ETSI TS 102 268 V7.0.0 (2015-04)

Smart Cards;
Test specification for the UICC
Application Programming Interface (API) for Java Card™
(Release 7)

Reference
RTS/SCP-T005v700

Keywords
API, smart card, testing

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

Contents

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://webapp.etsi.org/IPR/home.asp).

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 Project Smart Card Platform (SCP).

The contents of the present document are subject to continuing work within TC SCP and may change following formal TC SCP approval. If TC SCP decides to modify the contents of the present document, it will be re-released by TC SCP 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 TC SCP for information;
  - 2 presented to TC SCP for approval;
  - 3 or greater indicates TC SCP 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.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**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.

1 Scope

The present document covers the minimum characteristics considered necessary in order to provide compliance to ETSI TS 102 241 [9].

It describes the technical characteristics and methods for testing the UICC API for Java Card™ (ETSI TS 102 241 [9]) implemented in a UICC Platform. It specifies the following parts:

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

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

- In the case of a reference to a TC SCP document, a non specific reference implicitly refers to the latest version of that document in the same Release as the present document.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] Sun Microsystems Java Card™ Specification: "Java Card™ 2.2.2 Application Programming Interface".

[2] Sun Microsystems Java Card™ Specification: "Java Card™ 2.2.2 Runtime Environment (JCRE) Specification".

[3] Sun Microsystems Java Card™ Specification: "Java Card™ 2.2.2 Virtual Machine Specification".

NOTE: SUN Java Card Specifications can be downloaded at http://www.oracle.com/technetwork/java/javame/javacard/download/overview/index.html

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

[5] ETSI TS 102 221: "Smart cards; UICC-Terminal interface; Physical and logical characteristics".

[6] ETSI TS 102 223: "Smart cards; Card Application Toolkit (CAT)".

[7] ETSI TS 102 222: "Integrated Circuit Cards (ICC); Administrative commands for telecommunications applications".

[8] ETSI TS 102 226: "Smart Cards; Remote APDU structure for UICC based applications".

[9] ETSI TS 102 241: "Smart Cards; UICC Application Programming Interface (UICC API) for Java Card (TM)".

[10] ETSI TS 123 040: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Technical realization of Short Message Service (SMS) (3GPP TS 23.040)".

[11] ETSI TS 101 267: "Digital cellular telecommunications system (Phase 2+); Specification of the SIM Application Toolkit for the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface (3GPP TS 11.14)".

[12] ETSI TS 131 213: "Universal Mobile Telecommunications System (UMTS); LTE; Test specification for (U)SIM; Application Programming Interface (API) for Java Card™ (3GPP TS 31.213)".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

- In the case of a reference to a TC SCP document, a non specific reference implicitly refers to the latest version of that document in the same Release as the present document.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

Not applicable.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

applet installation parameters: values for applet installation parameters

Conformance Requirement Reference (CRR): description of the expected card behaviour according to ETSI TS 102 241 [9]

expected state: state in which the UICC 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, CAT RE behaviour, etc.) of the ETSI TS 102 241 [9]

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

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

test source file: java file containing methods that will load and install test applet in the card, execute and verify the test results, and restore the Default Initial Conditions on the UICC (when possible)

test toolkit applet: applet designed to test a specific functionality of the UICC API (ETSI TS 102 241 [9])

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ADF	Application Dedicated File
AID	Application IDentifier
APDU	Application Protocol Data Unit
API	Application Programming Interface
ARR	Access Rule Reference
ATR	Answer To Reset
BER	Basic Encoding Rules
CAP	Converted Applet
CARU	Cyclic Always Read and Update
CAT	Card Application Toolkit
CAT RE	Card Application Toolkit Runtime Environment
CR	Comprehension Required
CRE	CAT Runtime Environment
CRR	Conformance Requirements Reference
CRRC	Conformance Requirement Reference Context Error
CRRN	Conformance Requirement Reference Normal
CRRP	Conformance Requirement Reference Parameter Error
DCS	Data Coding Scheme
DF	Dedicated File
DIR	DIRectory
EF	Elementary File
FCP	File Control Parameters
FID	File IDentifier
GSM	Global System for Mobile communications
ICCID	Integrated Circuit Card IDentification
JCRE	Java Card™ Runtime Environment
LCSI	Life Cycle Status Information
ME	Mobile Equipment
MF	Master File
MSL	Minimum Security Level
MT	Mobile Terminated
NAA	Network Access Application
NOK	Not OK
PIN	Personal Identification Number
PL	Preferred Languages
RAPDU	Response Application Protocol Data Unit
RFU	Reserved for Future Use
SDK	Software Development Kit
SE	Security Environment
SFI	Short File Identifier
STK	SIM ToolKit
SW	Status Word
TAR	Toolkit Application Reference
TARU	Transparent Always Read and Update
TLV	Tag Length Value
TP	Transfer layer Protocol

4 Test environment

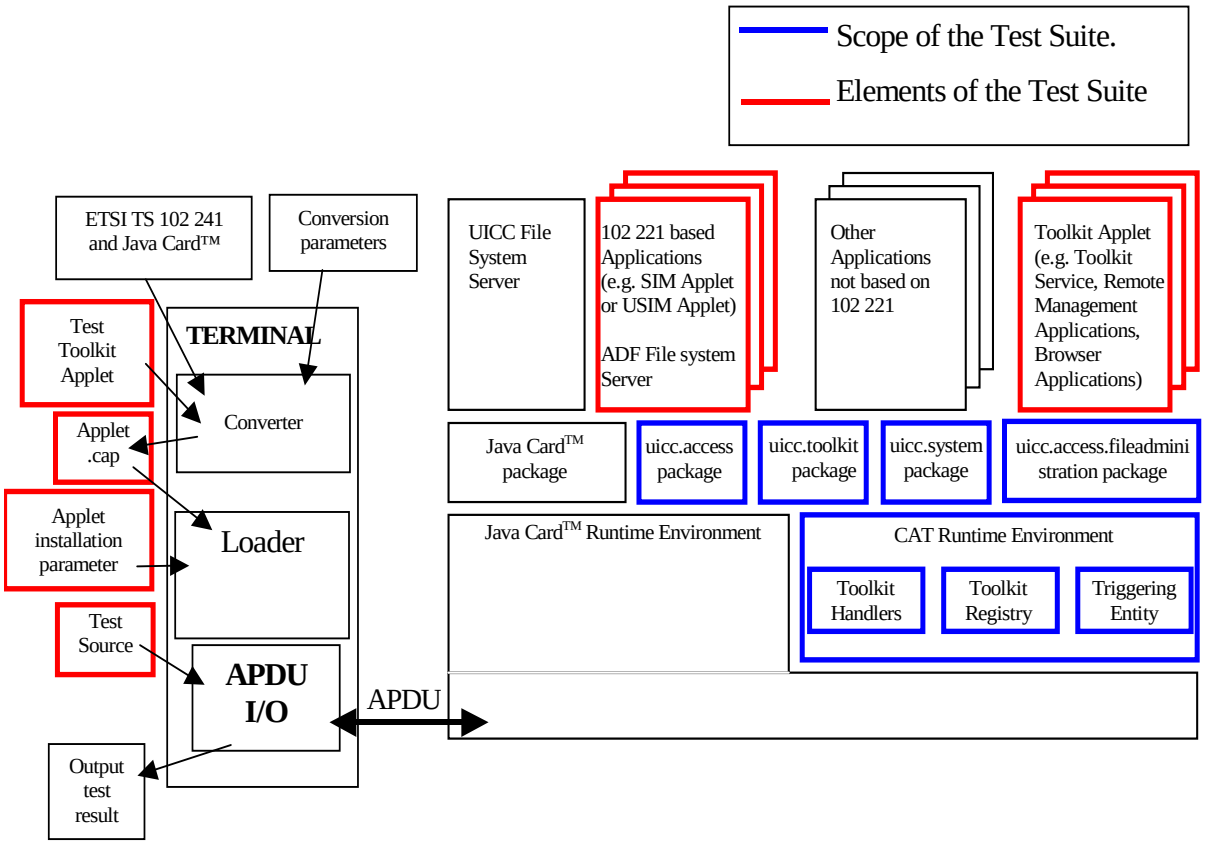
This clause specifies requirements that shall be met and the testing rules that shall be followed, during the test procedure.

4.1 Applicability

The test defined in the present document are applicable to cards implementing ETSI TS 102 241 [9] unless otherwise stated.

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]_[class name]_[method name]' where:

- package name:

- uicc.access package: '1'.
- uicc.toolkit package: '2'.
- uicc.system package: '3'.
- uicc.access.fileadministration: '4'.

- class/interface name:
  - yyy: 3 letters for each class.

NOTE 1: See annex A for full classes acronyms list.

- method name:
  - zzzz[input parameters].

NOTE 2: See annex A for full methods name acronyms list.

CRE: Cat Runtime Environment testing: 'Cre_[Clause name]_[Subclause name]':

- Clause name:
  - xxx: 3 letters for each clause

NOTE 3: See annex A for full clause acronyms list.

- Subclause name
  - yyyy: : 4 letters for each subclause

NOTE 4: See annex A for full subclause acronyms list.

4.3.1.1 Conformance requirements

The conformance requirements are expressed in the following way:

- Method prototype as listed in ETSI TS 102 241 [9].
- Normal execution:
  - Contains normal execution and correct parameters limit values, each referenced as a Conformance Requirement Reference Normal (CRRN).
- Parameter errors:
  - Contains parameter errors and incorrect parameter limit values, each referenced as a Conformance Requirement Reference Parameter Error (CRRP).
- Context errors:
  - 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 Source: Test_[Test Area Reference].java.
- Test Applet: [Test Area Reference]_[Test applet number].java.
- Cap File: [Test Area Reference].cap.

The applet numbers start from '1'.

The test source shall use common interfaces defined in annex D.

The Cap File format is described in Java Card™ Virtual Machine Specification [3].

Test files can be run in any order.

All files from the same test area are located in the same subfolder.

4.3.1.3 Test coverage

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

4.3.1.4 Test procedure

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

Test Case			
Id	**Description**	**API/CAT RE Expectation**	**APDU Expectation**
	Test Case detailed description	*API and/or CAT RE expected behaviour.*	*Expected response at APDU level.*

4.4 Initial conditions

The Initial Conditions are a set of general prerequisites for the UICC 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 B;
- 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:

uicc.test.access.[Test Area Reference]: Java Card packages containing Test Area References for the ETSI TS 102 241 [9] uicc.access package.

uicc.test.system.[Test Area Reference]: Java Card packages containing Test Area References for the ETSI TS 102 241 [9] uicc.system package.

uicc.test.toolkit.[Test Area Reference]: Java Card packages containing Test Area References for the ETSI TS 102 241 [9] uicc.toolkit package.

uicc.test.access.fileadministration.[Test Area Reference]: Java Card packages containing Test Area References for the ETSI TS 102 241 [9] uicc.access.fileadministration package.

uicc.test.catre.[Test Area Reference]: Java Card packages containing Test Area References for the ETSI TS 102 241 [9] CAT Runtime Environment.

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

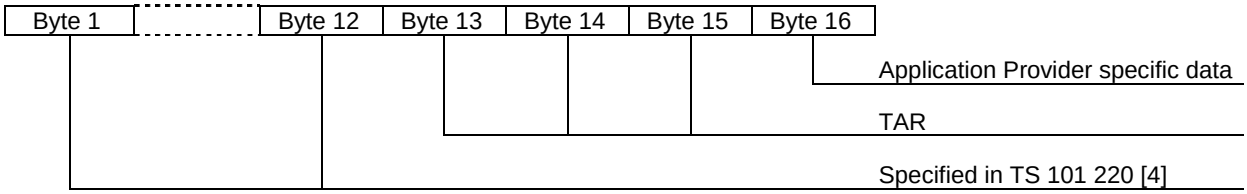
where the Test Area Reference is written in lower case.

EXAMPLE: The package *../uicc.test.access.[Test Area Reference]* creates the following directory structure *../uicc/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]*.

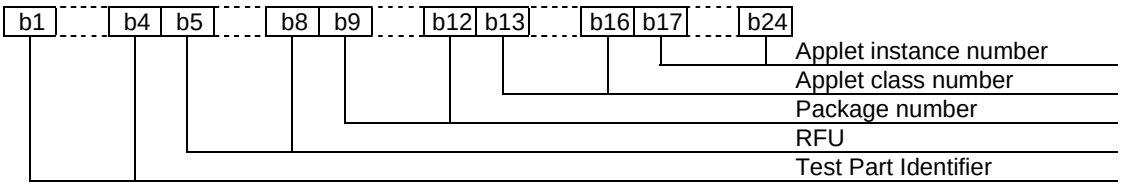
4.6 AID Coding

The AID coding for the Test Packages, Applet classes and Applets shall be as specified in ETSI TS 101 220 [4]. In addition, the following TAR and Application Provider specific data values are defined for use within the present document:

AID coding



TAR Coding (3 bytes/ 24 bits):



Applet instance number, Applet Class number, Package number:

- For package AID, package number shall start from 0 and class and instance numbers shall be 0.
- For class AID, package number is the number of the class package, class number shall start from 1 and instance shall be 0.
- For instance AID, package and class number are the number of class and package of which instance belongs, and instance number shall start from 1.

Test part Identifier (bits b1-b4):

- 0000 reserved (as TAR= '00.00.00' is reserved for Issuer Security Domain).
- 0001 API uicc.access.
- 0010 API uicc.toolkit.
- 0011 API uicc.system.
- 0100 API uicc.access.fileadministration.
- 0101 CAT RE.
- 1101 ADF2.
- 1110 ADF1.
- 1111 uicc.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 uicc.util is 'A0 00 00 00 09 00 05 FF FF FF FF 89 F0 00 00 00'.

4.7 Test equipment

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

4.7.1 Test tool

This test tool shall meet the following requirements:

- be able to send and receive APDU command to the UICC;
- the result of the I/O commands must be presented at the application layer;
- be able to provide results of the tests;
- shall send and/or compare all data specified in test file.

4.7.2 Interfaces and classes use

The test tool shall use some interfaces and classes, defined in annex D. They define the only allowed methods to write the test sources.

Interfaces and classes are defined as follow:

- UiccAdministrativeCommandsService defines administrative methods from ETSI TS 102 222 [7];
- UiccApplicationManagementService defines methods to load, install, select and delete applications;
- UiccCardManagementService defines methods to manage the card and its files;
- UiccToolkitService defines methods to manage toolkit commands;
- APDUResponse defines method to retrieve and check status words and data received from the card;
- UiccAPITestCardService defines the static method to get a reference of the class implementing all interfaces;
- UiccTestModel is an abstract class which shall be extended by every test source class; it defines the entry point run() method of the test script.

4.7.3 Util package

Annex D includes java source code of TestToolkitApplet abstract class of the uicc.util package. Each test applet shall extend this abstract class in order to retrieve test results when selecting it.

