (e.g. no more item data in applet loading parameter), a ToolkitException with the REGISTRY_ERROR reason code is thrown.

6.3.8.5.1.3 Context errors

No requirements.

6.3.8.5.2 Test suite files

Test Script: FWK_TIN_NBME_1.scr

Test Applet:	FWK_TIN_NBME_1.java
	FWK_TIN_NBME_2.java
Load Script:	FWK_TIN_NBME_1.ldr
Cleanup Script:	FWK_TIN_NBME_1.clr
Parameter File:	FWK_TIN_NBME_1.par

6.3.8.5.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation

1	**Installation of applet with 3 menus**
	Install (install) applet with max. number of menu entry is '3', defined at the install (install) command.
	initMenuEntry for each menu entry allowed (3 times)
	MenuEntry = "menu1", "menu2", "menu3"
	Offset = 0
	Length = 5
	NextAction = '00'
	HelpSupported = false
	IconQualifier = '00'
	IconIdentifier = 0
		No Exception is thrown	

2	**init of a 4th menu**
	initMenuEntry one more time
	MenuEntry = "menu4"
	Offset = 0
	Length = 5
	NextAction = '00'
	HelpSupported = false
	IconQualifier = '00'
	IconIdentifier = 0
		ToolkitException REGISTRY_ERROR is thrown	SET UP MENU (3 items) is issued with TLV item length equal to 6 (Identifier + Text string of item)

3	**Installation of 2nd applet with 0 menu**
	Install (install) another applet, with max. number of menu entry is '0', defined at the install (install) command.
	initMenuEntry once
	MenuEntry = "menu1"
	Offset = 0
	Length = 5
	NextAction = '00'
	HelpSupported = false
	IconQualifier = '00'
	IconIdentifier = 0
		ToolkitException REGISTRY_ERROR is thrown	Shall not receive a SET UP MENU different from the previous one

6.3.8.5.4 Test Coverage

CRR number	Test case number
CRRN1	1
CRRP1	2, 3

6.3.8.6 Access Domain

Test Area Reference: FWK_TIN_ACDO

6.3.8.6.1 Conformance Requirements

6.3.8.6.1.1 Normal execution

- CRRN1: The Access Domain parameter indicates the mechanism used to control the applet instance access to the GSM file System ('00' means full access to the GSM File System, 'FF' means no access to the GSM File System).

6.3.8.6.1.2 Parameters errors

- CRRP1: If the Access Domain Parameter requested is not supported, the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.
- CRRP2: If an applet with Access Domain Parameter 'FF' (i.e. No Access to the GSM File System) tries to access a GSM file (e.g. invoke the updateBinary(..) method) the framework shall throw a SIMViewException with a AC_NOT_FULFILLED reason.

6.3.8.6.1.3 Context errors

No requirements.

6.3.8.6.2 Test suite files

Test Script:	FWK_TIN_ACDO_1.scr
Test Applet:	FWK_TIN_ACDO_1.java
	FWK_TIN_ACDO_2.java
	FWK_TIN_ACDO_3.java
Load Script:	FWK_TIN_ACDO_1.ldr
Cleanup Script:	FWK_TIN_ACDO_1.clr
Parameter File:	FWK_TIN_ACDO_1.par

6.3.8.6.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
0	Install (install) applet1 with: - Length of Access Domain field value is '1' - Access Domain Parameter value is '00' (full access to the GSM File System) Install (install) applet2 with: - Length of Access Domain field value is '1' - Access Domain Parameter value is 'FF' (No access to the GSM File System) Install (install) applet3 with: - Length of Access Domain field value is '1' - Access Domain Parameter value is '00' (full access to the GSM File System)		

Id	Description	API/Framework Expectation	APDU Expectation

1	**readBinary/readRecord method with full Access Domain Parameter**
	1- Select EF-TARU file whose Read access condition is ALWAYS
	Perform the readBinary method:
	fileOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
	2- Select EF-SMS file whose Read access condition is CHV1
	Perform the readRecord method:
	recNumber = 1
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
	3- Select EF-TRAC file whose Read access condition is CHV2
	Perform the readBinary method:
	fileOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
	4- Select EF-SUME file Read access condition is ADM0
	Perform the readBinary method:
	fileOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
	5- Select EF-TNR file whose Read access condition is NEVER
	Perform the readBinary method:
	fileOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
		1 to 4- no exception is thrown
		5- SIMViewException AC_NOT_FULFILLED is thrown	

Id	Description	API/Framework Expectation	APDU Expectation

2	**updateBinary/updateRecord method with full Access Domain Parameter**
	For each case, send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event.
	1- Select EF-TNR file whose Update access condition is ALWAYS
	Perform the updateBinary method:
	fileOffset = 0
	resp = abUpdate[FFFFFF]
	respOffset = 0
	respLength = 3
	2- Select EF-SMS file whose Update access condition is CHV1
	Perform the updateRecord method:
	recNumber = 1
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = 0resp = abUpdate[]
	respOffset = 0
	respLength = 3
	3- Select EF-FDN file whose Update access condition is CHV2
	Perform the updateBinary method:
	recNumber = 1
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = 0
	resp = abUpdate[]
	respOffset = 0
	respLength = 3
	4- Select EF-SUME file Update access condition is ADM0
	Perform the updateBinary method:
	fileOffset = 0
	resp = abUpdate[]
	respOffset = 0
	respLength = 3
	5- Select EF-TNU file whose Update access condition is NEVER
	Perform the updateBinary method:
	fileOffset = 0
	resp = abUpdate[]
	respOffset = 0
	respLength = 3
		1 to 4- no exception is thrown
		5- SIMViewException AC_NOT_FULFILLED is thrown	

3	**invalidate method with full Access Domain Parameter** 1- Select EF-TNR file whose Invalidate access condition is ALWAYS Perform the invalidate method 2- Select EF-TIAC file whose Invalidate access condition is CHV1 Perform the invalidate method 3- Select EF-ADN file whose Invalidate access condition is CHV2 Perform the invalidate method 4- Select EF-SUME file Invalidate access condition is ADM0 Perform the invalidate method 5- Select EF-CNIV file whose Invalidate access condition is NEVER Perform the invalidate method	1 to 4- no exception is thrown 5- SIMViewException AC_NOT_FULFILLED is thrown	

Id	Description	API/Framework Expectation	APDU Expectation
4	**rehabilitate method with full Access Domain Parameter** 1- Select EF-TNR file whose Rehabilitate access condition is ALWAYS Perform the rehabilitate method 2- Select EF-IMSI file whose Rehabilitate access condition is CHV1 Perform the rehabilitate method 3- Select EF-ADN file whose Rehabilitate access condition is CHV2 Perform the rehabilitate method 4- Select EF-SUME file Rehabilitate access condition is ADM0 Perform the rehabilitate method 5- Select EF-CNRI file whose Rehabilitate access condition is NEVER Perform the rehabilitate method	1 to 4- no exception is thrown 5- SIMViewException AC_NOT_FULFILLED is thrown	

5	**increase method with full Access Domain Parameter**
	1- Select EF-CNR file whose Increase access condition is ALWAYS
	Perform the increase method:
	incr = abIncreaseValue[]
	incrOffset = 0
	resp = abRead[]
	respOffset = 0
	2- Select EF-ACM file whose Increase access condition is CHV1
	Perform the increase method:
	incr = abIncreaseValue[]
	incrOffset = 0
	resp = abRead[]
	respOffset = 0
	3- Select EF-CIAC file whose Increase access condition is CHV2
	Perform the increase method:
	incr = abIncreaseValue[]
	incrOffset = 0
	resp = abRead[]
	respOffset = 0
	4- Select EF-CIAA file Increase access condition is ADM0
	Perform the increase method:
	incr = abIncreaseValue[]
	incrOffset = 0
	resp = abRead[]
	respOffset = 0
	5- Select EF-CNU file whose Increase access condition is NEVER
	Perform the increase method
		1 to 4- no exception is thrown
		5- SIMViewException AC_NOT_FULFILLED is thrown	

Id	Description	API/Framework Expectation	APDU Expectation

6	**readBinary method with no Access Domain Parameter**
	`Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Select EF-TARU file whose Read access condition is ALWAYS`
	`Perform the readBinary method:`
	`fileOffset = 0`
	`resp = abRead[]`
	`respOffset = 0`
	`respLength = 3`
	`t`
		SIMViewException AC_NOT_FULFILLED is thrown	

7	**updateRecord method with no Access Domain Parameter**
	`Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Select EF-SMS file whose Update access condition is CHV1`
	`Perform the updateRecord method:`
	`fileOffset = 0`
	`resp = abUpdate[]`
	`respOffset = 0`
	`respLength = 3`
		SIMViewException AC_NOT_FULFILLED is thrown	

8	**invalidate method with no Access Domain Parameter**
	`Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Select EF-ADN file whose Invalidate access condition is CHV2`
	`Perform the invalidate method`
		SIMViewException AC_NOT_FULFILLED is thrown	

9	**rehabilitate method with no Access Domain Parameter**
	`Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Select EF-SUME file Rehabilitate access condition is ADM0`
	`Perform the rehabilitate method`
		SIMViewException AC_NOT_FULFILLED is thrown	

10	**increase method with no Access Domain Parameter**
	`Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Select EF-CNR file whose Increase access condition is NEVER`
	`Perform the increase method`
		SIMViewException AC_NOT_FULFILLED is thrown
		Applet2 finalizes
		Applet3 restore EF-SUME	

6.3.8.6.4 Test Coverage

NOTE: As Item Position management is not fully specified in the 3GPP TS 43.019 [7] or 3GPP TS 23.048 [8] all possible tests cannot be performed.

CRR number	Test case number
CRRN1	1, 2, 3, 4, 5
CRRP1	Not tested
CRRP2	6, 7, 8, 9, 10

6.3.8.7 Priority Level

Test Area Reference: FWK_TIN_PRLV

6.3.8.7.1 Conformance Requirements

6.3.8.7.1.1 Normal execution

- CRRN1: The priority specifies the order of activation of an applet compared to the other applet registered to the same event ('01': Highest priority level, 'FF' : Lowest priority level).
- CRRN2: If two or more applets are registered to the same event and have the same priority level, the applets are activated according to their installation date (i.e. the most recent applet is activated first).

6.3.8.7.1.2 Parameters errors

No requirements.

6.3.8.7.1.3 Context errors

No requirements.

6.3.8.7.2 Test suite files

Test Script:	FWK_TIN_PRLV_x.scr, x from 1 to 12
Test Applet:	FWK_TIN_PRLV_x.java, x from 1 to 12, 8A, 8B, 9A, 9B, 10A, 10B
Load Script:	FWK_TIN_PRLV_x.ldr, x from 1 to 12
Cleanup Script:	FWK_TIN_PRLV_x.clr, x from 1 to 12
Parameter File:	FWK_TIN_PRLV_x.par, x from 1 to 12, 8A, 8B, 9A, 9B, 10A, 10B

6.3.8.7.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
0	All applets are registered on an EVENT_UNFORMATTED_SMS_PP_ENV event		
1	**Trigger 2 applets with 2 different maximum Priority Levels** Install (install) applet1 with priority level '2' and applet2 with priority level '1', from package fwk_tin_prlv_1. Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Delete applets instances and packages	A static variable is used to validate triggering order: applet2 is triggered before applet1	

Id	Description	API/Framework Expectation	APDU Expectation

2	**Trigger 2 applets with 2 different maximum Priority Levels**
	`Install (install) applet1 with priority level '1' and applet2 with priority level '2', from package fwk_tin_prlv_2.`
	`Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Delete applets instances and packages`
		A static variable is used to validate triggering order: applet1 is triggered before applet2.	

3	**Trigger 2 applets with 2 different Priority Levels**
	`Install (install) applet1 with priority level '80' and applet2 with priority level '7F', from package fwk_tin_prlv_3.`
	`Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Delete applets instances and packages`
		A static variable is used to validate triggering order: applet2 is triggered before applet1	

4	**Trigger 2 applets with 2 different Priority Levels**
	`Install (install) applet1 with priority level '7F' and applet2 with priority level '80', from package fwk_tin_prlv_4.`
	`Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Delete applets instances and packages`
		A static variable is used to validate triggering order: applet2 is triggered before applet1	

5	**Trigger 3 applets with the same Priority Level**
	`Install (install) applet 1, 2, 3 in this order with same priority level from package fwk_tin_prlv_5.`
	`Send an Envelope that triggers the 3 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Delete applets instances and packages.`
		A static variable is used to validate triggering order: applet3 is triggered before applet2, and applet2 is triggered before applet1.	

6	**Trigger 2 applets from 2 classes, with 2 different Priority Level**
	`Install (install) applet1 from class A with priority level '2'`
	`Install (install) applet2 from class B with priority level '1'`
	`Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Delete applets instances and packages`
		A static variable is used to validate triggering order: applet2 is triggered before applet1	

Id	Description	API/Framework Expectation	APDU Expectation
7	**Trigger 2 applets from 2 classes, with the same Priority Level** Install (install) applet1 from class A with priority level '1' Install (install) applet2 from class B with priority level '1' Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Delete applets instances and packages	A static variable is used to validate triggering order: applet2 is triggered before applet1	
8	**Trigger 2 applets from 2 packages, with 2 different Priority Level** Install package fwk_tin_prlv_8. Install (install) applet1 from package fwk_tin_prlv_8A with priority level '2' Install (install) applet2 from package fwk_tin_prlv_8B with priority level '1' Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Delete applets instances ad packages	A static variable is used to validate triggering order: applet2 is triggered before applet1	
9	**Trigger 2 applets from 2 packages, with the same Priority Level** Install package fwk_tin_prlv_9. Install (install) applets 1 from package fwk_tin_prlv_9A and applet2 from package fwk_tin_prlv_9B in this order, with same priority level Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Delete applets instances and packages	A static variable is used to validate triggering order: applet2 is triggered before applet1	

Id	Description	API/Framework Expectation	APDU Expectation
10	**Trigger 4 applets from 2 packages** 1-Install packages fwk_tin_prlv_10, fwk_tin_prlv_10A and fwk_tin_prlv_10B. Install (install) 2 applets 1 then 2 from package fwk_tin_prlv_10A, with respectively priority levels 1 and 2. Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. 2- Install (install) 2 applets 3 then 4 from package fwk_tin_prlv_10B, with respectively priority levels 1 and 2. Send an Envelope that triggers the 4 applets. Delete applets instances and packages	1- A static variable is used to validate triggering order: applet1 is triggered before applet2 2- Applet3 is triggered before applets 1, 4, then 2.	
11	**Trigger 4 applets with the same Priority Level then delete them one after another and trigger them each time** 1- Install (install) applet1, 2, 3, 4 in this order with same priority level from package fwk_tin_prlv_11. Send an Enveloppe that triggers the 4 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Delete applet instance 4 2- Send an Enveloppe that triggers the 3 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Delete applet instance 3 3- Send an Enveloppe that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Delete remaining applet instances and packages	1- A static variable is used to validate triggering order: applets are triggered in order 4, 3, 2, 1. 2- Applets are triggered in order 3, 2, 1. 3- Applets are triggered in order 2, 1.	

Id	Description	API/Framework Expectation	APDU Expectation

12	**Trigger 5 applets with different Priority Levels, alternating install and delete**
	`1- Install (install) applets 1, 2, 3, 4 in this order with respective priority levels 1, 2, 1, 2`
	`Send an Enveloppe that triggers the 4 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event.`		

		1- A static variable is used to validate triggering order: applets are triggered in order 3, 1, 4, 2	

	`2- Delete applet instance 1 and install (install) applet5 with priority level 2`
	`Send an Enveloppe that triggers the 4 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event.`		

		2- Applets are triggered in order 3, 5, 4, 2	

	`3- Re-install (install) applet1 with priority level 1`
	`Send an Enveloppe that triggers the 5 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event.`		

		3- Applets are triggered in order 1, 3, 5, 4, 2	

6.3.8.7.4 Test Coverage

CRR number	Test case number
CRRN1	1, 2, 3, 4, 6, 8, 10, 12
CRRN2	5, 7, 9, 11

6.3.8.8 Channel Allocation

Test Area Reference: FWK_TIN_CHAL

6.3.8.8.1 Conformance Requirements

6.3.8.8.1.1 Normal execution

- CRRN1: One toolkit applet can register to several channels, but a channel can only be allocated to one toolkit applet.

6.3.8.8.1.2 Context errors

- CRRC1 : Allocated channels shall not exceed the maximum number of channels allowed for this applet instance.
- CRRC2 : The total number of channels allocated for all the applets shall not exceed 7. If the maximum number of channels required is greater than '07' (maximum numbers of channels specified in TS 11.14 [4]), the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

6.3.8.8.2 Test suite files

Test Script: FWK_TIN_CHAL_1.scr

Test Applet: FWK_TIN_CHAL_1.java

FWK_TIN_CHAL_2.java

FWK_TIN_CHAL_3.java

Load Script: FWK_TIN_CHAL_1.ldr

Cleanup Script: FWK_TIN_CHAL_1.clr

Parameter File: FWK_TIN_CHAL_1.par

6.3.8.8.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation

1	**More than 7 channels at the instantiation of applet1: check that applet1 is not installed**
	`1-Install for install of applet1 with maximum 8 channels allocated.`
	`A PoR is asked to be sent via SMS-DELIVER-REPORT.`
			1- The SIM answers to the Envelope with status words 9Fxx.
			A GET RESPONSE is sent and the additional data in the PoR is checked. It must be 01 6A 80.

	Reset the card		
2	**Good installation of applet2** `Install for install of applet2 (maximum 4 channels allocated).`		The SIM answers to the Envelope with status words 90 00
3	**Open 4 channels Applet2** `Applet2 builds a proactive command OPEN CHANNEL 4 times, calling init() and send() methods.`	No exception shall be thrown.	OPEN CHANNEL proactive command are fetched. Successful TERMINAL RESPONSE of OPEN CHANNEL are sent to the SIM with Channel Id = 01 to 04
4	**Open one more channel Applet2** `Applet2 builds a proactive command OPEN CHANNEL once again, calling init() and send() methods.`	Shall throw a ToolkitException with reason COMMAND_NOT_ALLOWED	
5	**Good installation of applet3** `Install for install of applet3 (maximum 7 channels allocated).`		The SIM answers to the Envelope with status words 90 00
6	**Open 3 channels Applet3** `Applet3 builds a proactive command OPEN CHANNEL 3 times, calling init() and send() methods.`	No exception shall be thrown.	OPEN CHANNEL proactive command is fetched. Successful TERMINAL RESPONSE of OPEN CHANNEL are sent to the SIM with Channel Id from 05 to 07
7	**Open one more channel Applet3** `Applet3 builds a proactive command OPEN CHANNEL once again, calling init() and send() methods.`	No exception shall be thrown.	OPEN CHANNEL proactive command is fetched. Unsuccessful Terminal Response is sent to the SIM with "No Channel Available" as Additional Information on Result.

6.3.8.8.4 Test Coverage

CRR number	Test case number
N1	2,3
C1	1, 7
C2	4,5,6

6.3.8.9 Minimum Security Level

Test Area Reference: FWK_TIN_MSL

6.3.8.9.1 Conformance Requirements

6.3.8.9.1.1 Normal execution

- CRRN1: The Receiving Entity shall check the Minimum Security Level during processing the security of the Command Packet.
- CRRN2: The Receiving Entity shall reject the message if the MSL check fails.
- CRRN3: If the MSL check fails, a Response Packet with the 'Insufficient Security Level' Response Status Code shall be sent if required.
- CRRN4: If the length of the Minimum Security Level field is greater than zero, the Minimum Security Level is used to specify the minimum level of security to be applied to Secured Packets. The first byte shall be the MSL Parameter, other bytes shall be the MSL Data.
- CRRN5: If the length of the Minimum Security Level field is zero, no minimum security level check shall be performed by the receiving entity.
- CRRN6: If no Minimum Security Level field is present (no MSL length, no MSL parameter and no MSL data), no minimum security level check shall be performed by the receiving entity.
- CRRN7: If the Maximum number of channels field is included in the command data then the Length of Minimum Security Level field shall also be included.
- CRRN8: If an optional parameter is included, then all the previous parameters shall be included also

6.3.8.9.2 Test suite files

Test Script:	FWK_TIN_MSL_1.scr
Test Applet:	FWK_TIN_MSL_1.java
Load Script:	FWK_TIN_MSL_1.ldr
Cleanup Script:	FWK_TIN_MSL_1.clr
Parameter File:	FWK_TIN_MSL_1.par

6.3.8.9.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Installation with MSL length of 0** 1- Install (install) applet with a MSL length = 0 2- Send formatted SMS PP env with no RC/CC/DS, no Ciphering and counter mode 0 (not checked) 3- Send a formatted SMS PP env with CC, ciphering and counter mode 1 (counter available and no checking) 4- Delete the applet instance	2- Applet is triggered 3- Applet is triggered	1- 9000
2	**Installation without MSL field** 1- Install (install) applet without MSL field (no MSL length, no MSL parameter and no data) 2- Send formatted SMS PP env with no RC/CC/DS, no Ciphering and counter mode 0 (not checked) 3- Send a formatted SMS PP env with CC, ciphering and counter mode 1 counter available and no checking) 4- Delete the applet instance	2- Applet is triggered 3- Applet is triggered	1- 9000

6.3.8.9.4 Test Coverage

CRR number	Test case number
CRRN1	Not applicable
CRRN2	Not applicable
CRRN3	Not applicable
CRRN4	Not applicable
CRRN5	1
CRRN6	2
CRRN7	Not testable
CRRN8	Not testable

6.3.9 File System Context

6.3.9.1 Initial Context

Test Area Reference: FWK_FSC_INIT

6.3.9.1.1 Conformance Requirements

6.3.9.1.1.1 Normal Execution

- CRRN1: At the invocation of the processToolkit method of a toolkit applet, the current file is the MF.