4.7.4 Java Software Development kit version

Java software development kit (SDK) version supported by JavaCard 2.2.2 specifications ([1], [2], [3]) is 1.4.1.

5 Test plan

5.1 Package uicc.access

5.1.1 Interface FileView

5.1.1.1 Method activateFile

Test Area Reference: Api_1_Fvw_Actf

5.1.1.1.1 Conformance requirement

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

```
public void activateFile()
               throws UICCException
```

5.1.1.1.1.1 Normal execution

- CRRN1: The currently selected EF of the calling applet shall be activated, as defined in ETSI TS 102 222 [7].

5.1.1.1.1.2 Parameter errors

No requirements.

5.1.1.1.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the calling applet does not fulfil the access condition, activate, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.
- CRRC3: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.1.1.1.2 Test area files

Test Source: Test_Api_1_Fvw_Actf.java.

Test Applet: Api_1_Fvw_Actf_1.java.

Cap File: api_1_fvw_actf.cap.

5.1.1.1.3 Test coverage

CRR number	Test case number
N1	2,3
C1	1
C2	4
C3	Not testable

5.1.1.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Test applet is installed with no access right on Application Pin2		
1	**No EF is selected** 1- Select DF_{TEST} fid=1111 2- Call activateFile()	2- A UICCException NO_EF_SELECTED is thrown	
2	**Activate deactivated File** 0- Select Root directory 1- Select EF_{UICC} fid=2FFF 2- ReadBinary EF_{UICC} 3- Deactivate EF_{UICC} 4- ReadBinary EF_{UICC} 5 -ActivateFile EF_{UICC} 6- ReadBinary EF_{UICC}	2- No Exception shall be thrown 4- UICCException.REF_DATA_INVALIDATED is thrown 6- No Exception shall be thrown	
3	**Activate activated File** ActiveFile EF_{UICC}	No Exception shall be thrown	
4	**Access condition not fulfilled** 1- Select DF_{TEST} fid=1111 2- Select EF_{LADA} fid=6F15 3- ActivateFile EF_{LADA}	3- A UICCException SECURITY_STATUS_NOT_SATISFIED is thrown	

5.1.1.2 Method deactivateFile

Test Area Reference: Api_1_Fvw_Dacf.

5.1.1.2.1 Conformance requirement

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

```
public void deactivateFile()
                    throws UICCException
```

5.1.1.2.1.1 Normal execution

- CRRN1: The currently selected EF of the calling applet shall be deactivated, as defined in ETSI TS 102 222 [7].

5.1.1.2.1.2 Parameter errors

No requirements.

5.1.1.2.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the calling applet does not fulfil the access condition, activate, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.
- CCRC3: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.1.1.2.2 Test area files

Test Source: Test_Api_1_Fvw_Dacf.java.

Test Applet: Api_1_Fvw_Dacf_1.java.

Cap File: api_1_fvw_dacf.cap.

5.1.1.2.3 Test coverage

CRR number	Test case number
N1	2,3
C1	1
C2	4
C3	Not testable

5.1.1.2.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Test applet is installed with no access right on Application Pin2		
1	**No EF is selected** 1- select DF_{TEST} fid=1111 2- call deactivateFile()	2- An UICCException NO_EF_SELECTED is thrown	
2	**Deactivate activated File** 0- Select root directory 1- Select EF_{UICC} fid=2FFF 2- ReadBinary EF_{UICC} 3- Deactivate EF_{UICC} 4- ReadBinary EF_{UICC}	2- No Exception shall be thrown 4- UICCException.REF_DATA_INVALIDATED is thrown	
3	**Deactivate deactivated File** 1- deactivateFile EF_{UICC} 2- activateFile EF_{UICC}	1- No Exception shall be thrown	
4	**Access condition not fulfilled** 1- select DF_{TEST} fid=1111 2- select EF_{LADA} fid=6F15 3- deactivateFile EF_{LADA}	3- An UICCException SECURITY_STATUS_NOT_SATISFIED is thrown	

5.1.1.3 Method increase

Test Area Reference: Api_1_Fvw_Incr.

5.1.1.3.1 Conformance requirement

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

```
public short increase(byte[] incr,
                      short incrOffset,
                      short incrLength,
                      byte[] resp,
                      short respOffset)
               throws java.lang.NullPointerException,
                      java.lang.ArrayIndexOutOfBoundsException,
                      UICCException
```

5.1.1.3.1.1 Normal execution

- CRRN1: This method increases the current cyclic EF record.
- CRRN2: The response buffer will only contain the value of the increased record.

5.1.1.3.1.2 Parameter errors

- CRRP1: If the array incr is null, an instance of NullPointerException shall be thrown.
- CRRP2: If the array resp is null, an instance of NullPointerException shall be thrown.
- CRRP3: If incrOffset is negative, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP4: If incrLength is negative, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP5: If respOffset is negative, an instance of ArrayIndexOutOfBoundsExceptoin shall be thrown.
- CRRP6: If incrOffset plus incrLength, is greater than the length of array incr, an instance of ArrayIndexOutOfBoundsException shall be thrown and no increase is performed.
- CRRP7: If respOffset is greater than the length of array resp, an instance of ArrayIndexOutOfboundsException shall be thrown.
- CRRP8: 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 UICCException shall be thrown. The reason code shall be UICCException.MAX_VALUE_REACHED.
- CRRP9: If incrLength is greater than 127, and exception shall be thrown.

5.1.1.3.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.
- CRRC3: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.
- CRRC4: If increase not allowed as indicated by the File descriptor byte of the File Descriptor (ETSI TS 102.221 [5] Response for an EF), or the file is not a cyclic one, an instance of the UICCException shall be thrown. The reason code shall be UICCException.COMMAND_INCOMPATIBLE.
- CRRC5: If the calling applet does not fulfil the access condition, INCREASE, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.
- CRRC6: If the currently selected EF is invalidated, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF_DATA_INVALIDATED.

- CRRC7: If the currently selected cyclic EF has no record, an instance of UICCException shall be thrown. The reason code shall be UICCException. RECORD_NOT_FOUND.

5.1.1.3.2 Test area files

Test Source: Test_Api_1_Fvw_Incr.java.

Test Applet: Api_1_Fvw_Incr_1.java.

Cap File: api_1_fvw_incr.cap.

5.1.1.3.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 16
N2	2, 3, 16
P1	4
P2	9
P3	6
P4	5
P5	10
P6	7
P7	11
P8	8
P9	16
C1	1
C2	Not testable
C3	Not testable
C4	12
C5	13
C6	14
C7	15

5.1.1.3.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**No EF selected** 1- select DF_{TEST} fid=1111 2- byte[] incr = new byte[4] byte[] resp = new byte[4] incrOffset = 0 incrLength = 2 respOffset = 0 increase()	2- An UICCException.NO_EF_SELECTED should be thrown	

2	**increase , verify response**
	
	1- select $EF_{CARU,}$ fid=6F09
	set the record pointer with readRecord() in PREVIOUS mode
	2-//Set both record to 00 00 00
	mode = REC_ACC_MODE_PREVIOUS
	data[] = {0x00,0x00,0x00}
	recOffset = 0
	dataOffset = 0
	dataLength = 3
	updateRecord() //update Record 1
	updateRecord() //update Record 2
	3- incr[] = {0x00,0x00,0x01}
	incrOffset = 0
	incrLength = 3
	resp.length = 4
	respOffset = 0
	ret = 3
	increase()
		3- resp[] = {0x00,0x00,0x01,0x00}	

Id	Description	API Expectation	APDU Expectation
3	**increase, verify file** 1- incr[]={0x00,0x00,0x00,0x02} incrOffset = 1 incrLength = 3 resp.length = 4 respOffset = 1 increase() 2- resp[] = {0x00,0x00,0x00,0x00} recNumber = 0 mode = REC_ACC_MODE_ABSOLUTE_CURRENT recOffset = 0 resp.length =4 respOffset = 0 respLength = 3 readRecord()	1- resp[] = {0x00,0x00,0x00,0x03} 2- resp[] = {0x00,0x00,0x03,0x00}	
4	**incr[] is null** incr[] = null incrOffset = 0 incrLength = 1 resp.length = 4 respOffset = 0 increase()	Shall throw java.lang.NullPointerException.	
5	**incrLength< 0** incr.length = 4 incrOffset = 0 incrLength = -1 resp.length = 4 respOffset = 0 increase()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
6	**incrOffset < 0** incr.length = 4 incrOffset = -1 incrLength = 1 resp.length = 4 respOffset = 0 increase()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
7	**IncrOffset + incrLength > incr.length** incr.length = 4 incrOffset = 1 incLength = 4resp.length = 4 respOffset = 0 increase()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	

8	**Reach Maximum Value**
	
	1- incr[0:3] = 0xFF
	incrOffset = 0
	incrLength = 3
	resp.length = 4
	respOffset = 0
	increase()
	
	2- //Set both record to FF FF FF
	mode = REC_ACC_MODE_PREVIOUS
	data[] = {0xFF,0xFF,0xFF}
	recOffset = 0
	dataOffset = 0
	dataLength = 3
	updateRecord() //update Record 1
	updateRecord() //update Record 2
	
	3- incr[] = {0x00,0x00,0x01}
	incrOffset = 0
	incrLength = 3
	resp.length = 4
	respOffset = 0
	increase()
		1- Shall throw uicc.access.UICCException with reason code MAX_VALUE_REACHED.
		
		2- Shall throw uicc.access.UICCException with reason code MAX_VALUE_REACHED.	

Id	Description	API Expectation	APDU Expectation
9	**resp[] is null** incr.length = 4 incrOffset = 0 incrLength = 1 resp[] = null respOffset = 0 increase()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
10	**respOffset < 0** incr.length = 4 incrOffset = 0 incrLength = 1 resp.length = 4 respOffset = -1 increase()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
11	**respOffset + recordLength > resp.length** incr.length = 4 incrOffset = 0 incrLength = 3 resp.length = 3 respOffset = 2 increase()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
12	**EF is not Cyclic** 1- select EF_{TARU} fid= 6F03 2- incr.length= 3 incrOffset = 0 incrLength = 3 resp.length = 3 respOffset = 0 increase() 3 - select EF_{LARU}, fid=6F0C 4 - incr.length= 3 incrOffset = 0 incrLength = 3 resp.length = 3 respOffset = 0 increase()	2 - Shall throw uicc.access.UICCException with reason code COMMAND_INCOMPATIBLE. 4 - Shall throw uicc.access.UICCException with reason code COMMAND_INCOMPATIBLE.	
13	**Access condition not fulfilled** 1- select EF_{CNIC}, fid=6F06 2- incr.length= 3 incrOffset = 0 incrLength = 3 resp.length = 3 respOffset = 0 increase()	2 - Shall throw uicc.access.UICCException with reason code SECURITY_STATUS_NOT_SATISFIED.	
14	**EF is invalidated** select EF_{CARU}, fid=6F09 2 - invalidate() 3 - incr.length= 3 incrOffset = 0 incrLength = 3 resp.length = 3 respOffset = 0 increase() 4 - rehabilitate() 5- Restore initial content of EF_{CARU}	3 - Shall throw uicc.access.UICCException with reason code REF_DATA_INVALIDATED	
15	**Record not found** 1- Create an EF Cyclic with no record in folder DF_{TEST}, fid=0x2C00 2- select EF Cyclic with no record, fid=0x2C00 3- incr.length= 3 incrOffset = 0 incrLength = 3 resp.length = 3 respOffset = 0 increase() 4- Delete Cyclic EF with no records.	3- Shall throw uicc.access.UICCException with reason code RECORD_NOT_FOUND	

Id	Description	API Expectation	APDU Expectation
16	**incrLength out of range** 1- Create an EF Cyclic with 1 record of 0xFD length in folder DF_{TEST}, fid=0x2CFD 2- Select EF Cyclic, fid=0x2CFD 3- Set record to following value rec[0..125] = 0; rec[126..252] = 0xFF with an update record. 4- incr.length=128 incrOffset = 1 incrLength = 127 resp.length = 253 respOffset = 0 Incr[] initialized to = {0x00,…,0x00,0x01} respOffset = 0 ret = 0xFD increase() 5- incr.length=128 incrOffset = 0 incrLength = 128 resp.length = 128 respOffset = 0 Incr[] initialized to 0 respOffset = 0 increase() 6- Delete EF Cyclic with fid=0x2CFD.	4- resp[0..125] = {0x00,…,0x00,0x01} and resp[126..252] = {0x00,…,0x00} 5- Shall throw an exception	

5.1.1.4 Method readBinary

Test Area Reference: Api_1_Fvw_Redb.

5.1.1.4.1 Conformance requirement

The method with 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,
                        UICCException
```

5.1.1.4.1.1 Normal execution

- CRRN1:. Reads the data bytes of the current transparent EF, as defined in ETSI TS 102 221 [5].
- CRRN2: The sum of respOffset plus respLength is returned. and the data bytes of the currently selected transparent file are returned in resp.

5.1.1.4.1.2 Parameter errors

- CRRP1: If fileOffset is negative, an instance of UICCException.OUT_OF_FILE_BOUNDARIES shall be thrown.
- CRRP2: If respOffset is negative, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP3: If respLength is negative, an instace of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP4: If the array resp is null, an instance of NullPointerException shall be thrown.
- CRRP5: If respOffset plus respLength is greater than the length of the array resp.length, an instance of ArrayIndexOutOfBoundsException shall be thrown and no read is performed.

- CRRP6: If fileOffset plus respLength exceeds the length of the file, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT_OF_FILE_BOUNDARIES.

5.1.1.4.1.3 Context errors

- CRRC1: If the method call causes an error to occur that is not expected and thus not handled, an instace of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.
- CRRC2: If the currently selected EF is not transparent, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND_INCOMPATIBLE.
- CRRC3: If the calling applet does not fulfil the access condition, READ, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.
- CRRC4: If the currently selected EF is deactivated and the file status of the EF does not allow for the reading of an deactivated file, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF_DAT_INVALIDATED.
- CRRC5:If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.

5.1.1.4.2 Test area files

Test Source: Test_Api_1_Fvw_Redb.java.

Test Applet: Api_1_Fvw_Redb_1.java.

Cap File: api_1_fvw_redb.cap.

5.1.1.4.3 Test coverage

CRR number	Test case number
N1	1, 2
N2	1, 2
P1	3
P2	6
P3	7
P4	5
P5	8
P6	4
C1	Not testable
C2	9
C3	10
C4	11
C5	12

5.1.1.4.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Read from EF_{TARU}** 1- select DF_{TEST}, fid=1111 select EF_{TARU}, fid=6F03 2- fileOffset = 0 resp.length = 260 resp[0:259] = 0x55 respOffset = 10 respLength = 250 readBinary()	2 - shall return 20. resp shall contain the contents of EF_{TARU} starting at index 10. <Description of resp: 55 55 55 55 55 55 55 55 55 55 FF FF FF FF FF FF .. .. FF>	

Id	Description	API Expectation	APDU Expectation
2	**Read from EF_{TARU}** fileOffset = 5 resp.length = 260 resp[0:259] = 0x55 respOffset = 10 respLength = 5 readBinary()	shall return 15 resp shall contain the last 5 bytes of EF_{TARU} 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 >	
3	**FileOffset is negative** fileOffset = -1 respOffset = 0 respLength = 10 readBinary()	Shall throw uicc.access.UICC Exception with reason code OUT_OF_FILE_BOUNDARIES.	
4	**FileOffset + respLength > EF length** fileOffset = 259 respOffset = 0 respLength = 2 readBinary()	Shall throw uicc.access.UICC Exception with reason code OUT_OF_FILE_BOUNDARIES.	
5	**resp[] is null** fileOffset = 0 resp = null 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 resp.length = 20 respOffset = 10 respLength = 11 readBinary()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
9	**EF is not Transparent** 1- select EF_{LARU}, fid=6F0C 2- fileOffset = 0 respOffset = 0 respLength = 1 readBinary()	2 - Shall throw uicc.access.UICC Exception with reason code COMMAND_INCOMPATIBLE.	
10	**Access condition not fulfilled** 1- select EF_{TRAC}, fid=6F0E 2- fileOffset = 0 respOffset = 0 respLength = 1 readBinary()	2- Shall throw uicc.access.UICC Exception with reason code SECURITY_STATUS_NOT_SATISFIED.	
11	**EF is deactivated** 1 - select EF_{TARU}, fid=6F03 2 - deactivateFile() 3 - readBinary() 4 - activateFile())	3 - Shall throw uicc.access.UICCException with reason code REF_DATA_INVALIDATED.	
12	**No EF selected** 1- select DF_{TEST} fid=1111 2 readBinary()	2 - Shall throw uicc.access.UICCException with reason code NO_EF_SELECTED.	

5.1.1.5 Method readRecord

Test Area Reference: Api_1_Fvw_Redr.

5.1.1.5.1 Conformance requirement

The method with 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,
                        UICCException
```

5.1.1.5.1.1 Normal execution

- CRRN1: Reads a record or a part of record of a current linear fixed or cyclic EF into byte array resp and the sum of respOffset plus respLength is returned.
- CRRN2: If the access mode is REC_ACC_MODE_CURRENT the current record will be read and the current record pointer shall not be changed.
- CRRN3: If the access mode is REC_ACC_MODE_ABSOLUTE the record addressed by recNumber will be read and the current record pointer shall not be changed.
- CRRN4: If the access mode is REC_ACC_MODE_NEXT the next record relative to the current selected record will be selected and read. The record pointer will be incremented.
- CRRN5: If the access mode is REC_ACC_MODE_NEXT and no current record is selected, the first record will be selected and read. The record pointer will be incremented.
- CRRN6: If the access mode is REC_ACC_MODE NEXT and the current record pointer, of a cyclic EF, is set to the last record, the record pointer is set to the first record and the record is read.
- CRRN7: If the access mode is REC_ACC_MODE_PREVIOUS the previous record relative to the current selected record will be selected and read.
- CRRN8: If the access mode is REC_ACC_MODE PREVIOUS and no current record is selected, the last record will be selected and read.
- CRRN9:If the access mode is REC_ACC_MODE_PREVIOUS and the current record pointer of a cyclic EF is set to the first record, the record pointer is set to the last record in this EF and this record shall be read.
- CRRN10: The current record pointer of any other applet shall not be changed.

5.1.1.5.1.2 Parameter errors

- CRRP1: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_ABSOLUTE and recNumber is less than 0 or greater than records available, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD_NOT_FOUND.
- CRRP2: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_CURRENT, recNumber is 0 and there is no current record selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD_NOT_FOUND.
- 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 UICCException shall be thrown. The reason code shall be UICCException.RECORD_NOT_FOUND.
- 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 UICCException shall be thrown. The reason code shall be UICCException.RECORD_NOT_FOUND.
- CRRP5: If the specified offset into the selected record recOffset is less than 0, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT_OF_RECORD_BOUNDARIES.

- CRRP6: If recOffset plus respLength is greater than the record length, an instance of UICCException shall be thrown. The reason code shall be UICC Exception.OUT_OF_RECORD_BOUNDARIES.
- CRRP7: If the access mode is not between 2 and 4 (2 = REC_ACC_MODE_NEXT, etc.), an instance of UICCException shall be thrown. The reason code shall be UICC Exception.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.

5.1.1.5.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the currently selected EF is neither linear fixed nor cyclic, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND_INCOMPATIBLE.
- CRRC3: If the calling applet does not fulfil the access condition, READ, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.
- 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 UICCException shall be thrown. The reason code shall be UICCException.REF_DATA_INVALIDATED.
- CRRC5: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.1.1.5.2 Test area files

Test Source: Test_Api_1_Fvw_Redr.java.

Test Applet: Api_1_Fvw_Redr_1.java.

Cap File: api_1_fvw_redr.cap.

5.1.1.5.3 Test coverage

CRR number	Test case number
N1	2, 3, 4, 5,6, 8, 9, 10, 11,12,13
N2	3, 9
N3	2, 8
N4	4, 5, 10, 11
N5	4, 11
N6	11
N7	6, 7, 12, 13
N8	6, 13
N9	12
N10	
P1	14
P2	15
P3	5
P4	7
P5	16
P6	17
P7	18
P8	19
P9	20

CRR number	Test case number
P10	21
P11	22
C1	1
C2	23
C3	24
C4	25
C5	Not testable

5.1.1.5.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**No EF selected** 1- select DF_{TEST},fid=1111 2- recNumber = 1 mode = REC_ACC_MODE_ABSOLUTE recOffset = 0 byte[] resp = new byte[20] respOffset = 0 respLength = 10 readRecord()	2-Shall throw uicc.access.UICCException with reason code NO_EF_SELECTED.	
2	**Read Absolute from Linear Fixed EF** 1 - select EF_{LARU}, fid=6F0c // Record pointer not set. 2 - recNumber = 2 mode = REC_ACC_MODE_ABSOLUTE recOffset = 0 respOffset = 0 respLength = 4 readRecord() 3- recNumber = 0 mode = REC_ACC_MODE_NEXT recOffset = 0 respOffset = 0 respLength = 4 readRecord()	2 - resp shall be: resp={0xAA,0xAA,0xAA,0xAA} 3- resp shall be: Resp={0x55,0x55,0x55,0x55}	
3	**Read Current from Linear Fixed EF** //record pointer shall not be changed 1- recNumber = 0 mode = REC_ACC_MODE_CURRENT recOffset = 0 respOffset = 0 respLength = 4 readRecord()	resp shall be: resp={0x55,0x55,0x55,0x55}	

4	**Read Next from Linear Fixed EF**
	1- select EF_{LARU}, fid=6F0c
	//no record selected
	recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	respOffset = 0
	respLength = 4
	readRecord()
	2- recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	respOffset = 0
	respLength = 4
	readRecord()
		1- resp shall be:
		resp={0x55,0x55,0x55,0x55}
		
		2- resp shall be:
		resp={0xAA,0xAA,0xAA,0xAA}	

5	**Read Next from Linear Fixed EF** recNumber = 0 mode = REC_ACC_MODE_NEXT recOffset = 0 respOffset = 0 respLength = 4 readRecord()	Shall throw uicc.access.UICC Exception with reason code RECORD_NOT_FOUND.	

Id	Description	API Expectation	APDU Expectation

6	**Read Previous from Linear Fixed EF**
	1- recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	respOffset = 0
	respLength = 4
	readRecord()
	2- select EF_{LARU}, fid=6F0c
	//no record selected
	recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	respOffset = 0
	respLength = 4
	readRecord()
		1- resp shall be:
		resp={0x55,0x55,0x55,0x55}
		
		2- resp shall be:
		resp={0xAA,0xAA,0xAA,0xAA}	

7	**Read Previous from Linear Fixed EF** recNumber = 0 mode = REC_ACC_MODE_PREVIOUS recOffset = 0 respOffset = 0 respLength = 4 readRecord()	Shall throw uicc.access.UICCException with reason code RECORD _NOT_FOUND.	
8	**Read Absolute from Cyclic EF** 1 select EF_{CARU}, fid = 6F09 2- recNumber = 2 mode = REC_ACC_MODE_ABSOLUTE recOffset = 0 respOffset = 0 respLength = 3 readRecord() 3- recNumber = 1 readRecord()	2 - resp shall be: resp={0xAA,0xAA,0xAA} 3 - resp shall be: resp={0xAA,0xAA,0xAA}	

9	**Read Current from Cyclic EF**
	//record pointer shall not be changed
	//from testcase before
	1- recNumber = 0
	mode = REC_ACC_MODE_CURRENT
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
		1- resp shall be:
		resp={0xAA,0xAA,0xAA}	

10	**Read Next from Cyclic EF** recNumber = 0 mode = REC_ACC_MODE_NEXT recOffset = 0 respOffset = 0 respLength = 3 readRecord()	resp shall be: resp={0xAA,0xAA,0xAA}	

11	**Read Next from Cyclic EF**
	1- recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
	2- select EF_{CARU}, fid = 6F09
	//no rec selected
	recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
		1- resp shall be:
		resp={0x55,0x55,0x55}
		
		2- Shall throw uicc.access.UICCException with reason code RECORD_NOT_FOUND.	

12	**Read Previous from Cyclic EF** 1- recNumber = 0 mode = REC_ACC_MODE_PREVIOUS recOffset = 0 respOffset = 0 respLength = 3 readRecord()	1- resp shall be: resp={0xAA,0xAA,0xAA}	

Id	Description	API Expectation	APDU Expectation

13	**Read Previous from Cyclic EF**
	1- recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
	2- select EF_{CARU}, fid = 6F09
	// no rec selected
	recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
		1- resp shall be:
		resp={0x55,0x55,0x55}
		
		2- resp shall be:
		resp={0xAA,0xAA,0xAA}	

14	**Read Absolute from Linear Fixed EF beyond Records** 1- select EF_{LARU}, fid=6F0C 2- recNumber = -1 mode = REC_ACC_MODE_ABSOLUTE recOffset = 0 respOffset = 0 respLength = 4 readRecord() 3- recNumber = 3 readRecord()	2- Shall throw an uicc.access.UICCException with reason code UICCException.RECORD_NOT_FOUND. 3- Shall throw an uicc.access.UICCException with reason code UICCException.RECORD_NOT_FOUND.	

15	**No current record in linear fixed EF, read current**
	1- select EF_{LARU}, fid=6F0C // No current record
	2- recNumber = 0 // curr rec
	mode = REC_ACC_MODE_CURRENT
	recOffset = 0
	respOffset = 0
	respLength = 4
	readRecord()
		2 - Shall throw uicc.access.UICC Exception with reason code RECORD_NOT_FOUND.	

16	**recOffset < 0**
	1- select EF_{LARU}, fid=6F0C
	2- recNumber = 1 // rec 1
	mode = REC_ACC_MODE_ABSOLUTE
	recOffset = -1
	respOffset = 0
	respLength = 4
	readRecord()
		2 - Shall throw uicc.access.UICCException with reason code OUT_OF_RECORD_BOUNDARIES.	

17	**recOffset + respLength > Record Length** 1- select EF_{LARU}, fid=6F0C 2- recNumber = 1 mode = REC_ACC_MODE_ABSOLUTE recOffset = 2 respOffset = 0 respLength = 4 readRecord()	2 - Shall throw sim.access.SIMViewException with reason code OUT_OF_RECORD_BOUNDARIES.	
18	**Reading with invalid mode** 1- select EF_{LARU}, fid=6F0C 2- recNumber = 0 mode = 1 recOffset = 0 respOffset = 0 respLength = 4 readRecord() 3- mode = 5 readRecord()	2 - Shall throw uicc.access.UICCException with reason code INVALID_MODE 3 - Shall throw uicc.access. UICCException with reason code INVALID_MODE.	
19	**resp is null** resp[] = null mode = REC_ACC_MODE_CURRENT respOffset = 0 respLength = 10 readRecord()	Shall throw java.lang.NullPointerException.	
20	**respOffset < 0** respOffset = -1 respLength = 10 readRecord ()	Shall throw java.lang.ArrayIndexOutOfBoundsException.	

Id	Description	API Expectation	APDU Expectation
21	**respLength < 0** respOffset = 0 respLength = -1 readRecord ()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
22	**respOffset + respLength > resp.length** respOffset = 10 respLength = 11 readRecord ()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
23	**EF is neither Cyclic nor Linear Fixed** 1- select EF_{TNU}, fid=6F02 2- respOffset = 0 respLength = 4 readRecord()	2 - Shall throw uicc.access.UICCException with reason code COMMAND_INCOMPATIBLE.	
24	**Access condition not fulfilled** 1- select EF_{CNR}, fid=6F04 2 - respLength = 3 readRecord()	2 - Shall throw uicc.access.UICCException with reason code SECURITY_STATUS_NOT_SATISFIED.	
25	**EF is deactivated** 1 - select EF_{CNU}, fid=6F05 deactivateFile() 2 - readRecord() 3 - activateFile	2 - Shall throw uicc.access.UICC Exception with reason code REF_DATA_INVALIDATED	

5.1.1.6 Method searchRecord

Test Area Reference: Api_1_Fvw_Sear.

5.1.1.6.1 Conformance requirement

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

```
public short searchRecord(byte mode,
                          short recordNum,
                          short searchIndication,
                          byte[] patt,
                          short pattOffset,
                          short pattLength,
                          short[] response,
                          short respOffset,
                          short respLength)
                   throws java.lang.NullPointerException,
                          java.lang.ArrayIndexOutOfBoundsException,
                          UICCException
```

5.1.1.6.1.1 Normal execution

- CRRN1: Search a given pattern in byte array patt[] of a current linear fixed or cyclic EF.
- CRRN2: If the pattern is found, the number of each record is stored in byte array response[] and the total number of updated bytes in the array response[] buffer is returned.
- CRRN3: If the value of respLength is greater than the number of records found, the whole response is copied into the response buffer and the number of elements copied is returned by the method.
- CRRN4: If the value of respLength is smaller than the number of found patterns, the first record numbers are copied into the response array and the value of respLength is returned.
- CRRN5: If mode is SIMPLE_SEARCH_START_FORWARD, the search starts at the given record number forward towards the end of the file.
- CRRN6: If mode is SIMPLE_SEARCH _START_BACKWARD, the search starts at a given record number backward towards to the beginning of the file.