6.3.9.1.1.2 Parameters errors

No requirements.

6.3.9.1.1.3 Context errors

No requirements.

6.3.9.1.2 Test Suite Files

Test Script: FWK_FSC_INIT_1.scr

Test Applet:	FWK_FSC_INIT_1.java
	FWK_FSC_INIT_2.java
Load Script:	FWK_FSC_INIT_1.ldr
Cleanup Script:	FWK_FSC_INIT_1.clr
	FWK_FSC_INIT_2.clr
Parameter File:	FWK_FSC_INIT_1.par

6.3.9.1.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored.	

1	**MF is the selected DF in processToolkit()**
	An ENVELOPE APDU containing a formatted SMS PP for Applet 1 is issued to the SIM
	byte[] fci = new byte[10]
	fciOffset = 0
	fciLength = 7
	status()
		No exception shall be thrown.
		Shall return 7.
		fci shall contain the following part of the FCI structure:
		< XX XX XX XX 3F 00 01 >	

2	**No EF is selected** rehabilitate ()	SIMView exception shall be thrown with reason NO_EF_SELECTED	
3	**MF is selected even when an applet triggered before selected any other file...** Applets 1 and 2 register to EVENT_DOWNLOAD_USER_ACTIVITY. Applet 1 has higher priority than Applet 2. An ENVELOPE "EVENT - USER ACTIVITY" is sent to the SIM 1 - Applet 1: - is triggered by event_event_download_user_activity - selects DF_GSM and EF_IMSI 2 - Applet 2: - is triggered by event_event_download_user_activity fciOffset = 0 fciLength = 7 status() 3 - rehabilitate ()	1 - No exception shall be thrown. 2 - No exception shall be thrown. Shall return 7. fci shall contain the following part of the FCI structure: < XX XX XX XX 3F 00 01 > 3 - SIMView exception shall be thrown with reason NO_EF_SELECTED	

6.3.9.1.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3

6.3.9.2 Context Preservation (current file)

Test Area Reference: FWK_FSC_CUFI

6.3.9.2.1 Conformance Requirements

6.3.9.2.1.1 Normal execution

- CRRN1: When calling the method select (), the current files (file context) of any other applets shall not be changed (see 3GPP TS 43.019 [7] - subclause 5.2).

- CRRN2: The select() methods select a file without changing the current file of any other applet or of the subscriber session.

- CRRN3: After invocation of ProactiveHandler.send() method: the current file context of the toolkit applet is unchanged (see 3GPP TS 43.019 [7] - subclause 5.2.).

6.3.9.2.1.2 Parameters errors

No requirements.

6.3.9.2.1.3 Context errors

No requirements.

6.3.9.2.2 Test Suite Files

Test Script:	FWK_FSC_CUFI_1.scr
Test Applet:	FWK_FSC_CUFI_1.java
	FWK_FSC_CUFI_2.java
Load Script:	FWK_FSC_CUFI_1.ldr
Cleanup Script:	FWK_FSC_CUFI_1.clr
	FWK_FSC_CUFI_2.clr
Parameter File:	FWK_FSC_CUFI_1.par

6.3.9.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored.	
1	**No change to file context by another applet** Applet1 registers to EVENT_FORMATTED_SMS_PP_ENV. Applet2 registers to EVENT_CALL_CONTROL_BY_SIM 1 - Applet 1: - is triggered by a formatted SMS - selects DF_SIMTEST and EF_TARU - fileOffset = 0; dataLength = 2; dataOffset = 0; - buffer = {0xCA, 0xFE } - updateBinary (): first 2 bytes of EF_TARU are written as 'CA FE'. - issues a proactive command "Get Inkey". 2 - An ENVELOPE APDU containing a CALL CONTROL BY SIM is issued to the SIM Applet 2: - is triggered by a CALL CONTROL BY SIM - selects DF_TELECOM and EF_ADN. 3 - The terminal response for Get Inkey reactivates Applet 1: - fileOffset = 0; respLength = 2; respOffset = 0; - readBinary () info buffer2	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - No exception shall be thrown. The value of buffer2 is { 0xCA, 0xFE }	A GET INKEY proactive command is fetched from the SIM

Id	Description	API/Framework Expectation	APDU Expectation

2	**No change to file context by subscriber session**
	`1 - Applet 1`
	`- issues a proactive command "Get Inkey".`
	`2 - Subscriber session selects DF_TELECOM and EF_ADN.`
	`3 - The terminal response for Get Inkey reactivates Applet 1:`
	`- fileOffset = 0; respLength = 2; respOffset = 0;`
	`- readBinary () info buffer2`
		1 - No exception shall be thrown.
		3 - No exception shall be thrown. The value of buffer2 is { 0xCA, 0xFE }
			1 - A GET INKEY proactive command is fetched from the SIM

3	**No change by applet of subscriber session context**
	`1 - Applet 1:`
	`- selects DF_SIMTEST and EF_TNU`
	`- issues a proactive command "Get Inkey".`
	`2 - subscriber session reads record 1 of current file (shall be EF_ADN)`
	`3 - The terminal response for Get Inkey reactivates Applet 1, which terminates execution`
		1 - No exception shall be thrown.
		3 - No exception shall be thrown.
			1 - A GET INKEY proactive command is fetched from the SIM
			2 - READ RECORD absolute number 1 shall read "FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF" (from EF_{ADN})

6.3.9.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1
CRRN2	1, 2, 3
CRRN3	1,2

6.3.9.3 Context Preservation (current record pointer)

Test Area Reference: FWK_FSC_CURE

6.3.9.3.1 Conformance Requirements

6.3.9.3.1.1 Normal execution

- CRRN1: When the seek method is called by one applet, the record pointer of any other applet is not changed.
- CRRN2: *updateRecord*: the current record pointer of other applets / subscriber shall not be changed in case of linear fixed EF
- CRRN3: *updateRecord*: the record pointer of a cyclic EF shall be changed for all other applets / subscriber to the record number 1.
- CRRN4: *readRecord:* read data bytes of the linear fixed or cyclic EF currently selected by the applet without changing the current record pointer of any other applet / subscriber.
- CRRN5: *increase*: the last updated record of the cyclic EF currently selected becomes record number 1 for every other applet and subscriber session.

6.3.9.3.1.2 Parameters errors

No requirements.

6.3.9.3.1.3 Context errors

No requirements.

6.3.9.3.2 Test Suite Files

Test Script:	FWK_FSC_CURE_1.scr
Test Applet:	FWK_FSC_CURE_1.java
	FWK_FSC_CURE_2.java
Load Script:	FWK_FSC_CURE_1.ldr
Cleanup Script:	FWK_FSC_CURE_1.clr
	FWK_FSC_CURE_2.clr
Parameter File:	FWK_FSC_CURE_1.par

6.3.9.3.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
0	**SIM Initialization**	Responses ignored.	
1	**Seek without affecting another record pointer** Applet1 registers to EVENT_FORMATTED_SMS_PP_ENV Applet 2 registers to EVENT_CALL_CONTROL_BY_SIM 1 - Applet 1: - is triggered by a formatted SMS event - selects DF_SIMTEST and EF_LARU - reads record 2 using NEXT so that the current record pointer is set to record 2 - issues a proactive command, e.g. Get Inkey. 2 - An ENVELOPE APDU containing a CALL CONTROL BY SIM is issued to the SIM Applet 2: - is triggered by a CALL CONTROL event - selects DF_SIMTEST and EF_LARU - performs a seek of pattern {0x55} from beginning forward, which finds record 1. - returns from processToolkit 3 - The terminal response for Get Inkey reactivates Applet 1: - call readRecord() using CURRENT - the record read should still be record 2 of EF_LARU, containing {0xAA, 0xAA, 0xAA, 0xAA}	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - No exception shall be thrown.	1 - A GET INKEY proactive command is fetched from the SIM

Id	Description	API/Framework Expectation	APDU Expectation
2	**updateRecord in linear fixed EF without affecting current pointer of others** 1 - Applet 1: - is triggered by a formatted SMS event - selects DF_SIMTEST and EF_LARU - reads record 2 using NEXT so that the current record pointer is set to record 2 - issues a proactive command, e.g. Get Inkey. 2 - An ENVELOPE APDU containing a CALL CONTROL BY SIM is issued to the SIM Applet 2: - is triggered by a CALL CONTROL BY SIM event - selects DF_SIMTEST and EF_LARU - updates record 1, by using mode "NEXT". - returns from processToolkit 3 - The terminal response for Get Inkey reactivates Applet 1: - call readRecord() using CURRENT - the record read should still be record 2 of EF_LARU, containing {0xAA, 0xAA, 0xAA, 0xAA}	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - No exception shall be thrown.	1 - A GET INKEY proactive command is fetched from the SIM
3	**readRecord in linear fixed EF without affecting current pointer of others** 1 - Applet 1: - is triggered by a formatted SMS event - selects DF_SIMTEST and EF_LARU - reads record 2 using NEXT so that the current record pointer is set to record 2 - issues a proactive command, e.g. Get Inkey. 2 - An ENVELOPE APDU containing a CALL CONTROL BY SIM is issued to the SIM Applet 2: - is triggered by a CALL CONTROL BY SIM event - selects DF_SIMTEST and EF_LARU - reads record 1, by using mode "NEXT". - returns from processToolkit 3 - The terminal response for Get Inkey reactivates Applet 1: - call readRecord() using CURRENT - the record read should still be record 2 of EF_LARU, containing {0xAA, 0xAA, 0xAA, 0xAA}	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - No exception shall be thrown.	1 - A GET INKEY proactive command is fetched from the SIM

6.3.9.3.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1
CRRN2	2
CRRN3	not tested (see note)
CRRN4	3
CRRN5	not tested (see note)
NOTE: These requirements have not been tested because of an inconsistent behaviour in 3GPP TS 43.019 [7], which is foreseen to be corrected in future releases.	