- CRRN7: If mode is ENHANCED_SEARCH and SEARCH_INDICATION_START_BACKWARD_FROM_PREVIOUS is set in searchIndication, the search is backward starting from previous record towards to the beginning of the file.
- CRRN8: If the mode is ENHANCED_SEARCH and SEARCH_INDICATION_START_BACKWARD_FROM_PREVIOUS_GR is set in searchIndication, the search is backward starting at a given record from previous record towards to the beginning of the file.
- CRRN9: If the mode is ENHANCED_SEARCH and SEARCH_INDICATION_START_FORWARD_FROM_NEXT is set in searchIndication, the search is forward starting at the next record towards the end of the file.
- CRRN10: If the mode is ENHANCED_SEARCH and SEARCH_INDICATION_FORWARD_FROM_NEXT_GR is set in searchIndication, the search is forward starting at a given record number towards to the end of the file.
- CRRN11: If the mode is ENHANCED_SEARCH, and bit 4 of the most significant byte of the searchIndication is not set, the search starts in the record from the offset (absolute position) given in the less significant byte of searchIndication.
- CRRN12: If the mode is ENHANCED_SEARCH, and bit 4 of the most significant byte of the searchIndication is set, the search starts in the record after the first occurrence of the value contained in the less significant byte of searchIndication.
- CRRN13: If pattern given in patt[] is not found, the method returns 0.
- CRRN14: If one or more matches are found the record pointer shall be set to the first record where the search pattern was found.

5.1.1.6.1.2 Parameter errors

- CRRP1: If mode is not 4, 5 or 6, an instance of UICCException shall be thrown. The reason code shall be UICCException.INVALID_MODE.
- CRRP2: If the pattern array patt is null, an instance of java.lang.NullPointerException shall be thrown.
- CRRP3: If the response array response is null, an instance of java.lang.NullPointerExceptino shall be thrown.
- CRRP4: If parameter pattOffset is negative, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
- CRRP5: If parameter pattLength is negative, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
- CRRP6: If parameter respOffset is negative, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
- CRRP7: If parameter respLength negative, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
- CRRP8: If parameter pattOffset plus pattLength are greater than the length of array patt, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
- CRRP9: If parameter respOffset plus respLength are greater than the length of array response a ArrayIndexOutOfBoundsException shall be thrown.
- CRRP10: If parameter recordNum is negative, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD_NOT_FOUND.
- CRRP11: If parameter recordNum is greather than, the total number of records from the currently selected EF, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD_NOT_FOUND.

- CRRP12 If pattLength is greater than record size of the currently selected EF an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT_OF_FILE_BOUNDARIES.

5.1.1.6.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the currently selected EF is not linear fixed or cyclic, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND_INCOMPATIBLE.
- CRRC3: If the calling applet does not fulfil the access condition, READ, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.
- CRRC4: If the currently selected EF is deactivated and the file status of the EF does not allow for reading a deactivated file, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF_DATA_INVALIDATED.
- CRRC5: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.1.1.6.2 Test area files

Test Source: Test_Api_1_Fvw_Sear.java.

Test Applet: Api_1_Fvw_Sear_1.java.

Cap File: api_1_fvw_sear.cap.

5.1.1.6.3 Test coverage

CRR number	Test case number
N1	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37
N2	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37
N3	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37
N4	12
N5	2,28
N6	3, 29
N7	6, 7, 34, 35
N8	8, 9, 36, 37
N9	10, 11, 30, 31
N10	12, 13, 32, 33
N11	6, 8, 10, 12, 30, 32, 34, 36
N12	7, 9, 11, 13, 31, 33, 35, 37
N13	2, 3, 5, 7, 9, 11, 28, 31
N14	6, 7, 10, 11, 30, 31, 34, 35
P1	13
P2	14
P3	15
P4	16
P5	17
P6	18
P7	19
P8	20
P9	21
P10	22
P11	23
P12	24
C1	1
C2	25
C3	26
C4	27
C5	Not testable

5.1.1.6.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**No EF selected** 1- select DF_{TEST}, fid=1111 2- searchRecord()	2-shall throw uicc.access.UICC Exception with reason code NO_EF_SELECTED.	
2	**Fixed linear EF, Simple mode search forward** 1- select EF_{LSEA}, fid=6F1A 2- mode = SIMPLE_SEARCH_START_FORWARD recordNum = 1 patt[]={0x10,0x03,0x04} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 3- Simple mode search forward mode = SIMPLE_SEARCH_START_FORWARD recordNum = 2 patt[]={0x10,0x03,0x04} pattOffset = 1 pattLength = 2 resp.length = 4 respOffset = 1 respLength = 3 searchRecord() 4- Simple mode search forward mode = SIMPLE_SEARCH_START_FORWARD recordNum = 0 patt[]={0x10,0x03,0x04} pattOffset = 1 pattLength = 2 resp.length = 4 respOffset = 1 respLength = 3 searchRecord()	2- no exception shall be thrown Shall return 0. response shall be: response={0,0,0,0} 3- Shall return 2. response shall be: response={0,2,4,0} 4- Shall return 2. response shall be: response={0,2,4,0}	
3	**Simple mode, search backward** 1- mode = SIMPLE_SEARCH_START_BACKWARD recordNum = 1 patt[]={0x08,0x0A,0x0B} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 2 respLength = 2 searchRecord() 2-mode = SIMPLE_SEARCH_START_BACKWARD recordNum = 6 patt[]={0x08,0x09,0x0A,0x0B } pattOffset = 1 pattLength = 2 response[] = {0,0,0,0} respOffset = 1 respLength = 3 searchRecord() 3-mode = SIMPLE_SEARCH_START_BACKWARD recordNum = 0 patt[]={0x08,0x09,0x0A,0x0B } pattOffset = 1 pattLength = 2 response[] = {0,0,0,0} respOffset = 1 respLength = 3 searchRecord()	1- shall return 0. response shall be: response={0,0,0,0} 2- shall return 3. response shall be: response={0,4,3,1} 3- shall return 3. response shall be: response={0,4,3,1}	

Id	Description	API Expectation	APDU Expectation
4	**Enhanced Mode, search backward from previous record, start from an offset in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS + 0x0009 recordNum = 0 patt[]={0x01,0x02,0x03,0x04} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS +0x0000 recordNum = 0 patt[]={0x0C,0x0D,0x0E,0x0F,0x01,0x02} pattOffset = 0 pattLength = 5 response[] = {0,0,0,0} respOffset = 2 respLength = 2 searchRecord()	1- shall return 1, response shall be: resp={3,0,0,0} 2- shall return 1 response shall be: response={0,0,2,0}	
5	**Enhanced Mode, search backward from previous record, start from a value in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS + 0x0810 recordNum = 0 patt[]={0x01,0x02,0x03,0x04} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 2- perform 3 readRecord() in next mode to set current pointer to pointer 5 3- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS +0x080E recordNum = 0 patt[]={0x01,0x02,0x03,0x04} pattOffset = 3 pattLength = 1 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 0, response shall be: resp={0,0,0,0} 3- shall return 2 response shall be: response={4,2,0,0}	

Id	Description	API Expectation	APDU Expectation
6	**Enhanced Mode, search backward from previous given record, start from an offset in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS _GR + 0x0000 recordNum = 1 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS _GR + 0x0004 recordNum = 6 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 1, response shall be: resp={1,0,0,0} 2- shall return 4 response shall be: response={5,4,3,2}	
7	**Enhanced Mode, search backward from previous given record, start from a value in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS _GR + 0x080D recordNum = 1 patt[]={0x0E,0x0E,0x0E} pattOffset = 1 pattLength = 1 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS _GR + 0x0800 recordNum = 6 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 1, response shall be: resp={1,0,0,0} 2- shall return 0 response shall be: response={0,0,0,0}	

Id	Description	API Expectation	APDU Expectation
8	**Enhanced Mode, search forward from next record, start from an offset in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT + 0x0003 recordNum = 0 patt[]={0x00,0x0A,0x0B} pattOffset = 1 pattLength = 2 response[] = {0,0,0} respOffset = 2 respLength = 2 searchRecord() 2- Perform readRecord() in previous mode 3- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT + 0x0003 recordNum = 0 patt[]={0x00,0x0A,0x0B} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 2 response shall be: resp={0,0,3,4} 3- shall return 1 response shall be: response={4,0,0,0}	
9	**Enhanced Mode, search forward from next record, start from a value in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT + 0x0804 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0} respOffset = 2 respLength = 2 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT + 0x0801 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 2 pattLength = 1 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 0, response shall be: resp={0,0,0,0} 2- shall return 2 response shall be: response={5,6,0,0}	

Id	Description	API Expectation	APDU Expectation
10	**Enhanced Mode, search forward from next given record, start from an offset in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT_GR + 0x0007 recordNum = 1 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 1 respLength = 3 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT_GR + 0x000C recordNum = 3 patt[]={0x03,0x02,0x01} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 3, response shall be: resp={0,3,4,5} 2- shall return 1 response shall be: response={6,0,0,0}	
11	**Enhanced Mode, search forward from next given record, start from a value in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT_GR + 0x080D recordNum = 5 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT_GR + 0x080C recordNum = 5 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 0, response shall be: resp={0,0,0,0} 2- shall return 1 response shall be: response={5,0,0,0}	
12	**Simple mode, total number of found patterns exceed response[]** 1- mode = SIMPLE_SEARCH_START_FORWARD recordNum = 1 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 2- mode = SIMPLE_SEARCH_START_FORWARD recordNum = 1 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 4 response shall be: response={1,2,3,4} 2- shall return 4 response shall be: response={1,2,3,4,0}	

Id	Description	API Expectation	APDU Expectation
13	**Invalid mode** mode = 0x14 (simple search forward with SFI) searchIndication= 0 recordNum = 2 patt[]={0x01,0x02,0x03} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 2 respLength = 2 searchRecord()	shall throw an uicc.access.UICC Exception with reason code INVALID_MODE.	
14	**Pattern array is null** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]= null pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 2 respLength = 2 searchRecord()	shall throw an java.lang.NullPointerException.	
15	**Response array is null** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = null respOffset = 0 respLength = 5 searchRecord()	shall throw an instance of java.lang.NullPointerException.	
16	**pattOffset<0** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = -1 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	shall throw an instance of java.lang.ArrayIndexOutOfBoundsException.	
17	**pattLength<0** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = -1 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	shall throw an instance of java.lang.ArrayIndexOutOfBoundsException.	
18	**respOffset <0** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = -1 respLength = 5 searchRecord()	shall throw an instance of java.lang.ArrayIndexOutOfBoundsException.	

Id	Description	API Expectation	APDU Expectation
19	**respLength <0** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 0 respLength = -1 searchRecord()	shall throw an instance of java.lang.ArrayIndexOutOfBoundsException.	
20	**PattOffset + pattLength > patt[]** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 2 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 1 respLength = 5 searchRecord()	shall throw an instance of java.lang.ArrayIndexOutOfBoundsException.	
21	**RespOffset + respLength > response[]** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 3 respLength = 3 searchRecord()	shall throw an instance of java.lang.ArrayIndexOutOfBoundsException.	
22	**recordNum < 0** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = -1 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	shall throw an uicc.access.UICC Exception with reason code RECORD_NOT_FOUND	
23	**RecordNum > total number of file records** 1- mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 7 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	1- shall throw an uicc.access.UICC Exception with reason code RECORD_NOT_FOUND	

Id	Description	API Expectation	APDU Expectation
24	**pattlength > record length** 1- mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 3 patt[16]={0x55,0x55,…,0x55} pattOffset = 0 pattLength = 16 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT_GR + 0x000E recordNum = 3 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	1- shall throw an uicc.access.UICCException with reason code OUT_OF_FILE_BOUNDARIES. 2- shall throw an uicc.access.UICCException with reason code OUT_OF_FILE_BOUNDARIES.	
25	**Wrong file structure** 1- select EF_{TDAC}, fid=6F0F 2- searchRecord()	2- shall throw an uicc.access.UICCException with reason code COMMAND_INCOMPATIBLE	
26	**Security status not satisfied** 1- select EF_{LNR}, fid=6F0A 2- searchRecord()	2- shall throw an uicc.access.UICCException with reason code SECURITY_STATUS_NOT_SATISFIED	
27	**File deactivated** 1- select EF_{LARU}, fid=6F10 2- deactivateFile EF_{LARU} 3- searchRecord() 4- activateFile()	3- shall throw an uicc.access.UICCException with reason code DATA_INVALIDATED	

Id	Description	API Expectation	APDU Expectation
28	**Cyclic EF, Simple mode search forward** 1- select EF_{CSEA}, fid=6F1B 2- mode = SIMPLE_SEARCH_START_FORWARD recordNum = 1 patt[]={0x10,0x03,0x04} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord() 3- mode = SIMPLE_SEARCH_START_FORWARD recordNum = 2 patt[]={0x10,0x03,0x04} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 2 respLength = 3 searchRecord() 4- updateRecord() in previous mode with value {0x03,0x02,0x01,0x03,0x02,0x01,0x03,0x02,0x01,0x03,0x02,0x01,0x03,0x02,0x01} (new record 1 is set to previous record 6) 5- mode = SIMPLE_SEARCH_START_FORWARD recordNum = 2 patt[]={0x10,0x03,0x04} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 2 respLength = 3 searchRecord()	2- shall return 0 response shall be: response={0,0,0,0,0} 3- Shall either return 3 and response shall be: response={0,0,2,4,1} or shall return 2 and response shall be: response={0,0,2,4,0} 5- Shall return 3. response shall be: response={0,0,2,3,5}	
29	**Cyclic EF, Simple mode search backward** mode = SIMPLE_SEARCH_START_BACKWARD recordNum = 3 patt[]={0x10,0x03,0x04} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 1 respLength = 4 searchRecord()	Shall either return 3 and response shall be: response={0,3,2,5,0} or shall return 2 and response shall be: response={0,3,2,0,0}	
30	**Cyclic EF, Enhanced mode, search forward from next record, start from an offset in record** mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_FORWARD_FROM_NEXT + 0x0009 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0,0} respOffset = 2 respLength = 3 searchRecord()	shall return 3 response shall be: response={0,0,4,5,6}	

Id	Description	API Expectation	APDU Expectation
31	**Cyclic EF, Enhanced mode, search forward from next record, start from a value in record** mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_FORWARD_FROM_NEXT + 0x0810 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0,0} respOffset = 2 respLength = 3 searchRecord()	shall return 0 response shall be: response={0,0,0,0,0}	
32	**Cyclic EF, Enhanced mode, search forward from next given record, start from an offset in record** mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_FORWARD_FROM_NEXT_GR + 0x0005 recordNum = 3 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	shall either return 5 and response shall be: response={3,4,5,6,1} or shall return 4 and response shall be: response={3,4,5,6,0}	
33	**Cyclic EF, Enhanced mode, search forward from next given record, start from a value in record** 1- mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_FORWARD_FROM_NEXT_GR + 0x0805 recordNum = 6 patt[]={0x0E,0x0F,0x00} pattOffset = 0 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord() 2- Restore EF initial state (record 1 shall be assigned to the record that content is {0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0A, 0x0B,0x0C,0x0D,0xE,0x0F}) using 5 updateRecord() in previous mode	1- shall either return 2 and response shall be: response={2,4,0,0,0} or shall return 0 and response shall be: response={0,0,0,0,0}	
34	**Cyclic EF, Enhanced mode, search backward from previous record, start from an offset in record** 1- Set current record pointer to record 6 using 5 readRecord() in next mode 2- mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_BACKWARD_FROM_PREVIOUS + 0x0003 recordNum = 0 patt[]={0x02,0x01,0x00} pattOffset = 0 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 3 respLength = 2 searchRecord()	2- shall either return 1 and response shall be: response={0,0,0,6,0} or shall return 0 and response shall be: response={0,0,0,0,0}	

Id	Description	API Expectation	APDU Expectation
35	**Cyclic EF, Enhanced mode, search backward from previous record, start from a value in record** mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_BACKWARD_FROM_PREVIOUS + 0x0801 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	shall return 5 response shall be: response={5,4,3,2,1}	
36	**Cyclic EF, Enhanced mode, search backward from given record, start from an offset in record** mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_BACKWARD_FROM_PREVIOUS_GR + 0x0003 recordNum = 5 patt[]={0x02,0x01,0x00} pattOffset = 0 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 3 respLength = 2 searchRecord()	shall either return 1 and response shall be: response={0,0,0,6,0} or shall return 0 and response shall be: response={0,0,0,0,0}	
37	**Cyclic EF, Enhanced mode, search backward from given record, start from a value in record** mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_BACKWARD_FROM_PREVIOUS_GR + 0x0801 recordNum = 3 patt[]={0x01,0x02,0x03} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	Shall either return 5 and response shall be: response={3,2,1,5,4} or shall return 3 and response shall be: response={3,2,1,0,0}	

5.1.1.7 Method select (byte sfi)

Test Area Reference: Api_1_Fvw_Slctb.