6.3.10 Other parts transferred to framework from API

6.3.10.1 A handler is a temporary JCRE Entry Point object

Test Area Reference: FWK_API_HEPO

6.3.10.1.1 Conformance Requirement:

6.3.10.1.1.1 Normal execution

- CRRN1: The EnvelopeHandler is a Temporary JCRE Entry Point Object (see Javacard 2.1 Runtime Environment (JCRE) Specification [12]).
- CRRN2: The EnvelopeResponseHandler is a Temporary JCRE Entry Point Object (see Javacard 2.1 Runtime Environment (JCRE) Specification [12]).
- CRRN3: The ProactiveHandler is a Temporary JCRE Entry Point Object (see Javacard 2.1 Runtime Environment (JCRE) Specification [12]).
- CRRN4: The ProactiveResponseHandler is a Temporary JCRE Entry Point Object (see Javacard 2.1 Runtime Environment (JCRE) Specification [12]).

6.3.10.1.1.2 Parameters errors

No requirements.

6.3.10.1.1.3 Context errors

No requirements.

6.3.10.1.2 Test suite files

Test Script:	FWK_API_HEPO_1.scr
Test Applet:	FWK_API_HEPO_1.java
Load Script:	FWK_API_HEPO_1.ldr
Cleanup Script:	FWK_API_HEPO_1.clr
Parameter File:	FWK_API_HEPO_1.par

6.3.10.1.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**EnvelopeHandler.getTheHandler and store it in a static field of the toolkit applet**	SecurityException is thrown	
2	**EnvelopeHandler.getTheHandler and store it in a field of the toolkit applet**	SecurityException is thrown	
3	**EnvelopeResponseHandler.getTheHandler and store it in a static field of the toolkit applet**	SecurityException is thrown	
4	**EnvelopeResponseHandler.getTheHandler and store it in a field of the toolkit applet**	SecurityException is thrown	
5	**ProactiveHandler.getTheHandler and store it in a static field of the toolkit applet**	SecurityException is thrown	
6	**ProactiveHandler.getTheHandler and store it in a field of the toolkit applet**	SecurityException is thrown	
7	**Build and send a DISPLAY TEXT command to be able to get the reference of the ProactiveReponseHandler** **ProactiveResponseHandler.getTheHandler and store it in a static field of the toolkit applet**	 SecurityException is thrown	Proactive command fetched and terminal response is issued

Id	Description	API/Framework Expectation	APDU Expectation
8	**ProactiveResponseHandler.getTheHandler and store it in a field of the toolkit applet**	SecurityException is thrown	

6.3.10.1.4 Test Coverage

CRR number	Test case number
N1	1, 2
N2	3, 4
N3	5, 6
N4	7, 8

6.3.10.2 Transaction

Test Area Reference: FWK_API_TRAN

6.3.10.2.1 Conformance Requirement:

6.3.10.2.1.1 Normal execution

- CRRN1: A pending toolkit applet transaction at the ProactiveHandler.send() method invocation is aborted.

6.3.10.2.1.2 Parameters errors

No requirements.

6.3.10.2.1.3 Context errors

No requirements.

6.3.10.2.2 Test suite files

Test Script:	FWK_API_TRAN_1.scr
Test Applet:	FWK_API_TRAN_1.java
Load Script:	FWK_API_TRAN_1.ldr
Cleanup Script:	FWK_API_TRAN_1.clr
Parameter File:	FWK_API_TRAN_1.par

6.3.10.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Verify that transaction is aborted when a proactive command is sent** Initialise a byte field with 0x05 Build a display text proactive command. beginTransaction() Update the byte with 0x02 send the proactive command Verify that the byte value is 0x05 JCSystem.getTransactionDepth()	`Shall return 0`	Proactive command fetched and terminal response is issued

6.3.10.2.4 Test Coverage

CRR number	Test case number
N1	1

6.3.10.3 Timer Id between Applets

Test Area Reference: FWK_API_TMID

6.3.10.3.1 Conformance Requirement:

6.3.10.3.1.1 Normal execution

No requirements.

6.3.10.3.1.2 Parameters errors

No requirements.

6.3.10.3.1.3 Context errors

- CRRC1: The method ToolkitRegistry.releaseTimer() shall throw a ToolkitException with INVALID_TIMER_ID reason if the timer is valid but isn't allocated to this applet.

6.3.10.3.2 Test suite files

Test Script:	FWK_API_TMID_1.scr
Test Applet:	FWK_API_TMID_1.java
Load Script:	FWK_API_TMID_1.ldr
Cleanup Script:	FWK_API_TMID_1.clr
Parameter File:	FWK_API_TMID_1.par

6.3.10.3.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**During installation :** **First instance allocate a timer and store the returned value in a static field.** **Second instance allocate a timer.** **Trig second instance and try to releaseTimer() with the static field value.**	**releaseTimer() shall throw a ToolkitException with INVALID_TIMER_ID reason**	

6.3.10.3.4 Test Coverage

CRR number	Test case number
N1	1

6.3.11 Concatenated SMS

6.3.11.1 Concatenation processing

6.3.11.1.1 Conformance Requirements

6.3.11.1.1.1 Normal execution

- CRRN1: The SIM Toolkit Framework shall link single Short Messages together to re-assemble the original message before any further processing.
- CRRN2: The concatenation control headers used to re-assemble the short messages in the correct order shall not be present in the SMS TPDU.
- CRRN3: The TP-elements of the SMS TPDU and the Address (TS-Service-Centre-Address) shall correspond to the ones in the last received Short Message (independently of the Sequence number of Information-Element-Data).
- CRRN4: The original Short Message shall be placed in one SMS TPDU TLV (with TP-UDL field coded on one octet) included in the EnvelopeHandler.
- CRRN5: The SIM Toolkit Framework shall be able to process messages with the following properties:

  - The Information Element Identifier is equal to the 8-bit reference number

  - It contains uncompressed 8 bit data or uncompressed UCS2 data.

6.3.11.2 Test Suite Files

Test Script:	FWK_CSM_PROC_1.scr
Test Applet:	FWK_CSM_PROC_1.java
Load Script:	FWK_CSM_PROC_1.ldr
Cleanup Script:	FWK_CSM_PROC_1.clr
Parameter File:	FWK_CSM_PROC_1.par

6.3.11.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
	Applet registration to EVENT_FORMATTED_SMS_PP_ENV and triggering Applet is registered to EVENT_FORMATTED_SMS_PP_ENV and EVENT_UNFORMATTED_SMS_PP_ENV A concatenated formatted SMS_PP short message is sent to the SIM (composed of two segments).		
1	The second segment of a concatenated short message is sent to the SIM.	Applet is not triggered.	
2	The first segment of the concatenated short message is sent to the SIM	Applet is triggered.	
3	Call the EnvelopeHanlder.getTheHandler()	No exception is thrown.	
4	Call the EnvelopeHandler.findTLV()to select the Dev Id, the adress and the TPDU TLV and the EnvelopeHandler.compareValue() to check each content.	Check that the message has been re-assembled in the correct order. Check that TP-UDL field is coded one octet. Check that the concatenation control header is not present in	

		the message. Check the integrity of the message.	
5	A new concatenated formatted short message is sent to the SIM composed of two segments. The Address field of the first segment is different from the address field in the second segment.	Applet is triggered.	
6	Call the EnvelopeHandler.getTheHandler()	No exception is thrown.	
7	Call the EnvelopeHandler.findTLV()to select the the address TLV and the EnvelopeHandler.compareValue() to check its content.	Check that the address field of the message is equal to the address field of the second segment.	
8	A new concatenated formatted short message is sent to the SIM composed of two segments. Some TP_elements of the TP_DU of the first segment are different from the TP elements in the second segment.	Applet is triggered.	
9	Call the EnvelopeHandler.getTheHandler()	No exception is thrown.	
10	Call the EnvelopeHandler.findTLV()to select the the TP DU TLV and the EnvelopeHandler.compareValue() to check its TP elements.	Check that the TP elements of the message are equal to the ones of the second segment.	
11	Send a concatenated formatted short message (composed of 2 segment)with uncompressed 8 bits data.	Applet is triggered.	
	Applet registration to EVENT_UNFORMATTED_SMS_PP_ENV and triggering Same test as 1 but with an unformatted SMS_PP envelope. A concatenated unformatted SMS_PP short message is sent to the SIM (composed of two segments).		
12	The second segment of a concatenated short message is sent to the SIM.	Applet is not triggered.	
13	The first segment of the concatenated short message is sent to the SIM	Applet is triggered.	
14	Call the EnvelopeHanlder.getTheHandler()	No exception is thrown.	
15	Call the EnvelopeHandler.findTLV()to select the Dev Id, the adress and the TPDU TLV and the EnvelopeHandler.compareValue() to check each content.	Check that the message has been re-assembled in the correct order. Check that TP-UDL field is coded one octet. Check that the concatenation control header is not present in the message. Check the integrity of the message.	
16	A new concatenated formatted short message is sent to the SIM composed of two segments. The Address field of the first segment is different from the address field in the second segment.	Applet is triggered.	
17	Call the EnvelopeHandler.getTheHandler()	No exception is thrown.	
18	Call the EnvelopeHandler.findTLV()to select the the address TLV and the EnvelopeHandler.compareValue() to check its content.	Check that the address field of the message is equal to the address field of the second segment.	
19	A new concatenated unformatted short message is sent to the SIM composed of two segments. Some TP_elements of the TP_DU of the first segment are different from the TP_elements in the second segment.	Applet is triggered.	
20	Call the EnvelopeHandler.getTheHandler()	No exception is thrown.	
21	Call the EnvelopeHandler.findTLV()to select the the TP DU TLV and the EnvelopeHandler.compareValue() to check its TP elements.	Check that the TP elements of the message are equal to the ones of the second segment.	
22	Send a concatenated unformatted short message (composed of 2 segments) with uncompressed UCS2	Applet is triggered.	

	data.		