5.1.1.7.1 Conformance requirement

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

```
public void select(byte sfi)
            throws UICCException
```

5.1.1.7.1.1 Normal execution

- CRRN1: Selects a file by its Short File Identifier in the current directory of the FileView.
- CRRN2: Allows to update the current file without handling the Select Response.
- CRRN3: The current EF it self can be selected.
- CRRN4: The file context associated with the FileView object is changed after successful execution.

5.1.1.7.1.2 Parameter errors

- CRRP1: If the file which sfi matches is not in the current directory or no file matches the sfi, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE_NOT_FOUND.

5.1.1.7.1.3 Context errors

- CRRC1: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.

- CRRC2: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.1.1.7.2 Test area files

Test Source: Test_Api_1_Fvw_ Slctb.java.

Test Applet: Api_1_Fvw_ Slctb _1.java.

Cap File: api_1_fvw_ slctb.cap.

5.1.1.7.3 Test coverage

CRR number	Test case number
N1	1, 2, 4
N2	Not testable
N3	3
N4	5
P1	4
C1	Not testable
C2	Not testable

5.1.1.7.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Selection possibilities, UICC file system** 1- get a FileView UICCSystem.getTheUICCView(CLEAR_ON_RESET) 2- select DF_{TEST}, fid=1111 3- select with sfi EF_{TNR}, sfi=0x01 4- select with sfi EF_{TNU}, sfi=0x02 5- select with sfi EF_{CNU}, sfi=0x05	2- no exception shall be thrown 3- no exception shall be thrown 4- no exception shall be thrown 5- no exception shall be thrown	
2	**Selection possibilities, ADF1** 1- get a FileView UICCSystem.getTheFileView(AID_ADF1,CLEAR_ON_RESET) 2- select DF_{TEST}, fid=1111 3- select with sfi EF_{TNR}, sfi=0x01 4- select with sfi EF_{TNU}, sfi=0x02 5- select with sfi EF_{CNU}, sfi=0x05	2- no exception shall be thrown 3- no exception shall be thrown 4- no exception shall be thrown 5- no exception shall be thrown	
3	**Current EF itself can be selected** 1- get a FileView UICCSystem.getTheUICCView(CLEAR_ON_RESET) 2- select DF_{TEST}, fid=1111 3- select with sfi EF_{TNR}, sfi=0x01 4- select with sfi EF_{TNR}, sfi=0x01	4- no exception shall be thrown	
4	**FILE_NOT_FOUND** 1- try to select a file with sfi=0x55	1- shall throw an uicc.access.UICCException with reason code FILE_NOT_FOUND	

Id	Description	API Expectation	APDU Expectation
5	**File context changed** 1- select EF_{TARU}, sfi=0x03 read 3 first bytes 2- select EF_{TNU}, sfi=0x02 read file content	1- file content should be {0xFF,0xFF,0xFF} 2- file content should be {0x55,0x55,0x55}	

5.1.1.8 Method select (short fid)

Test Area Reference: Api_1_Fvw_Slcts.

5.1.1.8.1 Conformance requirement

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

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

5.1.1.8.1.1 Normal execution

- CRRN1: Selects a file of the UICC file system or of an ADF file system by file identifier.
- CRRN2: Allows to update the current file without handling the Select Response.
- CRRN3: After selecting an ADF/MF/DF no EF is selected.
- CRRN4: After selecting a linear fixed EF no record is selected.
- CRRN5: Any file can be selected by FID which is an immediate child of the current directory.
- CRRN6: Any DF can be selected by FID which is an immediate child of the parent of the current DF.
- CRRN7: The parent of the current directory can be selected by the FID.
- CRRN8: The ADF of the current active application can be selected by the FID.
- CRRN9: The ADF/MF/EF can always be self selected.
- CRRN10: The file context associated with the FileView object is changed after successful execution.

5.1.1.8.1.2 Parameter errors

No requirements.

5.1.1.8.1.3 Context errors

- CRRC1: If the file with a File Identifier which matches fid could not be found according to the selection rule listed in CCRN3, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE_NOT_FOUND.
- CRRC2: If the file with a File Identifier which matches fid could not be found according to the selection rule listed in CCRN4, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE_NOT_FOUND.
- CRRC3: If the file with a File Identifier which matches fid could not be found according to the selection rule listed in CCRN5, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE_NOT_FOUND.
- CRRC4: If the file with a File Identifier which matches fid could not be found according to the selection rule listed in CCRN6, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE_NOT_FOUND.

- CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.
- CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.1.1.8.2 Test area files

Test Source: Test_Api_1_Fvw_Slcts.java.

Test Applet: Api_1_Fvw_Slcts_1.java.

Cap File: api_1_fvw_slcts.cap.

5.1.1.8.3 Test coverage

CRR number	Test case number
N1	1, 2
N2	5
N3	5
N4	6
N5	1, 2
N6	1, 2
N7	1, 2
N8	1, 2
N9	4
N10	Tested in Api_1_Cont, test case 1 and 2
C1	3
C2	3
C3	3
C4	3
C5	Not testable
C6	Not testable

5.1.1.8.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	**Get a FileView object, UICC file system** 1- get a FileView UICCSystem.getTheUICCView(CLEAR_ON_RESET)		
1	**Selection possibilities** 1- select EF_{UICC}, fid=2FF0 2- select DF_{TEST}, fid=1111 3- select EF_{CNU}, fid=6F05 4- select EF_{TAAA}, fid=6F16 5- select DF_{SUB_TEST}, fid=2211 6- select DF_{TEST}, fid=1111 7- select EF_{TAAA}, fid=6F16 8- select DF_{TEST}, fid=1111 9- select MF, fid=3F00 10- select DF_{TEST}, fid=1111 11- select EF_{TAAA}, fid=6F16 12- select MF, fid=3F00	No exception shall be thrown	
2	**Selection possibilities, ADF1** 1- get a FileView UICCSystem.getTheFileView(AID_ADF1,CLEAR_ON_RESET) 2- select EF_{UICC}, fid=2FF0 3- select DF_{TEST}, fid=1111 4- select EF_{CNU}, fid=6F05 5- select EF_{TAAA}, fid=6F16 6- select DF_{SUB_TEST}, fid=2211 7- select DF_{TEST}, fid=1111 8- select EF_{TAAA}, fid=6F16 9- select DF_{TEST}, fid=1111	No exception shall be thrown	

Id	Description	API Expectation	APDU Expectation
3	**No selection of unreachable file** 1- get a FileView UICCSystem.getTheUICCView(CLEAR_ON_RESET) 2- select EF_{CNU}, fid=6F05 3- select DF_{TEST}, fid=1111 4- select EF_{TAA}, fid=2222 5- select EF_{CNU}, fid=6F05 6- select DF_{SUB_TEST}, fid=2211 7- select EF_{TAA}, fid=2222 8- select $DF_{TELECOM}$, fid=7F10	2- A UICCException.FILE_NOT_FOUND shall be thrown. 3- No exception shall be thrown 4- A UICCException.FILE_NOT_FOUND shall be thrown. 5- No exception shall be thrown 6- No exception shall be thrown 7- No exception shall be thrown 8- A UICCException.FILE_NOT_FOUND shall be thrown.	
4	**Self selection** 1- select MF, fid=3F00 2- select MF, fid=3F00 3- select DF_{TEST}, fid=1111 4- select DF_{TEST}, fid=1111 5- select EF_{TAAA}, fid=6F16 6- select EF_{TAAA}, fid=6F16 7- get a FileView UICCSystem.getTheFileView(AID_ADF1,CLEAR_ON_RESET) 8- select ADF, fid=7FFF 9- select ADF, fid=7FFF	2- No exception shall be thrown 4- No exception shall be thrown 6- No exception shall be thrown 8- No exception shall be thrown 9- No exception shall be thrown	
5	**EF not selected after MF/DF selection** 1- select MF, fid=3F00 2- updateRecord() 3- select DF_{TEST}, fid=1111 4- updateRecord()	2- A UICCException.NO_EF_SELECTED shall be thrown 4- A UICCException.NO_EF_SELECTED shall be thrown	
6	**No record is selected after selecting linear fixed EF** 1- select MF, fid = 3F00 2- select DF_{TEST}, 3- select EF_{LARU}, 4 - recNumber = 0 mode = REC_ACC_MODE_ CURRENT readRecord() 5- select EF_{CARU}, 6 - recNumber = 0 mode = REC_ACC_MODE_ CURRENT readRecord()	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - No exception shall be thrown. 4 - Shall throw uicc.access.UICCException with reason code RECORD_NOT_FOUND. 5 - No exception shall be thrown. 6 - Shall throw uicc.access.UICCException with reason code RECORD_NOT_FOUND.	

5.1.1.9 Method select(short fid, byte[] fcp, short fcpOffset, short fcpLength)

Test Area Reference: Api_1_Fvw_Slctb_bss.

5.1.1.9.1 Conformance requirement

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

```
public short select(short fid,
                    byte[] fcp,
                    short fcpOffset,
                    short fcpLength)
             throws java.lang.NullPointerException,
                    java.lang.ArrayIndexOutOfBoundsException,
```

```
UICCException
```

5.1.1.9.1.1 Normal execution

- CRRN1: Selects a file of the UICC file system or of an ADF file system as defined in ETSI TS 102 221 [5].
- CRRN2: The method returns the FCP information in a form of a TLV structure as specified in ETSI TS 102 221 [5].
- CRRN3: If the fcpLength is greater than the length of the response, the whole response is copied into the fcp buffer and the length of the response is returned by the method.
- CRRN4: If the fcpLength is smaller than the length of the response, the first part of the response is copied into the fcp buffer and the fcpLength is returned by the method.
- CRRN5: After selecting an ADF/MF/DF no EF is selected.
- CRRN6: After selecting a linear fixed EF no record is selected.
- CRRN7: After selecting a cyclic EF the last updated record is the first record.
- CRRN8: Any file can be selected by FID which is an immediate child of the current directory.
- CRRN9: Any DF can be selected by FID which is an immediate child of the parent of the current DF.
- CRRN10: The parent of the current directory can be selected by the FID.
- CRRN11: The ADF of the current active application can be selected by the FID.
- CRRN12: The ADF/MF can always be selected.
- CRRN13: The file context associated with the FileView object is changed after successful execution.
- CRRN14: The current file context of any other applets shall not be changed. This will be tested during the testing of the framework.

5.1.1.9.1.2 Parameter errors

- CRRP1: If the array fcp is null, an instance of NullPointerException shall be thrown.
- CRRP2: If fcpOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP3: If fcpLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP4: If fcpOffset plus fcpLength is greater than the length of the array fcp.length an instance of ArrayIndexOutOfBoundsException shall be thrown.

5.1.1.9.1.3 Context errors

- CRRC1: If the file with a File-ID which matches fid could not be found according to the selection rules, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE_NOT_FOUND.
- CRRC2: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.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 UICCException.INTERNAL_ERROR.

5.1.1.9.2 Test area files

Test Source: Test_Api_1_Fvw_ Slctb_bss.java.

Test Applet: Api_1_Fvw_ Slctb_bss _1.java.

Cap File: api_1_fvw_ slctb_bss.cap.

5.1.1.9.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5, 6, 7, 8, 14, 18, 20, 21
N2	1, 2, 3, 4, 5, 6, 7, 8, 14, 18, 20, 21
N3	1
N4	2, 3, 4 ,5, 6, 7, 8
N5	15, 19
N6	17
N7	18
N8	14
N9	14
N10	14
N11	19, 20
N12	20
N13	20
N14	1, 2, 3, 4, 5, 6, 7, 8, 14, 18, 20
P1	9
P2	10
P3	11
P4	12, 13
C1	16
C2	Not testable
C3	Not testable

5.1.1.9.4 Test procedure

Id	Description	API Expectation	APDU Expectation

0	**Get a FileView object, UICC file system**
	
	`get a FileView`
	`FileView.getTheUICCView(CLEAR_ON_RESET)`		

1	**Select EF_{TARU} in MF (Transparent EF)**
	
	`Select` DF_{TEST}
	`select` EF_{TARU}`,`
	`fid=6F03`
	`byte[] fcp = new byte[132]`
	`fcpOffset = 0`
	`fcpLength = 127`
		Shall return at least 19.
		fcp[] shall contain following TLVs:
		1. 82 02 41 21 //file descriptor
		2. 83 02 2F E2 //file id
		3. 8A 01 05 //life cycle status
		4. 80 02 00 0A // file size	

2	**Select EF_{TARU} in MF (Transparent EF)**
	
	`select` EF_{TARU}`,`
	`fid=6F03`
	`fcpOffset = 0`
	`fcpLength = 7`
	`select()`
		Shall return 7.
		fcp[] shall contain the first 7 bytes of the FCP structure and contain following TLV.
		1. 82 02 41 21 //file descriptor	

3	**Select DF_{TEST} in MF**
	
	`fid =` DF_{TEST}`, fid=1111`
	`fcpOffset = 0`
	`fcpLength = 127`
	`select()`
		Shall return at least 17.
		fcp[] shall contain following TLVs
		1. 82 02 78 21 //file descriptor
		2. 83 02 11 11 //file id
		3. 8A 01 05 //life cycle status	