6.3.11.4 Test Coverage

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

Annex A (normative): Class and Methods AID numbering and acronyms

A.1 Sim.access

Class Name	Acronyms	Numbering on 5 bits
SIMView	SVW	00001
SIMSystem	SSY	00010
SIMViewException	SVE	00011

A.1.1 SIMView methods

Method Name	Acronyms	Numbering on 6 bits
static final Constants		000001
short increase(byte[] incr, short incrOffset, byte[] resp, short respOffset)	INCR_BS_BS	000010
void invalidate()	INVL	000011
void readBinary(short fileOffset, byte[] resp, short respOffset, short respLength)	REDBS_BSS	000100
short readRecord(short recNumber, byte mode, short recOffset, byte[] resp, short respOffset, short respLength)	REDRSBS_BSS	000101
void rehabilitate()	REHA	000110
short seek(byte mode, byte[] patt, short pattOffset, short pattLength)	SEEKB_BSS	000111
void select(short fid)	SLCTS	001000
short select(short fid, byte[] fci, short fciOffset, short fciLength)	SLCTS_BSS	001001
short status(byte[] fci, short fciOffset, short fciLength)	STAT_BSS	001010
short updateBinary(short fileOffset, byte[] data, short dataOffset, short dataLength)	UPDBS_BSS	001011
void updateRecord(short recNumber, byte mode, short recOffset, byte[] data, short dataOffset, short dataLength)	UPDRSBS_BSS	001100

A.1.2 SIMSystem methods

Method Name	Acronyms	Numbering on 6 bits
static SIMView getTheSIMView()	GETS	000001

A.1.3 SIMViewException methods

Method Name	Acronyms	Numbering on 6 bits
static void throwIt(short reason)	THITS	000001
SIMViewException(short reason)	COORS	000010
Constants	CONS	000011

A.2 Sim.toolkit

Class Name	Acronyms	Numbering on 5 bits
ToolkitConstants	TKC	00001
ToolkitInterface	TKI	00010

Class Name	Acronyms	Numbering on 5 bits
EditHandler	EDH	00011
EnvelopeHandler	ENH	00100
EnvelopeResponseHandler	ERH	00101
MEProfile	MEP	00110
ProactiveHandler	PAH	00111
ProactiveResponseHandler	PRH	01000
ToolkitRegistry	TKR	01001
ViewHandler	VWH	01010
ToolkitException	TKE	01011

A.2.1 ToolkitConstants

Method Name	Acronyms	Numbering on 6 bits
Constants	CONS	000001

A.2.2 ToolkitInterface methods

Method Name	Acronyms	Numbering on 6 bits
void processToolkit (byte event)	PRTKB	000001

A.2.3 EditHandler methods

The numbering of the EditHandler methods it will be done in the classes inherit it: EnvelopeResponseHandler, ProactiveHandler, because the methods provided by this class as it is declared 'abstract'.

A.2.4 EnvelopeHandler methods

Method Name	Acronyms	Numbering on 6 bits
Byte getEnvelopeTag()	GENT	000001
Byte getItemIdentifier()	GIID	000010
Short getSecuredDataLength()	GSDL	000011
Short getSecuredDataOffset()	GSDO	000100
EnvelopeHandler getTheHandler()	GTHD	000101
Short getTPUDLOffset()	GTPO	000110
Short getCapacity()	GCAP	010010
Short getUserDataLength()	GUDL	010011
Byte getChannelIdentifier()	GCID	010100
Inherited Method Name: ViewHandler		
Byte compareValue(short valueOffset,byte[] compareBuffer,short compareOffset, short compareLength)	CPRVS_BSS	000111
Short copy(byte[] dstBuffer,short dstOffset,short dstLength)	COPY_BSS	001000
Short copyValue(short valueOffset, byte[] dstBuffer,short dstOffset,short dstLength)	CPYVS_BSS	001001
Byte findAndCompareValue(byte tag,byte[] compareBuffer,short compareOffset)	FACRB_BS	001010
Byte findAndCompareValue(byte tag,byte occurrence, short valueOffset,byte[] compareBuffer,short compareOffset,short compareLength)	FACRBBS_BSS	001011
Short FindAndCopyValue(byte tag,byte occurrence,short valueOffset, byte[] dstBuffer, short dstOffset,	FACYBBS_BSS	001100

Method Name	Acronyms	Numbering on 6 bits
short dstLength)		
Short findAndCopyValue(byte tag,byte[] dstBuffer,short dstOffset)	FACYB_BS	001101
Byte FindTLV(byte tag,byte occurrence)	FINDBB	001110
Short GetLength()	GLEN	001111
Byte GetValueByte(short valueOffset)	GVBYS	010000
Short GetValueLength()	GVLE	010001

A.2.5 EnvelopeResponseHandler methods

Method Name	Acronym	Numbering on 6 bits
EnvelopeResponseHandler getTheHandler()	GTHD	000001
Void post(byte statusType)	POSTB	000010
Void postAsBERTLV(byte statusType, byte tag)	POSTBB	000011
Short getCapacity()	GCAP	010101
Inherited Method Name: EditHandler		
Void appendArray(byte[] buffer, short offset, short length, short dstLength)	APDA_BSS	000100
Void appendTLV(byte tag, byte value)	APTLBB	000101
Void appendTLV(byte tag, byte[] value, short valueOffset, short valueLength)	APTLB_BSS	000110
Void appendTLV(byte tag, byte value1, byte value2)	APTLBBB	000111
Void appendTLV(byte tag, byte value1, byte[] value2, short value2Offset, short value2Length)	APTLBB_BSS	001000
Void clear()	CLER	001001
Inherited Method Name: ViewHandler		
Byte compareValue(short valueOffset,byte[] compareBuffer,short compareOffset, short compareLength)	CPRVS_BSS	001010
Short Copy(byte[] dstBuffer,short dstOffset,short dstLength)	COPY_BSS	001011
Short CopyValue(short valueOffset, byte[] dstBuffer,short dstOffset,short dstLength)	CPYVS_BSS	001100
Byte findAndCompareValue(byte tag,byte[] compareBuffer,short compareOffset)	FACRB_BS	001101
Byte findAndCompareValue(byte tag,byte occurence, short valueOffset,byte[] compareBuffer,short compareOffset,short compareLength)	FACRBBS_BSS	001110
Short findAndCopyValue(byte tag,byte occurence,short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	FACYBBS_BSS	001111
Short findAndCopyValue(byte tag,byte[] dstBuffer,short dstOffset)	FACYB_BS	010000
Byte findTLV(byte tag,byte occurrence)	FINDBB	010001
Short GetLength()	GLEN	010010
Byte getValueByte(short valueOffset)	GVBYS	010011
Short getValueLength()	GVLE	010100

A.2.6 MEProfile methods

Method Name	Acronym	Numbering on 6 bits
static boolean check(byte index)	CHECB	000001
static boolean check(byte[] mask, short offset, short length)	CHECBSS	000010
static boolean check(short index)	CHECS	000011
static short copy(short startOffset, byte[] dstBuffer, short dstOffset, short dstLength)	COPYS_BSS	000100
static short getValue(short indexMSB, short indexLSB)	GVALSS	000101

A.2.7 ProactiveHandler methods

Method Name	Acronyms	Numbering on 6 bits
ProactiveHandler getTheHandler()	GTHD	000001
Void init(byte type, byte qualifier, byte dstDevice)	INITBBB	000010
Void initDisplayText(byte qualifier, byte dcs, byte[] buffer, short offset, short length)	INDTBB_BSS	000011
Void initGetInkey(byte qualifier, byte dcs, byte[] buffer, short offset, short length)	INGKBB_BSS	000100
Void initGetInput(byte qualifier, byte dcs, byte[] buffer, short offset, short length, short minRespLength, short maxRespLength)	INGPBB_BSSSS	000101
Byte send()	SEND	000110
Short getCapacity()	GCAP	011000
Void initCloseChannel(byte bChannelIdentifier)	ICCHB	011001
Inherited Method Name: EditHandler		
Void appendArray(byte[] buffer, short offset, short length, short dstLength)	APDA_BSS	000111
Void appendTLV(byte tag, byte value)	APTLBB	001000
Void appendTLV(byte tag, byte[] value, short valueOffset, short valueLength)	APTLB_BSS	001001
Void appendTLV(byte tag, byte value1, byte value2)	APTLBBB	001010
Void appendTLV(byte tag, byte value1, byte[] value2, short value2Offset, short value2Length)	APTLBB_BSS	001011
Void clear()	CLER	001100
Inherited Method Name: ViewHandler		
Byte compareValue(short valueOffset,byte[] compareBuffer,short compareOffset, short compareLength)	CPRVS_BSS	001101
Short copy(byte[] dstBuffer,short dstOffset,short dstLength)	COPY_BSS	001110
Short copyValue(short valueOffset, byte[] dstBuffer,short dstOffset,short dstLength)	CPYVS_BSS	001111
Byte findAndCompareValue(byte tag,byte[] compareBuffer,short compareOffset)	FACRB_BS	010000
Byte findAndCompareValue(byte tag,byte occurence, short valueOffset,byte[] compareBuffer,short compareOffset,short compareLength)	FACRBBS_BSS	010001
Short findAndCopyValue(byte tag,byte occurence,short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	FACYBBS_BSS	010010
Short findAndCopyValue(byte tag,byte[] dstBuffer,short dstOffset)	FACYB_BS	010011
Byte findTLV(byte tag,byte occurrence)	FINDBB	010100
Short getLength()	GLEN	010101
Byte getValueByte(short valueOffset)	GVBYS	010110
Short getValueLength()	GVLE	010111

A.2.8 ProactiveResponseHandler methods

Method Name	Acronyms	Numbering on 6 bits
Short copyAdditionalInformation(byte[] dstBuffer, short dstOffset, short dstLength)	CPAI_BSS	000001
Short copyTextString(byte[] dstBuffer, short dstOffset)	CPTS_BS	000010
Short getAdditionalInformationLength()	GTIL	000011
Byte getGeneralResult()	GTGR	000100
Byte getItemIdentifier()	GTII	000101
Byte getTextStringCodingScheme()	GTCS	000110
Short getTextStringLength()	GTTL	000111
ProactiveResponseHandler getTheHandler()	GTHD	001000
Short getCapacity()	GCAP	010100
Byte getChannelIdentifier()	GCID	010101
Short copyChannelData(byte[] dstBuffer, short dstOffset, short dstLength)	CCHD_BSS	010110
Inherited Method Name: ViewHandler		
Byte CompareValue(short valueOffset,byte[] compareBuffer,short compareOffset, short compareLength)	CPRVS_BSS	001001
Short Copy(byte[] dstBuffer,short dstOffset,short dstLength)	COPY_BSS	001010
Short CopyValue(short valueOffset, byte[] dstBuffer,short dstOffset,short dstLength)	CPYVS_BSS	001011
Byte FindAndCompareValue(byte tag,byte[] compareBuffer,short compareOffset)	FACRB_BS	001100
Byte findAndCompareValue(byte tag,byte occurence, short valueOffset,byte[] compareBuffer,short compareOffset,short compareLength)	FACRBBS_BSS	001101
Short FindAndCopyValue(byte tag,byte occurence,short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	FACYBBS_BSS	001110
Short findAndCopyValue(byte tag,byte[] dstBuffer,short dstOffset)	FACYB_BS	001111
Byte FindTLV(byte tag,byte occurrence)	FINDBB	010000
Short GetLength()	GLEN	010001
Byte GetValueByte(short valueOffset)	GVBYS	010010
Short GetValueLength()	GVLE	010011

A.2.9 ToolkitRegistry methods

Method Name	Acronyms	Numbering on 6 bits
AllocateTimer()	ATIM	000001
changeMenuEntry(byte id, byte[] menuEntry, short offset, short length, byte nextAction, boolean helpSupported, byte iconQualifier, short iconIdentifier)	CMETB_BSSBZBS	000010
clearEvent(byte event)	CEVTB	000011
disableMenuEntry(byte id)	DMETB	000100
enableMenuEntry(byte id)	EMETB	000101
getEntry()	GETY	000110
getPollInterval()	GPOL	000111
initMenuEntry(byte[] menuEntry, short offset, short	IMET_BSSBZBS	001000

Method Name	Acronyms	Numbering on 6 bits
`length, byte nextAction, boolean helpSupported, byte iconQualifier, short iconIdentifier)`		
`isEventSet(byte event)`	IEVSB	001001
`releaseTimer(byte timerIdentifier)`	RTIM	001010
`requestPollInterval(short duration)`	RPOL	001011
`setEvent(byte event)`	SEVTB	001100
`setEventList(byte[] eventList, short offset, short length)`	SEVL_BSS	001101

A.2.10 ViewHandler methods

The numbering of the ViewHandler methods it will be done in the classes inherit it: EditHandler, EnvelopeHandler, ProactiveResponseHandler, ProactiveHandler, because the methods provided by this class as it is declared 'abstract'.

A.2.11 ToolkitException methods

Method Name	Acronyms	Numbering on 6 bits
`Static void throwIt(short reason)`	THITS	000001
`ToolkitException(short reason)`	COORS	000010
`Constants`	CONS	000011

Annex B (normative): Script file syntax and format description

B.1 Syntax description

Following is a syntax description in BNF.

<statement list> ::=	[<statement> **\n**] +
<statement> ::=	<simple> | <switch> | <*blank line*>
<simple> ::=	<reset> | <init> | <command> | <remark>
<reset> ::=	**RST**
<init> ::=	**INI** <*hexdata*>
<command> ::=	**CMD** <*hexdata*> [<response>] (<status>)
<response> ::=	**[** <*hexdata*> **]**
<status> ::=	**(** <*hexdata*> **)**
<remark> ::=	**REM** <*text line*>
<switch> ::=	**SWI {** [<labelled list>] + **}**
<labelled list> ::=	<label> **: \n** <statement list>

Description of syntax metalanguage :

\n	represents a linebreak
[x]	means x can appear optionally
[x] +	means 1 or more appearances of x
x | y	means x or y
[]{}:	(bold) these are characters that appear literally in the script files
<text line>	any character until the end of the line
<blank line>	a line containing no text is acceptable
<hexdata>	data written in hexadecimal, each byte separated from the following by a whitespace

Each simple statement beginning with 3 characters different than the ones defined indicates another tool command, and shall be ignored by the parser if not recognized.

`' ', '\t'` : Can be used as separator
A long statement can be broken into several lines by using the character '****' at the end of each line which is not the last one in the statement.

For more details refer to the examples in B.3.

B.2 Semantics

Following is the meaning of each of the statements:

`CMD` : Sends an APDU Command to the card, including (optionally) the expected response data and also (optionally) the expected status words SW1, SW2.

`RST` : Resets and powers on the card

`INI` : Performs the terminal profile with the following data. Afterwards, it shall perform all the fetch and terminal response commands until there is no proactive session in progress.

`REM` : Used for comments

`SWI` : Activates a switch condition. Every labelled list represents a list of statements to be executed, if the label matches the SW resulting from the previously executed command.

Evaluation of expected response and status in the case of a CMD:

`<response>` data within **[**…**]** has to be checked, it needs to be present for an outgoing command. Bytes written as XX shall not be checked by the APDU tool.

`<status>` status contained within **(**…**)** has to be checked; when several status are valid they shall be separated by commas. Nibble written as X shall not be checked by the APDU tool.

B.3 Example

```
REM this is an example

RST
INI FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
REM Case 1 example
CMD A0 C2 00 00 00 (91 33 , 69 XX)

REM Case 2 example
CMD A0 B6 00 00 07 \
    [XX XX XX 55 55 XX 55] \
    (91 33 , 67 XX)


CMD A0 B6 00 00 07 \
    (91 33 , 67 XX)

CMD A0 C0 00 00 1F \
    [10 A0 00 00 00 09 00 02 FF FF FF FF 89 28 A4 05 \
    02 0D CC CC CC CC CC CC CC CC CC CC CC CC CC ] \
    (90 00)

REM Case 3 example
CMD A0 C2 00 00 33 \
    D1 31 82 02 83 81 06 05 80 11 22 33 44 8B 24 40 \
    08 00 24 23 85 18 41 04 51 10 10 00 00 00 00 13 \
    02 70 00 00 0E 0D 00 00 00 00 28 A4 05 00 00 00 \
    00 00 00 \
    (90 00)

REM Case 4 example with switch statement
CMD 00 A4 04 00 10 \
    A0 00 00 00 09 00 02 FF FF FF FF 89 41 04 44 02 \
    (61 XX, 6A 82)

SWI {
61 XX:
CMD 00 C0 00 00 14 \
    [10 A0 00 00 00 09 00 02 FF FF FF FF 89 41 04 44 \
    02 02 CC CC] \
    (90 00)

CMD A0 A4 00 00 02 \
    3F 00

6A 82:
RST
}

REM Case 5 example
CMD A0 C2 00 00 33 \
    D1 31 82 02 83 81 06 05 80 11 22 33 44 8B 24 40 \
    08 00 24 23 85 18 41 04 51 10 10 00 00 00 00 13 \
    02 70 00 00 0E 0D 00 00 00 00 28 A4 05 00 00 00 \
    00 00 00 \
    (6X 00)
```

B.4 Style and formatting

In order to show a common appearance all the scripts shall follow those format rules:

- start always with a 'RST'.
- The command, data to be checked and status to be checked shall be presented in the following order:

  CMD *COMMAND* [*EXPECTED DATA*] (*EXPECTED STATUS*)
- APDU shall be presented with command (CLA INS P1 P2 P3) in one line and data (if present) in next line grouped 16 bytes per line (see example above).
- The expected data (if present) shall be presented in 16 bytes groups per line (see example above).

Annex C (normative): Default Prepersonalization

C.1 General Default Prepersonalization

This table shows the default prepersonalization, the file system and the files' content, that the test SIM cards shall contain unless otherwise stated.

Name	Identifier	Default Value	Special Features
EF_{ICCID}	2FE2	0F FF FF FF FF FF FF FF FF FF	This value is not compliant with 3GPP TS 51.011 [3]
EF_{IMSI}	6F07	FF FF FF FF FF FF FF FF FF	This value is not compliant with 3GPP TS 51.011 [3]
EF_{LP}	6F05	01 FF FF FF	
EF_{Kc}	6F20	FF FF FF FF FF FF FF FF 07	
$EF_{PLMNsel}$	6F30	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
EF_{HPLMN}	6F31	05	
EF_{ACMmax}	6F37	00 00 00	Access condition UPDATE: CHV1
EF_{SST}	6F38	FF 3F C3 0F 0C 00 FF 0F 00 33	
EF_{ACM}	6F39	00 00 00	Access condition UPDATE: CHV1
EF_{PUCT}	6F41	FF FF FF 00 00	Access condition UPDATE: CHV1
EF_{BCCH}	6F74	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
EF_{ACC}	6F78	00 00	
EF_{FPLMN}	6F7B	FF FF FF FF FF FF FF FF FF FF FF FF	
EF_{LOCI}	6F7E	FF FF FF FF 00 F0 00 00 00 FF 01	
EF_{AD}	6FAD	00 FF FF	
EF_{Phase}	6FAE	03	
EF_{FDN}	6F3B	Default value in all the records: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 5
EF_{SMSP}	6F42	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF_{LND}	6F44	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF_{SMSS}	6F43	FF FF	
EF_{SMS}	6F3C	1st record: 00 FF … FF(length 176) 2nd record:00 FF … FF(length 176) 3rd record: 00 FF … FF(length 176)	Records: 3
EF_{ADN}	6F3A	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF_{CCP}	6F3D	FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
EF_{MSISDN}	6F40	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF_{SDN}	6F49	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF_{SUME}	6F54	85 0C 54 4F 4F 4C 4B 49 54 20 54 45 53 54 FF FF FF FF	
EF_{CBMI}	6F45	FF FF	
EF_{CBMID}	6F48	10 80	
EF_{CBMIR}	6F50	10 80 10 9F	
EF_{IMG}	4F20	FF FF FF FF FF FF FF FF FF FF FF	

The default value for the CHV1 shall be "0x31 0x31 0x31 0x31 0xFF 0xFF 0xFF 0xFF" and its state shall be 'disabled' during test applets execution.