Id	Description	API Expectation	APDU Expectation
4	**Select EF_{CARU} in DF_{TEST} (Cyclic EF)** select EF_{CARU}, fid=6f09 fcpOffset = 0 fcpLength = 11 select()	Shall return: 11 fcp[] shall contain following TLV: 82 05 46 21 00 03 02	
5	**Select ADF1** select ADF fid=7FFF fcp[0:5]=0x00 fcpOffset=5 fcpLength=127 select	Shall return: at least 27 The first 5 bytes of fcp[] shall be 0x00 and contains following TLVs: 1. 82 02 78 21 //file descriptor 2. 84 10 A0 00 00 00 09 00 05 FF FF FF FF 89 60 00 00 00 //DF Name 3. 8A 01 05 //life cycle	
6	**Select MF** select MF, fid= 3F00 fcpOffset = 0 fcpLength = 11 select()	Shall return: 11 fcp[] shall contain following TLVs: 1. 82 02 38/78 21 //file descriptor 2. 83 02 3F 00 //file ID	
7	**Select $DF_{TELECOM}$ in MF** select $DF_{TELECOM}$, fid=7F10 fcp[0] = fcp[1] = 0x05 fcpOffset = 2 fcpLength = 13 select()	Shall return 13. The first 2 bytes of fcp[] shall be 0x05 and fcp[] shall contain following TLVs 1. 82 02 38/78 21 //file descriptor 2. 83 02 7F 10 //file id	
8	**Select EF_{LARU} in $DF_{TELECOM}$ (Linear FixedEF)** select EF_{LARU}, fid = 6F0C fcpOffset = 0 fcpLength = 14	Shall return 14. fcp[] shall contain following TLVs: 1. 82 05 42 21 00 04 02 2. 83 02 6F 0C	
9	**fcp is null** select EF_{LARU}, fid = 6F0C byte[] nullBuffer = null fcpOffset = 0 fcpLength = 15	Shall throw java.lang.NullPointerException	
10	**fcpOffset < 0** select EF_{LARU}, fid = 6F0C fcpOffset = -1 fcpLength = 15	Shall throw java.lang.ArrayIndexOutOfBoundsException	
11	**fcpLength < 0** select EF_{LARU}, fid = 6F0C fcpOffset = 0 fcpLength = -1	Shall throw java.lang.ArrayIndexOutOfBoundsException	
12	**fcpOffset + fcpLength > fcp.length** select EF_{LARU}, fid = 6F0C fcpOffset = 115 fcpLength = 18	Shall throw java.lang.ArrayIndexOutOfBoundsException	
13	**fcpOffset + fcpLength > fcp.length** select EF_{LARU}, fid = 6F0C fcpOffset = fcpLength + 1 fcpLength = 0	Shall throw java.lang.ArrayIndexOutOfBoundsException	

Id	Description	API Expectation	APDU Expectation
14	**Selection possibilities** 0- select MF, fid=3F00 1- select EF_{UICC}, fid=2FF0 2- select DF_{TEST}, fid=1111 3- select EF_{CNU}, fid=6F05 4- select EF_{TAAA}, fid=6F16 5- select DF_{SUB_TEST}, fid=2211 6- select DF_{TEST}, fid=1111 7- select EF_{TAAA}, fid=6F16 8- select DF_{TEST}, fid=1111 9- select MF, fid=3F00 10- select DF_{TEST}, fid=1111 11- select EF_{TAAA}, fid=6F16 12- select MF, fid=3F00	No exception shall be thrown.	
15	**EF not selected after MF/DF selection** 1- select MF, fid = 3F00 select EF_{ICCID}, fid = 2FE2 2 - select MF fid = 3F00 select() readBinary()	2 - Shall throw uicc.access.UICCException with reason code NO_EF_SELECTED.	
16	**No selection of non-reachable file** 1 - select MF, fid = 3F00 2 - select EF_{CARU}, fid= 0x6F09	2 - Shall throw uicc.access.UICCException with reason code FILE_NOT_FOUND.	
17	**No record is selected after selecting linear fixed EF** 1- select MF, fid = 3F00 2- select DF_{TEST}, fid=1111 3- select EF_{LARU}, fid=6F0C 4 - recNumber = 0 mode = REC_ACC_MODE_ CURRENT readRecord()	4 - Shall throw uicc.access.UICC Exception with reason code RECORD_NOT_FOUND.	
18	**Record pointer in selected cyclic EF** 1- select MF, fid = 3F00 2- select DF_{TEST}, fid=1111 3- select EF_{CARU}, fid=6F09 4- byte[] data1 = { 1,2,3 } mode = REC_ACC_MODE_PREVIOUS updateRecord(data1) 5- select EF_{CARU} fid = 6F09 select() mode = REC_ACC_MODE_PREVIOUS readRecord() readRecord(data2) compare data1 to data2 6- restore original data of EF_{CARU}	5 - The contents of data1 and data2 shall be identical.	
19	**EF not selected after ADF/DF selection** 1- get a FileView UICCSystem.getTheFileView(AID_ADF1,CLEAR_ON_RESET) 2- select ADF, fid = 7FFF select EF_{UICC}, fid = 2FF0 3 - select ADF fid = 7FFF select() readBinary()	3 - Shall throw uicc.access.UICCException with reason code NO_EF_SELECTED.	

Id	Description	API Expectation	APDU Expectation
20	**Reselection** 1- Using the ADF FileView select ADF, fid=7FFF select ADF, fid=7FFF 2- Using the UICC FileView select MF, fid=3F00 select MF, fid=3F00 3- select DF_{TEST}, fid=1111 select DF_{TEST}, fid=1111 5- select EF_{TAAA}, fid=6F16 select EF_{TAAA}, fid=6F16	No exceptions shall be thrown	
21	**Security attributes** 1- Using the ADF FileView select ADF, fid=7FFF select DF_{TEST}, fid=1111 select EF_{LARR1}, fid=6FA1 2- Using the UICC FileView select MF, fid=3F00 select DF_{TEST}, fid=1111 select EF_{TARR3}, fid=6FB3	1- fcp[] shall contain the following TLV 8B 03 AC 00 01 or 8B 06 AC 00 00 01 01 01 2- fcp[] shall contain the following TLV 8B 03 AC 00 03 or 8B 06 AC 00 00 03 01 03	

5.1.1.10 Method status

Test Area Reference: Api_1_Fvw_Stat.

5.1.1.10.1 Conformance requirement

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

```
public short status(byte[] fcp,
                    short fcpOffset,
                    short fcpLength)
             throws java.lang.NullPointerException,
                    java.lang.ArrayIndexOutOfBoundsException,
                    UICCException
```

5.1.1.10.1.1 Normal execution

- CRRN1: The method returns the File Control Parameter of the current selected DF/MF or ADF as defined in ETSI TS 102 221 [5].
- CRRN2: If the `fcpLength` is greater than the length of the response, the whole response is copied into the `fcp` buffer and the length of the response is returned by the method.
- CRRN3: f the `fcplength` is smaller than the length of the response, the first part of the response is copied into the `fcp` buffer and the `fcpLength` is returned by the method.

5.1.1.10.1.2 Parameter errors

- CRRP1: If the array fcp is null, an instance of NullPointerException shall be thrown.
- CRRP2: If fcpOffset is negative, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP3: If fcpLength is negative, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP4: If fcpOffset+fcpLength is greater than fcp.length an ArrayIndexOutOfBoundsException shall be thrown.

5.1.1.10.1.3 Context errors

- CRRC1: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.
- CRRC2: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.1.1.10.2 Test area files

Test Source: Test_ Api_1_Fvw_Stat.java.

Test Applet: Api_1_Fvw_Stat_1.java.

Cap File: Api_1_Fvw_Stat.cap.

5.1.1.10.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5, 11
N2	2, 3
N3	1, 4
P1	6
P2	7
P3	8
P4	9, 10
C1	Not testable
C2	Not testable

5.1.1.10.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Status of MF** 1- Get a FileView object, UICC file get a FileView FileView.getTheUICCView(CLEAR_ON_RESET) 2- select MF byte[] fcp = new byte[127] fcp[0:2]= 0xCC fcpOffset = 3 fcpLength = 11 status()	2- Shall return 11. The first 3 bytes of fcp[] shall contain 0xCC. fcp[] shall contain following TLVs: 1. 82 02 38/78 21 //file descriptor 2. 83 02 3F 00 //file ID	
2	**Status after select** EF_{TARU} **in MF** 1 - select DF_{TEST} select EF_{TARU}, fid = 6F03 fcpOffset = 0 fcpLength = 127 select() status()	1- Shall return at least 19. fcp[] shall contain following TLVs: 1. 82 02 38/78 21 //file descriptor 2. 83 02 11 11 //file ID	
3	**Status of** $DF_{TELECOM}$ 1 - fid = 7F10 fcpOffset = 0 fcpLength = 127 status()	1 - Shall return at least 17 and the entire structure of the file control parameters. The file identifier shall be contain the fid of $DF_{TELECOM}$.	
4	**Status** $DF_{TELECOM}$ Select $DF_{TELECOM}$, fid=7F10 fcpOffset = 0 fcpLength = 11 status()	Shall return 11. fcp shall contain the first 11 bytes of the FCP structure starting at index 0. fcp[] shall contain following TLVs: 1. 82 02 38/78 21 //file descriptor 2. 83 02 7F 10 //file id	

Id	Description	API Expectation	APDU Expectation
5	**Status ADF1** select ADF, fid=7FFF fcpOffset = 0 fcpLength = 127	Shall return at least 27 fcp[] shall contain the entire FCP structure fcp[] shall contain following TLVs: 1. 82 02 78 21 //file descriptor 2. 84 10 A0 00 00 00 09 00 05 FF FF FF FF 89 60 00 00 00 //DF Name 3. 8A 01 05 //life cycle	
6	**fcp is null** byte[] nullBuffer = null fcpOffset = 0 fcpLength = 34 status()	Shall throw java.lang.NullPointerException.	
7	**fcpOffset < 0** fcpOffset = -1 fcpLength = 34 status()	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
8	**fcpLength < 0** fcpOffset = 0 fcpLength = -1 status()	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
9	**fcpOffset + fcpLength > fcp.length** fcpOffset = fcp.length-1 fcpLength = 15 status()	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
10	**fcpOffset + fcpLength > fcp.length** fcpOffset = fcp.length+1 fcpLength = 0 status()	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
11	**Security attributes** 1- Using the ADF FileView select ADF, fid=7FFF select DF_{TEST}, select DF_{ARR2}, 2- Using the UICC FileView select MF, fid=3F00 select DF_{TEST}, select DF_{ARR4},	1- fcp[] shall contain the following TLV 8B 03 AC 00 02 or 8B 06 AC 00 00 02 01 02 2- fcp[] shall contain the following TLV 8B 03 AC 00 04 or 8B 06 AC 00 00 04 01 04	

5.1.1.11 Method updateBinary

Test Area Reference: Api_1_Fvw_Updb.

5.1.1.11.1 Conformance requirement

The method with 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,
                         UICCException
```

5.1.1.11.1.1 Normal execution

- CRRN1: Updated the data bytes of the current selected transparent EF.

5.1.1.11.1.2 Parameter errors

- CRRP1: If recOffset is less than 0, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT_OF_FILE _BOUNDARIES.
- CRRP2: : If fileOffset plus dataLength exceeds the length of the file, an instance of UICCException shall be thrown. The reason code shall be UICCException.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.

5.1.1.11.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the currently selected EF is not transparent, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND_INCOMPATIBLE.
- CRRC3: If the calling applet does not fulfil the access condition, UPDATE, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException. SECURITY_STATUS_NOT_SATISFIED.
- CRRC4: If the currently selected EF is deactivated and the file status of the EF does not allow for updating of a deactivated file, an instance of UICCException shall be thrown. The reason code shall be UICCException. REF_DATA_INVALIDATED.
- CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.
- CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.1.1.11.2 Test area files

Test Source: Test_Api_1_Fvw_Updb.java.

Test Applet: Api_1_Fvw_Updb _1.java.

Cap File: api_1_fvw_updb.cap.

5.1.1.11.3 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,	Not Testable
C6	Not Testable

5.1.1.11.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	**Get the UICC FileView** FileView.getTheUICCView(CLEAR_ON_RESET)		0
1	**No EF selected** fileOffset = 0 byte[] data = new byte[20] data[0] = '55' dataOffset = 0 dataLength = 10 updateBinary()	Shall throw uicc.access.UICC Exception with reason code NO_EF_SELECTED.	1
2	**Update Transparent EF** 1- select DF_{TEST}, fid = 1111 2- select EF_{TARU}, fid = 6F03 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	**fileOffset = 254** 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 uicc.access.UICCException with reason code OUT_OF_FILE_BOUNDARIES.	
5	**fileOffset + dataLength > EF length** fileOffset = 259 dataOffset = 0 dataLength = 2 updateBinary()	Shall throw uicc.access.UICC Exception 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- select DF_{TEST}, fid = 1111 2- select DF_{LARU}, fid = 6F0C 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 uicc.access.UICC Exception with reason code COMMAND_INCOMPATIBLE.	

Id	Description	API Expectation	APDU Expectation
11	**Access condition not fulfilled** 1- select DF_{TEST}, fid = 1111 2- select EF_{TNU}, fid = 6F02 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 uicc.access.UICCException with reason code SECURITY_STATUS_NOT_SATISFIED.	
12	**EF is deactivated** 1- select EF_{TNR}, fid = 6F01 deactiveFile() 2- fileOffset = 0 data[0] = '55' dataOffset = 0 dataLength = 1 updateBinary() 3- activateFile()	1- No exception shall be thrown. 2- Shall throw uicc.access.UICCException with reason code REF_DATA_INVALIDATED 3- No exception shall be thrown.	

5.1.1.12 Method updateRecord

Test Area Reference: Api_1_Fvw_Updr.

5.1.1.12.1 Conformance requirement

The method with 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,
                         UICCException
```

5.1.1.12.1.1 Normal execution

- CRRN1: Reads a record or a part of record of the current linear fixed or cyclic EF into byte array data[].
- CRRN2: If the access mode is REC_ACC_MODE_ CURRENT the current selected record will be updated. The current record pointer shall not be changed.
- CRRN3:If the access mode is REC_ACC_MODE_ABSOLUTE and the file is linear fixed EF, the record addresss by recNumber will be updated. The current record pointer shall not be changed.
- CRRN4: 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. The record pointer shall be updated.
- CRRN5: If the access mode is REC_ACC_MODE_NEXT and the record pointer has not been previously set within the selected EF, the record pointer shall be set to the first record and this record should be updated.
- CRRN6: If the access mode is REC_ACC_MODE_PREVIOUS the previous record relative to the current selected record will be selected and updated. The record pointer shall be updated.
- CRRN7: If the access mode is REC_ACC_MODE_PREVIOUS and the record pointer has not been previously set within the selected EF, then the record pointer should be set to the last record in this EF. This record should be updated.
- CRRN8: If the access mode is REC_ACC_MODE_PREVIOUS, the file is a cyclic EF, the oldest record will be updated independent of the current record pointer and this record becomes record number 1 and the current record.