C.2 Sim.Access.SimView test default prepersonalization

C.2.1 DF$_{\text{SIMTEST}}$ (SIM Test)

Identifier: '0319'

C.2.2 EF$_{\text{TNR}}$ (Transparent Never Read)

<table>
<tr><td colspan="3">Identifier: '6F01'</td><td>Structure: transparent</td><td colspan="2">Mandatory</td></tr>
<tr><td colspan="3">File size: 3 bytes</td><td colspan="3">Update activity: low</td></tr>
<tr><td colspan="6">Access Conditions:
READ NEVER
UPDATE ALWAYS
INVALIDATE ALWAYS
REHABILITATE ALWAYS</td></tr>
<tr><td>Bytes</td><td>Description</td><td colspan="2">Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1 - 3</td><td>Test Data</td><td colspan="2">AA AA AA</td><td>M</td><td>3 bytes</td></tr>
</table>

C.2.3 EF$_{\text{TNU}}$ (Transparent Never Update)

<table>
<tr><td colspan="3">Identifier: '6F02'</td><td>Structure: transparent</td><td colspan="2">Mandatory</td></tr>
<tr><td colspan="3">File size: 3 bytes</td><td colspan="3">Update activity: low</td></tr>
<tr><td colspan="6">Access Conditions:
READ ALWAYS
UPDATE NEVER
INVALIDATE ALWAYS
REHABILITATE ALWAYS</td></tr>
<tr><td>Bytes</td><td>Description</td><td colspan="2">Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1 - 3</td><td>Test Data</td><td colspan="2">55 55 55</td><td>M</td><td>3 bytes</td></tr>
</table>

C.2.4 EF$_{\text{TARU}}$ (Transparent Always Read and Update)

<table>
<tr><td colspan="3">Identifier: '6F03'</td><td>Structure: transparent</td><td colspan="2">Mandatory</td></tr>
<tr><td colspan="3">File size: 260 bytes</td><td colspan="3">Update activity: low</td></tr>
<tr><td colspan="6">Access Conditions:
READ ALWAYS
UPDATE ALWAYS
INVALIDATE ALWAYS
REHABILITATE ALWAYS</td></tr>
<tr><td>Bytes</td><td>Description</td><td colspan="2">Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1 - 260</td><td>Test Data</td><td colspan="2">FF ... FF</td><td>M</td><td>260 bytes</td></tr>
</table>

C.2.5 EF$_{CNR}$ (Cyclic Never Read)

Identifier: '6F04'	Structure: cyclic		Mandatory	
Record length: 3 bytes		Update activity: high		
Access Conditions: READ NEVER UPDATE ALWAYS INCREASE ALWAYS INVALIDATE ALWAYS REHABILITATE ALWAYS				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.6 EF$_{CNU}$ (Cyclic Never Update)

Identifier: '6F05'	Structure: cyclic		Mandatory	
Record length: 3 bytes		Update activity: high		
Access Conditions: READ ALWAYS UPDATE NEVER INCREASE NEVER INVALIDATE ALWAYS REHABILITATE ALWAYS				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.7 EF$_{CNIC}$ (Cyclic Never Increase)

Identifier: '6F06	Structure: cyclic		Mandatory	
Record length: 3 bytes		Update activity: high		
Access Conditions: READ ALWAYS UPDATE ALWAYS INCREASE NEVER INVALIDATE ALWAYS REHABILITATE ALWAYS				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.8 EF$_{\text{CNIV}}$ (Cyclic Never Invalidate)

Identifier: '6F07	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	NEVER
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.9 EF$_{\text{CNRH}}$ (Cyclic Never Rehabilitate)

Identifier: '6F08'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	ALWAYS
REHABILITATE	NEVER

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.10 EF$_{\text{CARU}}$ (Cyclic Always Read and Update)

Identifier: '6F09'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	55 55 55	M	3 bytes
2	Test Data	AA AA AA	M	3 bytes

C.2.11 EF$_{LNR}$ (Linear Fixed Never Read)

<table>
<tr><td colspan="2">Identifier: '6F0A'</td><td colspan="2">Structure: linear fixed</td><td>Mandatory</td></tr>
<tr><td colspan="2">Record length: 4 bytes</td><td colspan="3">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ NEVER
UPDATE ALWAYS
INVALIDATE ALWAYS
REHABILITATE ALWAYS</td></tr>
<tr><td>Logical Record Number</td><td>Description</td><td>Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>Test Data - Record 1</td><td>FF FF FF FF</td><td>M</td><td>4 bytes</td></tr>
<tr><td>2</td><td>Test Data - Record 2</td><td>FF FF FF FF</td><td>M</td><td>4 bytes</td></tr>
</table>

C.2.12 EF$_{LNU}$ (Linear Fixed Never Update)

<table>
<tr><td colspan="2">Identifier: '6F0B'</td><td colspan="2">Structure: linear fixed</td><td>Mandatory</td></tr>
<tr><td colspan="2">Record length: 4 bytes</td><td colspan="3">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALWAYS
UPDATE NEVER
INVALIDATE ALWAYS
REHABILITATE ALWAYS</td></tr>
<tr><td>Logical Record Number</td><td>Description</td><td>Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>Test Data - Record 1</td><td>FF FF FF FF</td><td>M</td><td>4 bytes</td></tr>
<tr><td>2</td><td>Test Data - Record 2</td><td>FF FF FF FF</td><td>M</td><td>4 bytes</td></tr>
</table>

C.2.13 EF$_{LARU}$ (Linear Fixed Always Read and Update)

<table>
<tr><td colspan="2">Identifier: '6F0C'</td><td colspan="2">Structure: linear fixed</td><td>Mandatory</td></tr>
<tr><td colspan="2">Record length: 4 bytes</td><td colspan="3">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALWAYS
UPDATE ALWAYS
INVALIDATE ALWAYS
REHABILITATE ALWAYS</td></tr>
<tr><td>Logical Record Number</td><td>Description</td><td>Default Value</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>Test Data - Record 1</td><td>55 55 55 55</td><td>M</td><td>4 bytes</td></tr>
<tr><td>2</td><td>Test Data - Record 2</td><td>AA AA AA AA</td><td>M</td><td>4 bytes</td></tr>
</table>

C.2.14 EF$_{CINA}$ (Cyclic Increase Not Allowed)

Identifier: '6F0D'		Structure: cyclic		Mandatory
Record length: 3 bytes			Update activity: high	
Access Conditions: READ ALWAYS UPDATE ALWAYS INCREASE ALWAYS (see note) INVALIDATE ALWAYS REHABILITATE ALWAYS				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes
NOTE: This file will be personalized in a way such that increase is not allowed, as indicated by the FCI byte 8, bit 7 (3GPP TS 51.011 [3]: FCI structure of an EF returned by the SELECT command)				

C.2.15 EF$_{TRAC}$ (Transparent Read Access Condition CHV2)

Identifier: '6F0E'		Structure: transparent		Mandatory
Record length: 3 bytes			Update activity: low	
Access Conditions: READ CHV2 UPDATE ALWAYS INCREASE ALWAYS INVALIDATE ALWAYS REHABILITATE ALWAYS				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes

C.2.16 EF$_{TIAC}$ (Transparent Invalidate Access Condition CHV1)

Identifier: '6F0F'		Structure: transparent		Mandatory
Record length: 3 bytes			Update activity: low	
Access Conditions: READ ALWAYS UPDATE ALWAYS INCREASE ALWAYS INVALIDATE CHV1 REHABILITATE ALWAYS				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes

C.2.17 EF$_{\text{CIAC}}$ (Cyclic Increase Access Condition CHV2)

Identifier: '6F10'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	CHV2
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.18 EF$_{\text{CIAA}}$ (Cyclic Increase Access Condition ADM)

Identifier: '6F11'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ADM
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.19 EF$_{\text{CNRI}}$ (Cyclic Never Rehabilitate Invalidated)

Identifier: '6F12'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	ALWAYS
REHABILITATE	NEVER

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

The file status shall be invalidated as defined in 3GPP TS 51.011 [3].

Annex D (normative): sim.test.util package and loading, testing and cleaning script examples

See attached files:

- Annex_D_SimTestUtil.zip
- Annex_D_Examples.zip

Annex E (normative): Test Area files

See attached file:

- Annex_E_SourceCode.zip

Annex F (normative): AID numbering and acronyms for Framework tests

F.1 Toolkit Installation Parameters (TIN)

Test Area within the chapter	Acronyms	Numbering on 6 bits
Timer allocation	TMAL	000001
Item identifier	ITID	000010
Item position	ITPO	000011
Access conditions	ACCO	000100
Priority level	PRLV	000101
Maximum length for each menu entry	MLME	000110
Number of menu entries	NBME	000111
Memory space	MESP	001000
Channel Allocation	CHAL	001001
Minimum Security Level	MSL	001010

F.2 Minimum Handler Availability (MHA)

Test Area within the chapter	Acronyms	Numbering on 6 bits
ProactiveHandler	PAHD	000001
ProactiveResponseHandler	PRHD	000010
EnvelopeHandler	ENHD	000011
EnvelopeResponseHandler	ERHD	000100

F.3 Handler Integrity (HIN)

Test Area within the chapter	Acronyms	Numbering on 6 bits
ProactiveHandler	PAHD	000001
ProactiveResponseHandler	PRHD	000010
EnvelopeHandler	ENHD	000011
EnvelopeResponseHandler	ERHD	000100

F.4 Applet Triggering (APT)

Test Area within the chapter	Acronyms	Numbering on 6 bits
EVENT_PROFILE_DOWNLOAD	EPDW	000001
EVENT_MENU_SELECTION	EMSE	000010
EVENT_MENU_SELECTION_HELP_REQUEST	EMSH	000011
EVENT_FORMATTED_SMS_PP_ENV	EFSE	000100
EVENT_UNFORMATTED_SMS_PP_ENV	EUSE	000101
EVENT_CALL_CONTROL_BY_SIM	ECCN	000110
EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM	EMCN	000111
EVENT_TIMER_EXPIRATION	ETEX	001000
EVENT_UNFORMATTED_SMS_CB	EUCB	001001
EVENT_EVENT_DOWNLOAD_MT_CALL	EDMC	001010
EVENT_EVENT_DOWNLOAD_CALL_CONNECTED	EDCC	001011
EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED	EDCD	001100
EVENT_EVENT_DOWNLOAD_LOCATION_STATUS	EDLS	001101
EVENT_EVENT_DOWNLOAD_USER_ACTIVITY	EDUA	001110
EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE	EDIS	001111

Test Area within the chapter	Acronyms	Numbering on 6 bits
EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS	EDCR	010000
EVENT_UNRECOGNIZED_ENVELOPE	EUEV	010001
EVENT_STATUS_COMMAND	ESTC	010010
EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION	EDLG	010011
EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION	EDBT	010100
EVENT_FORMATTED_SMS_CB	EFCB	010101
EVENT_FIRST_COMMAND_AFTER_SELECT	EFCA	010110
EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE	EDDA	010111
EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS	EDCS	011000
EVENT_FORMATTED_SMS_PP_UPD	EFSU	011001
EVENT_UNFORMATTED_SMS_PP_UPD	EUSU	011010

F.5 Proactive Command Sending (PCS)