5.1.1.12.1.2 Parameter errors

- CRRP1: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_ABSOLUTE and recNumber is less than 0 or greater than records available, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD_NOT_FOUND.
- CRRP2: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE _CURRENT, recNumber is 0 and there is no current record selected, an instance of UICCException shall be thrown. The reason code shall be UICCException. RECORD_NOT_FOUND.
- 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 UICCException shall be thrown. The reason code shall be UICCException. RECORD_NOT_FOUND.
- 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 UICCException shall be thrown. The reason code shall be UICCException. RECORD_NOT_FOUND.
- CRRP5: If the specified offset into the selected record recOffset is less than 0, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT_OF_RECORD_BOUNDARIES.
- CRRP6: If recOffset plus dataLength is greater than the record length, an instance of UICCException shall be thrown. The reason code shall be UICCException.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 UICCException shall be thrown. The reason code shall be UICCException.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 UICCException shall be thrown. The reason code shall be UICCException.INVALID_MODE or UICCException.COMMAND_INCOMPATIBLE.
- 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.

5.1.1.12.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the currently selected EF is neither linear fixed nor cyclic, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND_INCOMPATIBLE.
- CRRC3: If the calling applet does not fulfil the access condition, UPDATE, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException. SECURITY_STATUS_NOT_SATISFIED.
- CRRC4: If the currently selected EF is deactivated and the file status of the EF does not allow for updating an deactivated file, an instance of UICCException shall be thrown. The reason code shall be UICCException. REF_DATA_INVALIDATED.
- CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.
- CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.1.1.12.2 Test area files

Test Source: Test_Api_1_Fvw_Updr.java.

Test Applet: Api_1_Fvw_ Updr_1.java.

Cap File: api_1_fvw_updr.cap.

5.1.1.12.3 Test coverage

CRR number	Test case number
N1	2, 3, 4, 5, 7, 8, 10
N2	2
N3	3
N4	5
N5	4
N6	7, 8, 9, 10
N7	7
N8	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	Not testable
C6	Not testable

5.1.1.12.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	**Get the UICC FileView** FileView.getTheUICCView(CLEAR_ON_RESET)		
1	**No EF selected** RecNumber = 1 mode = REC_ACC_MODE_ABSOLUTE recOffset = 0 byte[] data = new byte[20] dataOffset = 0 dataLength = 10 updateRecord()	Shall throw uicc.access.UICC Exception with reason code NO_EF_SELECTED.	
2	**Update Absolute from Linear Fixed EF** 1- select DF_{TEST}, fid = 1111 2- select EF_{LARU}, fid = 6F0C // Record pointer not set. 3- recNumber = 2 mode = REC_ACC_MODE_ABSOLUTE data[0:3] = '11' recOffset = 0 dataOffset = 0 dataLength = 4 updateRecord() respOffset = 0 respLength = 0 readRecord() 4- // verify result read respOffset = 0 respLength = 4 recNumber = 0 mode = REC_ACC_MODE_CURRENT readRecord()	1- No exception shall be thrown. 2- No exception shall be thrown. 3- No exception shall be thrown. 4- Resp shall be: 11 11 11 11	

3	**Update Current from Linear Fixed EF**
	1- select DF_{TEST}, fid = 1111
	2- select EF_{LARU}, fid = 6F0C
	// 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_MODECURRENT
	updateRecord()
	5- //verify result
	respOffset = 0
	respLength = 4
	recNumber = 0
	mode = REC_ACC_MODE_ CURRENT
	readRecord()
		1- No exception shall be thrown.
		2- No exception shall be thrown.
		3- No exception shall be thrown.
		5- 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- select DF_{TEST}, fid = 1111 2- select EF_{LARU}, fid = 6F0C 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 4-// verify result readRecord()	1- No exception shall be thrown. 2- No exception shall be thrown. 4- No exception shall be thrown. resp shall be: resp[0] = '33' resp[1] = '33' resp[2] = '33' resp[3] = '33'	

Id	Description	API Expectation	APDU Expectation

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- //verify result
	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 uicc.access.UICCException with reason code RECORD_NOT_FOUND.	

7	**Update Previous from Linear Fixed EF, no record pointer set**
	1- select DF_{TEST}, fid = 1111
	2- select EF_{LARU}, fid = 6F0C
	3- recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	data[0:3] = '66'
	dataOffset = respOffset = 0
	dataLength = respLength = 4
	updateRecord()
	4- //verify result
	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'	

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()
	2- //verify result
	readRecord()
		1- No exception shall be thrown
		
		2- No exception shall be thrown.
		esp shall be:
		resp[0] = '77'
		resp[1] = '77'
		resp[2] = '77'
		resp[3] = '77'	

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	Shall throw sim.access.SIMViewException with reason code RECORD_NOT_FOUND.	

Id	Description	API Expectation	APDU Expectation

10	**Update Previous from Cyclic EF**
	1- select DF_{TEST}, fid = 1111
	2- select EF_{CARU}, fid = 6F09
	3- recNumber = 2
	mode = REC_ACC_MODE_ABSOLUTE
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
	4- recNumber = 2
	mode = REC_ACC_MODE_PREVIOUS
	data[0:2] = 'FF'
	dataOffset = 0
	dataLength = 3
	updateRecord()
	5- //verify result
	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] = 'FF'resp[1] = 'FF'
		resp[2] = 'FF'	

11	**Update Absolute from Linear Fixed EF beyond Records** 1- select EF_{LARU}, fid = 6F0C 2-recNumber = -1 mode = REC_ACC_MODE_ABSOLUTE recOffset = 0 dataOffset = 0 dataLength = 4 updateRecord() 3- recNumber = 3 updateRecord()	1- No exception shall be thrown. 2- Shall throw uicc.access.UICCException with reason code RECORD_NOT_FOUND. 3- Shall throw uicc.access.UICCException with reason code RECORD_NOT_FOUND.	

12	**No current record in linear fixed EF, update current**
	1- select EF_{LARU}, fid = 6F0C
	// No curr rec
	2- recNumber = 0 // curr rec
	mode = REC_ACC_MODE _CURRENT
	recOffset = 0
	dataOffset = 0
	dataLength = 4
	updateRecord()
		1 - No exception shall be thrown.
		
		2 - Shall throw uicc.access.UICCException with reason code RECORD_NOT_FOUND.	

13	**recOffset < 0**
	1- select EF_{LARU}, fid = 6F0C
	2- recNumber = 1 // rec 1
	mode = REC_ACC_MODE_ABSOLUTE
	dataOffset = 0
	dataLength = 4
	updateRecord()
		1- No exception shall be thrown.
		2- Shall throw uicc.access.UICC Exception with reason code OUT_OF_RECORD_BOUNDARIES.	

14	**recOffset + dataLength > record.length** 1- select EF_{LARU}, fid = 6F0C 2- recNumber = 1 mode = REC_ACC_MODE_ABSOLUTE recOffset = 2 dataOffset = 0 dataLength = 4 updateRecord()	1- No exception shall be thrown. 2- Shall throw uicc.access.UICC Exception with reason code OUT_OF_RECORD_BOUNDARIES.	
15	**Updating with invalid mode** 1- select EF_{LARU}, fid = 6F0C 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 uicc.access.UICC Exception with reason code INVALID_MODE. 3 - Shall throw uicc.access. UICC Exception with reason code INVALID_MODE.	

Id	Description	API Expectation	APDU Expectation
16	**Updating Cyclic EF with invalid mode** 1- select DF_{TEST}, fid = 1111 2- select EF_{CARU}, fid = 6F09 set record pointer to record nr 1 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_CURRENT updateRecord() 5- recNumber = 2 mode = REC_ACC_MODE_ABSOLUTE updateRecord() 6- recNumber = 0 mode = 5 updateRecord()	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - Shall throw uicc.access. UICC Exception with reason code INVALID_MODE or COMMAND_INCOMPATIBLE. 4 - Shall throw uicc.access.UICC Exception with reason code INVALID_MODE or COMMAND_INCOMPATIBLE. 5 - Shall throw uicc.access. UICC Exception with reason code INVALID_MODE or COMMAND_INCOMPATIBLE. 6 - Shall throw uicc.access. UICC Exception with reason code INVALID_MODE or COMMAND_INCOMPATIBLE.	
17	**data[] is null** data[] = 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- select DF_{TEST}, fid = 1111 2- select EF_{TNR}, fid = 6F01 3- dataOffset = 0 dataLength = 4 updateRecord()	1- No exception shall be thrown. 2- No exception shall be thrown. 3- Shall throw uii.access.UICC Exception with reason code COMMAND_INCOMPATIBLE.	
22	**Access condition not fulfilled** 1- select EF_{CNU}, fid = 6F05 2- recOffset = 0 dataOffset = 0 dataLength = 1 mode = REC_ACC_MODE_PREVIOUS updateRecord() 3- fid = EFLNU select() 4- recNumber = 1 mode = REC_ACC_MODE_ CURRENT recOffset = 0 dataOffset = 0 dataLength = 1 updateRecord()	1- No exception shall be thrown. 2- Shall throw uicc.access.UICC Exception with reason code SECURITY_STATUS_NOT_SATISFIED. 3- No exception shall be thrown. 4- Shall throw uicc.access.UICC Exception with reason code SECURITY_STATUS_NOT_SATISFIED	

Id	Description	API Expectation	APDU Expectation
23	**EF is deactivated** 1- select EF_{CNR}, fid = 6F04 invalidate() 2- updateRecord() 3- activateFile() 4- restore the file content EF_{LARU}, EF_{CARU}	1- No exception shall be thrown. 2- Shall throw uicc.access.UICCException with reason codeREF_DATA_INVALIDATED 3- No exception shall be thrown.	
24	**Restore the file content** 1- restore the file content of EF_{LARU}: record 1 = 0x55,0x55,0x55,0x55 record 2 = 0xAA,0xAA,0xAA,0xAA 2- restore the file content of EF_{CARI}: record 1 = 0x55,0x55,0x55,0x55 record 2 = 0xAA,0xAA,0xAA,0xAA		

5.1.2 Interface UICCConstants

This interface hold all the constants defined in ETSI TS 102 221 [5]. No test of constants will be performed.

5.1.3 Interface UICCSystem

5.1.3.1 Method getTheUICCView

Test Area Reference: Api_1_Usy_Getfb.

5.1.3.1.1 Conformance requirement:

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

```
public static FileView getTheUICCView(byte event)
                throws javacard.framework.SystemException
```

5.1.3.1.1.1 Normal execution

- CRRN1: returns a reference to class which implements the FileView interface on the UICC file system.
- CRRN2: return null if one of the javacard.framework.Applet.register(..) method has not previously invoked by the applet invoking this method or the file system server returns null.
- CRRN3: It is not possible to get access to files which are located under any ADF with this FileView.
- CRRN4: After a successful invocation of the method, the MF is the current selected file.
- CRRN5: A separate and independent file context shall be associated with each and every FileView object: the operation performed on files in a given FileView object shall not affect the file context associated with any other FileView object. This context can be transient or persistent depending on what was required by the Applet during the creation of the FileView object.

5.1.3.1.1.2 Parameter errors

- CRRP1: If event is not one of the following values JCSystem.NOT_A_TRANSIENT_OBJECT, JCSystem.CLEAR_ON_DESELECT, or JCSystem.CLEAR_ON_RESET a SystemException is thrown. The value of the SystemException shall be SystemException.ILLEGAL_VALUE.

5.1.3.1.1.3 Context errors

- CRRC1: If event is JCSystem.CLEAR_ON_RESET or JCSystem.CLEAR_ON_DESELECT and not enough transient memory space is available a SystemException shall be thrown with reason code SystemException.NO_TRANSIENT_SPACE.
- CRRC2: If event is JCSystem.CLEAR_ON_DESELECT and the applet is not a currently selected applet a SystemException with reason code SystemException.ILLEGAL_TRANSIENT shall be thrown.

5.1.3.1.2 Test area files

Test Source: Test_ Api_1_Usy_Getfb.java.

Test Applet: Api_1_Usy_Getfb_1.java.

Cap File: api_1_usy_getfb.cap.

5.1.3.1.3 Test coverage

CRR number	Test case number
N1	2
N2	1
N3	2
N4	2
N5	3
P1	7
C1	5, 6
C2	4

5.1.3.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Method returns null** Install Applet1 with full access rights on the UICC file system Invoke the method getTheUICCView before the javacard.framework.Applet.register(..) method invocation	The method returns null	

Id	Description	API Expectation	APDU Expectation
2	**Normal execution**		
	1- Envelope menu selection is sent to the UICC	1- Applet1 is triggered	
	Invoke the method getTheUICCView() with the event JCSystem.NOT_A_TRANSIENT_OBJECT and stores the result in a class variable FV1	No exception shall be thrown	
	Applet1 calls FV1.status() command	Current selected DF is the MF	
	Select DF_{Test} using FV1 Select EF_{TARU} using FV1 Read first 3 bytes using FV1	Expected value is {FF FF FF}	
	Reset Terminal profile		
	2- Envelope menu selection is sent to the UICC	2- Applet1 is triggered	
	Applet1 calls FV1.status() command	Current selected DF is DF_{Test}	
	Read first 3 bytes using FV1 Applet1 calls FV1.select(0x7FFF)	Expected value is {FF FF FF} UICCException.FILE_NOT_FOUND is thrown	
	Invoke the method getTheUICCView() with the event JCSystem.CLEAR_ON_RESET and stores the result in a class variable FV2	No exception shall be thrown	
	Applet1 calls FV2.status() command	Current selected DF the MF	
	Select DF_{Test} using FV2 Select EF_{TARU} using FV2 Read first 3 bytes using FV2	Expected value is {FF FF FF}	
	Reset Terminal profile		
	4 - Envelope menu selection is sent to the UICC	4- Applet1 is triggered	
	Applet1 calls FV2.status() command	Current selected DF is the MF	
	Read first 3 bytes using FV2	UICCException. NO_EF_SELECTED	
	Applet1 calls FV2.select(0x7FFF)	UICCException.FILE_NOT_FOUND is thrown	
	5- Select the Applet by AID	5- Applet1 is selected	
	Invoke the method in the method process() getTheUICCView() with the event JCSystem.CLEAR_ON_DESELECT and stores the result in a class variable FV3	No exception shall be thrown	
	Applet1 calls FV3.status() command	Current selected DF the MF	
	Select DF_{Test} using FV3 Select EF_{TARU} using FV3 Read first 3 bytes using FV3	Expected value is {FF FF FF}	
	Select ADF2 by AID		
	6- Select the Applet by AID	6- Applet1 is selected	
	Applet1 calls FV3.status() command		
	Read first 3 bytes using FV3		