Test Area within the chapter	Acronyms	Numbering on 6 bits
System Proactive commands	SPCO	000001
Interaction with GSM commands	IGCO	000010
Errors during proactive command sending	EPCS	000011
Proactive Command Control	PCCO	000100

F.6 Envelope Response Posting (ERP)

Test Area within the chapter	Acronyms	Numbering on 6 bits
EVENT_CALL_CONTROL_BY_SIM	ECCN	000001
EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM	EMCN	000010
EVENT_UNRECOGNIZED_ENVELOPE	EUEN	000011
EVENT_FORMATTED_SMS_PP_ENV	EFSE	000010

F.7 Framework Security (FWS)

Test Area within the chapter	Acronyms	Numbering on 6 bits
Input data	INDA	000001
Output data	OUDA	000010

F.8 File System Context (FSC)

Test Area within the chapter	Acronyms	Numbering on 6 bits
Initial Context	INIT	000001
Context Preservation for Current File	CUFI	000010
Context Preservation for Current Record	CURE	000011

F.9 Exception Handling (EXH)

Test Area within the chapter	Acronyms	Numbering on 6 bits
Hide exception to the mobile	HEME	000001
Interaction with multi-triggering	IMTG	000010

F.10 Other parts transferred to framework from API (API)

Test Area within the chapter	Acronyms	Numbering on 6 bits
A handler is a temporary JCRE Entry Point object	HEPO	000001
Transaction	TRAN	000010
Timer Id between Applets	TMID	000011

F.11 Concatenation processing (PROC)

Test Area within the chapter	Acronyms	Numbering on 6 bits
Concatenation processing	PROC	000001

Annex G (normative): Configuration Parameters File

This file describes all the mandatory and optional parameters that are used in order to create the loading script(s) for one test area. The configuration parameters file contains the values for the parameters needed in order to generate the loading and cleanup scripts.

The name of the parameters file will be ***<test area reference>*_*<n>*.par**.

The number <n> is associated with the loading/cleanup script number, i.e. API_2_TKR_ SEVL_BSS_1.par is used to generate API_2_TKR_ SEVL_BSS_1.ldr etc.

G.1 Syntax

The general syntax for this file will be:

```
<file> ::= <section>+
<section> ::= <section heading> <line break> <section body>
<section heading> ::= '[' <name> ']'
<section body> ::= <parameter assignment>+
<parameter assignment> ::= <name> '=' <value> <line break>
```

Where '+' indicates one or more repetitions of the previous syntax element.

Any text included between the symbol ';' and the end of line is considered a comment and ignored by parsing tools.

Empty values are considered valid. They are used to indicate that an optional value is not present.

Names of sections, names of parameters and values are case-sensitive.

Blank spaces and Tabs between tokens are allowed and will be ignored by the parser.

When values represent a sequence of bytes, they are expressed in hexadecimal format, where every 2 digits represent one byte. Blank space between bytes is optional.

Example:

```
; comment



[Section1]

Parameter11 = 00 11 22 33

Parameter12 = 0101   ; another comment



[Section2]

Parameter21 = vvwwxxyyzz
```

G.2 File Contents and Organization

Parameters in this file are organized in the following sections:

[CONVERT]	Conversion parameters used during conversion (i.e. CAP file generation)
[INSTALL(load)]	Parameters used by the Install for Load command
[LOAD]	Parameters used by the Load command
[INSTALL(install)]	Parameters used by the Install for Install command

All sections may appear only once in the file, except for the "INSTALL(install)" section. If that section appears more than once, it will apply to different applet instances, in sequence.

G.2.1 Default values, order and processing

The ordering of the parameters and the sections is relevant, since parameter names may be repeated and apply to different applets.

When one single parameter is repeated within one section, it refers to different applets. The value of the n^{th} appearance of the parameter applies to applet *n*.

When one section is repeated (INSTALL(install)), then the n^{th} appearance of the section applies to applet *n*. Parameter/value pairs which are found in one appearance of the section are valid for the subsequent applets as long as they are not overridden. For example, first INSTALL(install) may contain all values for parameters, whereas the subsequent INSTALL(install) sections may only contain parameters whose values change.

If one required parameter is missing from one section, the last defined value of this parameter in a previous section of the same file will be used.

G.2.2 CONVERT Section

These parameters allow configuration of the conversion process of the Java class file(s) into one CAP file.

Parameter	Description
PackageAID	AID of the package
PackageName	Fully qualified name of the package
PackageVersion	Version of the package
AppletClassAID	AID of the applet
AppletClassName	Name of the applet

G.2.3 INSTALL(load) Section

Here are the parameters to be included in the Install(Load) command (as specified in TS 23.048 [8]).

Parameter	Description
PackageAID	AID of the package
PackageNonVolatileMemSize	Non Volatile memory space (in bytes) required for package loading
InstallationNonVolatileMemSize	Non volatile memory required for installation, in bytes
InstallationVolatileMemSize	Volatile memory required for installation, in bytes

G.2.4 LOAD Section

Here are the parameters to be included in the Load command (as specified in TS 23.048 [8]).

Parameter	Description
MaxLoadCommandDataLength	Maximum length of the data provided in the load command (P3 parameter of the LOAD APDU embedded in the command packet)

G.2.5 INSTALL(install) Section

Here are the parameters to be included in the Install(Install) command (as specified in 3GPP TS 23.048 [8]).

Parameter	Description
PackageAID	AID of the package
AppletClassAID	AID of the applet
InstanceAID	AID of the instance of the applet
InstallationNonVolatileMemSize	Non volatile memory required for installation, in bytes
InstallationVolatileMemSize	Volatile memory required for installation, in bytes
AccessDomain	Specify the SIM files that may be accessed by the applet and the operations allowed on these files. This parameter includes the Access Domain Parameter (ADP) and Access Domain Data (ADD)
PriorityLevel	Priority level of the Toolkit applet instance
MaxNumberOfTimers	Maximum number of timers allowed for this applet instance
MaxMenuEntryTextLength	Maximum text length for a menu entry
MaxNumberOfMenuEntries	Maximum number of menu entries allowed for this applet instance
MenuEntriesPositionIdentifier	For each menu entry: Position and identifier of that menu entry
MaxNumberOfChannels	Maximum Number of channels for this applet instance
MSLFieldLength	Length of Minimum Security Level field
MSLParameter	MSL Parameter
MSLData	MSL Data
AppletSpecificParameters	Parameters specific to the applet

The applet shall be installed with install(install and make selectable) command.

G.3 Full example

```
[CONVERT]

PackageAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 00

PackageName = sim.test.access.api_1_svw_updrbs

PackageVersion = 1.0

AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 01

AppletClassName = API_1_SVW_UPDRBS_1

AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 02

AppletClassName = API_1_SVW_UPDRBS_2


[INSTALL(load)]

PackageNonVolatileMemSize = 0D27

;InstallationNonVolatileMemSize = 0400

;InstallationVolatileMemSize = 0000


[LOAD]

MaxLoadCommandDataLength = 6C ; max value


[INSTALL(install)]

AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 01
```

```
InstanceAID =    A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 01

InstallationNonVolatileMemSize = 0400

InstallationVolatileMemSize = 0000

AccessDomain = 00

PriorityLevel = FF

MaxNumberOfTimers = 00

MaxMenuEntryTextLength = 10

MaxNumberOfMenuEntries = 01

MenuEntriesPositionIdentifier = 0001

AppletSpecificParameters =


[INSTALL(install)]

AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 02

InstanceAID =    A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 02

InstallationNonVolatileMemSize = 0200

InstallationVolatileMemSize = 0000

MenuEntriesPositionIdentifier = 0002

MaxNumberOfChannels = 05

MSLFieldLength = 00

MSLParameter =

MSLData =
```

```
; rest of INSTALL(install) parameters are taken from previous INSTALL(install)...
```

Annex H (informative): Change History

The table below indicates all changes that have been made to the present document since drafting work began.

TSG / Date	TSG Doc	WG doc	CR	Rev	Subject/Comment	New
TP-20	TP-030125		001	-	Update of 51.013 Specification for Release 5 based on version 4.0.1	5.0.0
					editorial: replacment of annex E due to problems in the folder-structure.	5.0.1
TP-22	TP-030258		003	-	Essential corrections	5.1.0
TP-26	TP-040269		005	-	Correction of release references.	5.2.0
			006	-	ProactiveHandler appendTLV(byte tag, byte value1, byte value2) method conformance requirement.	
			007	-	Correction to ProactiveHandler appendTLV(byte tag, byte[] value, short valueOffset, short valueLength) method test.	
			008	-	Correction to dstBuffer length and dstLength in ProactiveResponseHandler copyChannelData() method tests.	
			009	-	Correction to updateRecord() method test in access package.	
			010	-	Addition of tests on HANDLER_NOT_AVAILABLE toolkitException in EnvelopeResponseHandler class for alignment with TS 43.019.	
			011	-	MEProfile getValue(short indexMSB, short indexLSB) method conformance requirement.	
			012	-	Addition of tests on Proactive Command Control for alignment with TS 43.019.	
			013	-	Correct in some script files wrong command qualifiers value for COMMAND DETAILS TLV and wrong source value and destination value for Device Identities TLV.	
			014	-	Correction to EnvelopeHandler getTheHandler() method test procedure description.	
			015	-	PRH getGeneralResult() method test: Unexpected API expectations.	
			016	-	Cross references insertion.	
TP-27	TP-050026	T3-050179	017		Correction of TP-DCS used for uncompressed 8 bits data SMS envelope.	5.3.0
CP-28	CP-050143	C6-050437	017	1	Correction of TP-DCS used for uncompressed 8 bits data SMS envelope	5.4.0
CP-28	CP-050143	C6-050438	018	1	Correction of FWK_ERP_EFSE script file syntax	5.4.0
CP-28	CP-050143	C6-050439	019	1	Correction of security level in API_2_ENH_GSDL test	5.4.0
CP-28	CP-050143	C6-050440	020		Restore files content in API_1_SVW_UPDRSBS_BSS_1.clr file	5.4.0
CP-29	CP-050334	C6-050722	021		Modification of the triggering event to test the post() and postAsBERTLV() methods	5.5.0
CP-30	CP-050670	C6-050902	022		Modification of the triggering event to test the post() and postAsBERTLV() methods	5.6.0
					replacement of incorrect Annex E	5.6.1
CP-61	CP-130529	C6-130396	0024	2	Modification of the statements on security parameters	5.7.0
CP-64	CP-140420	C6-140287	0027		Creation of Release 8	8.0.0

History

Document history		
V8.0.0	July 2014	Publication