Id	Description	API Expectation	APDU Expectation
	Applet1 calls FV3.select(0x7FFF)	Current selected DF is the MF UICCException. NO_EF_SELECTED UICCException.FILE_NOT_FOUND is thrown	
3	**FileView context independency** 1- Envelope menu selection is sent to the UICC 2- Check that previous fileviews are different (FV1 != FV2 != FV3) 3- Select DF_{Test}/EF_{LARU} using FV1 4- Select DF_{Test}/EF_{CARU} using FV2 5- Select DF_{Test}/EF_{CARU} using FV3 6- Read record number 1 using FV1 (in absolute mode) 7- Read record number 2 using FV2 (in absolute mode)	1- Applet1 is triggered 3- No exception shall be thrown 4- No exception shall be thrown 5- A security exception shall be thrown 6- Expected value is “55 55 55 55” 7- Expected value is “AA AA AA”	
4	**ILLEGAL_TRANSIENT SystemException** 1- Envelope menu selection is sent to the UICC 2- Applet1 calls getTheUICCView() method with the event JCSystem.CLEAR_ON_DESELECT	1- Applet1 is triggered 2- SystemException. ILLEGAL_TRANSIENT is thrown	
5	**NO_TRANSIENT_SPACE SystemException with CLEAR_ON_RESET FileView object** 1- Get the available transient memory space using method length=JCSystem.getAvailableMemory(MEMORY_TYPE_TRANSIENT_RESET) 2- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR_ON_RESET) 3- Applet calls method getTheUICCView() with event JCSystem.CLEAR_ON_RESET 4- Reset	1- No exception shall be thrown 2- No exception shall be thrown 3- SystemException. NO_TRANSIENT_SPACE is thrown	

Id	Description	API Expectation	APDU Expectation
6	**NO_TRANSIENT_SPACE SystemException with CLEAR_ON_DESELECT FileView object** 1- Select the Applet by AID 2- Get the available transient memory space using method length=JCSystem.getAvailableMemory(MEMORY_TYPE_TRANSIENT_DESELECT) 3- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR_ON_DESELECT) 4- Applet calls method getTheUICCView() with event JCSystem.CLEAR_ON_DESELECT	 1- Applet1 is selected 2- No exception shall be thrown 3- No exception shall be thrown 4- SystemException. NO_TRANSIENT_SPACE is thrown	
7	**ILLEGAL_VALUE SystemException** 1- Invoke the method getTheUICCView() with event different from 0,1,2	 1- SystemException.ILLEGAL_VALUE is thrown	

5.1.3.2 Method getTheFileView

Test Area Reference: Api_1_Usy_Getfob.

5.1.3.2.1 Conformance requirement:

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

```
public static FileView getTheFileView(javacard.framework.AID aid, byte event)
               throws NullPointerException,
                      javacard.framework.SystemException
```

5.1.3.2.1.1 Normal execution

- CRRN1: returns a reference to class which implements the FileView interface on an ADF file system defined by its AID.
- CRRN2: returns null if the ADF with the AID does not exist.
- CRRN3: returns null if one of the javacard.framework.Applet.register(..) method has not previously invoked by the applet invoking this method or the filesystem server does not exist or the filesystem server returns null.
- CRRN4: After a successful invocation of the method the ADF is the currently selected file.
- CRRN5: A separate and independent file context shall be associated with each and every FileView object: the operation performed on files in a given FileView object shall not affect the file context associated with any other FileView object. This context can be transient or persistent depending on what was required by the Applet during the creation of the FileView object.
- CRRN6: It is not possible to access files which are not located under the ADF.

5.1.3.2.1.2 Parameters error

- CRRP1: If event is not one of the following values JCSystem.NOT_A_TRANSIENT_OBJECT, JCSystem.CLEAR_ON_DESELECT, or JCSystem.CLEAR_ON_RESET a SystemException is thrown. The value of the SystemException shall be SystemException.ILLEGAL_VALUE.
- CRRP2: If the AID is null a NullPointerException shall be thrown.

5.1.3.2.1.3 Context errors

- CRRC1: If event is JCSystem.CLEAR_ON_RESET or JCSystem.CLEAR_ON_DESELECT and not enough transient memory space is available a SystemException shall be thrown with reason code SystemException.NO_TRANSIENT_SPACE.
- CRRC2: If event is JCSystem.CLEAR_ON_DESELECT and the applet is not a currently selected applet a SystemException with reason code SystemException.ILLEGAL_TRANSIENT shall be thrown.

5.1.3.2.2 Test area files

Test Source: Test_ Api_1_Usy_Getfob.java.

Test Applet: Api_1_Usy_Getfob.java.

Cap File: api_1_usy_getfob.cap.

5.1.3.2.3 Test coverage

CRR number	Test case number
N1	1 to 3
N2	1
N3	1
N4	2
N5	3
N6	2
P1	7
P2	8
C1	5, 6
C2	4

5.1.3.2.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Method returns null** 1- Install Applet1 with full access rights on the UICC file system 2- Invoke the method getTheFileView before the javacard.framework.Applet.register(..) method invocation 3- Envelope menu selection is sent to the UICC 4- Invoke the method getTheFileView() with AID = unknown ADF AID	2- returns null 3- applet is triggered 4- returns null	

Id	Description	API Expectation	APDU Expectation
2	**Normal execution** 1- Envelope menu selection is sent to the UICC Invoke the method getTheFileView() with AID = ADF1 with the event JCSystem.NOT_A_TRANSIENT_OBJECT and stores the result in a class variable FV1 Applet1 calls FV1.status() command Select DF_{Test} using FV1 Select EF_{TARU} using FV1Read first 3 bytes using FV1ResetTerminal profile 2 - Envelope menu selection is sent to the UICC Applet1 calls FV1.status() command Read first 3 bytes using FV1 Applet1 calls FV1.select(0x3F00) Invoke the method getTheFileView() with the event JCSystem.CLEAR_ON_RESET and stores the result in a class variable FV2 Applet1 calls FV2.status() command Select DF_{Test} using FV2 Select EF_{TARU} using FV2 Read first 3 bytes using FV2 Reset Terminal profile 4 - Envelope menu selection is sent to the UICC Applet1 calls FV2.status() command Read first 3 bytes using FV2 Applet1 calls FV2.select(0x3F00) 5- Select the Applet by AID Invoke the method getTheFileView() with AID = ADF1with theevent:JCSystem.CLEAR_ON_DESELECT and stores the result in a class variable FV3 Applet1 calls FV3.status() command Select DF_{Test} using FV3 Select EF_{TARU} using FV3 Read first 3 bytes using FV3 6- Select the Applet by AID Applet1 calls FV3.status() command Read first 3 bytes using FV3 Applet1 calls FV3.select(0x3F00)	1- Applet1 is triggered No Exception shall be thrown Current selected DF is ADF1 Expected value is {FF FF FF} 2- Applet1 is triggered Current selected DF is DF_{Test} Expected value is {FF FF FF} UICCException.FILE_NOT_FOUND is thrown No exception shall be thrown Current selected DF is the ADF1 Expected value is {FF FF FF} 4- Applet1 is triggered Current selected DF is the ADF1 UICCException. NO_EF_SELECTED UICCException.FILE_NOT_FOUND is thrown 5- Applet1 is selected No Exception shall be thrown Current selected DF is ADF1 Expected value is {FF FF FF} 6- Applet1 is selected Current selected DF is ADF1	

Id	Description	API Expectation	APDU Expectation
		UICCException. NO_EF_SELECTED UICCException.FILE_NOT_FOUND is thrown	
3	**FileView context independency** 1- Envelope menu selection is sent to the UICC 2- Check that previous fileviews are different (FV1 != FV2 != FV3) 3- Select DF_{Test}/EF_{LARU} using FV1 4- Select DF_{Test}/EF_{CARU} using FV2 5- Select DF_{Test}/EF_{CARU} using FV3 6- Read record number 1 using FV1 (in absolute mode) 7- Read record number 2 using FV2 (in absolute mode)	1- Applet1 is triggered 3- No exception shall be thrown 4- No exception shall be thrown 5- A security exception shall be thrown 6- Expected value is “55 55 55 55” 7- Expected value is “AA AA AA”	
4	**ILLEGAL_TRANSIENT SystemException** 1- Envelope menu selection is sent to the UICC 2- Applet1 calls getTheFileView() method with the event JCSystem.CLEAR_ON_DESELECT	1- Applet1 is triggered 2- SystemException. ILLEGAL_TRANSIENT is thrown	
5	**NO_TRANSIENT_SPACE SystemException with CLEAR_ON_RESET FileView object** 1 Get the available transient memory space using method length=JCSystem.getAvailableMemory(MEMORY_TYPE_TRANSIENT_RESET) 2- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR_ON_RESET) 3- Applet calls method getTheFileView() with AID = ADF1 with event JCSystem.CLEAR_ON_RESET 4- Reset	1- No Exception shall be thrown 2- No Exception shall be thrown 3- SystemException. NO_TRANSIENT_SPACE is thrown	
6	**NO_TRANSIENT_SPACE SystemException with CLEAR_ON_DESELECT FileView object** 1 - Select the Applet by AID 2- Get the available transient memory space using method length=JCSystem.getAvailableMemory(MEMORY_TYPE_TRANSIENT_DESELECT) 3- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR_ON_DESELECT) } 4- Applet calls method getTheFileView() with AID = ADF1 with event: JCSystem.CLEAR_ON_DESELECT	1- Applet1 is triggered 2- No Exception shall be thrown 3- No Exception shall be thrown 4- SystemException. NO_TRANSIENT_SPACE is thrown	

Id	Description	API Expectation	APDU Expectation
7	**ILLEGAL_VALUE SystemException** 1- Invoke the method getTheFileView() with event different from 0,1,2	1- SystemException.ILLEGAL_VALUE is thrown	
8	**NullPointerException** Invoke the method getTheFileView() with AID = NULL with event: 1 - JCSystem.CLEAR_ON_RESET	1- Shall be thrown java.lang.NullPointerException	

5.1.3.3 Method getTheFileView

Test Area Reference: Api_1_Usy_Getf_Bsbb.

5.1.3.3.1 Conformance requirement:

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

```
public static FileView getTheFileView(byte[] buffer, short bOffset, short bLength, byte event)
                throws NullPointerException,
                       javacard.framework.SystemException,
                       ArrayIndexOutOfBoundException
```

5.1.3.3.1.1 Normal execution

- CRRN1: returns a reference to class which implements the FileView interface on an ADF file system defined by its AID contains in buffer parameter.
- CRRN2: returns null if the ADF with the full AID given in the buffer does not exist.
- CRRN3: returns null if one of the javacard.framework.Applet.register(..) method has not previously invoked by the applet invoking this method or the filesystem server does not exist or the filesystem server returns null.
- CRRN4: After a successful invocation of the method the ADF is the currently selected file.
- CRRN5: A separate and independent file context shall be associated with each and every FileView object: the operation performed on files in a given FileView object shall not affect the file context associated with any other FileView object. This context can be transient or persistent depending on what was required by the Applet during the creation of the FileView object.
- CRRN6: It is not possible to access files which are not located under the ADF.

5.1.3.3.1.2 Parameters error

- CRRP1: If event is not one of the following values JCSystem.NOT_A_TRANSIENT_OBJECT, JCSystem.CLEAR_ON_DESELECT, or JCSystem.CLEAR_ON_RESET a SystemException is thrown. The value of the SystemException shall be SystemException.ILLEGAL_VALUE.
- CRRP2: If the buffer is null a NullPointerException shall be thrown.
- CRRP3: If bOffset is less than 0, an instance ArrayIndexOutOfBoundException shall be thrown.
- CRRP4: if bOffset plus bLength is greater than the length of the array buffer.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP5: if bLength is not in the range of 5 - 16 bytes a SystemException.ILLEGAL_VALUE shall be thrown.

5.1.3.3.1.3 Context errors

- CRRC1: If event is JCSystem.CLEAR_ON_RESET or JCSystem.CLEAR_ON_DESELECT and not enough transient memory space is available a SystemException shall be thrown with reason code SystemException.NO_TRANSIENT_SPACE.
- CRRC2: If event is JCSystem.CLEAR_ON_DESELECT and the applet is not a currently selected applet a SystemException with reason code SystemException.ILLEGAL_TRANSIENT shall be thrown.

5.1.3.3.2 Test area files

Test Source: Test_ Api_1_Usy_Getf_Bsbb.java.

Test Applet: Api_1_Usy_Getf_Bsbb.java.

Cap File: api_1_usy_getf_bsbb.cap.

5.1.3.3.3 Test coverage

CRR number	Test case number
N1	2
N2	1
N3	2
N4	2
N5	3
N6	2
P1	7
P2	8
P3	9
P4	9
P5	10
C1	5, 6
C2	4

5.1.3.3.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Method returns null** 1- Install Applet1 with full access rights on the UICC file system 2- Invoke the method getTheFileView before the javacard.framework.Applet.register(..) method invocation Invoke the method getTheFileView() with buffer[20] = {ADF1,…} bOffset= 0 bLength= 16 3- Envelope menu selection is sent to the UICC 4- Invoke the method getTheFileView before the javacard.framework.Applet.register(..) method invocation Invoke the method getTheFileView() with buffer[20] = null bOffset= 0 bLength= 16	2- returns null 3- Applet is triggered 4- returns null	

Id	Description	API Expectation	APDU Expectation
2	**Normal execution** 1- Envelope menu selection is sent to the UICC Invoke the method getTheFileView() with buffer[20] = {ADF1,…} bOffset= 0 bLength= 16 JCSystem.NOT_A_TRANSIENT_OBJECT and stores the result in a class variable FV1 Applet1 calls FV1.status() command Select DF_{Test} using FV1 Select EF_{TARU} using FV1 Read first 3 bytes using FV1 Reset Terminal profile 2 - Envelope menu selection is sent to the UICC Applet1 calls FV1.status() command Read first 3 bytes using FV1 Applet1 calls FV1.select(0x3F00) Invoke the method getTheFileView() with the event JCSystem.CLEAR_ON_RESET and stores the result in a class variable FV2 Applet1 calls FV2.status() command Select DF_{Test} using FV2 Select EF_{TARU} using FV2 Read first 3 bytes using FV2 Reset Terminal profile 4 - Envelope menu selection is sent to the UICC Applet1 calls FV2.status() command Read first 3 bytes using FV2 Applet1 calls FV2.select(0x3F00) 5- Select the Applet by AID Invoke the method getTheFileView() with AID = ADF1 with buffer[20] = {ADF1,…} bOffset= 0 bLength= 16 the event:JCSystem.CLEAR_ON_DESELECT and stores the result in a class variable FV3 Applet1 calls FV3.status() command Select DF_{Test} using FV3 Select EF_{TARU} using FV3 Read first 3 bytes using FV3 6- Select the Applet by AID Applet1 calls FV3.status() command Read first 3 bytes using FV3 Applet1 calls FV3.select(0x3F00)	1- Applet1 is triggered No Exception shall be thrown Current selected DF is ADF1 Expected value is {FF FF FF} 2- Applet1 is triggered Current selected DF is DF_{Test} Expected value is {FF FF FF} UICCException.FILE_NOT_FOUND is thrown No exception shall be thrown Current selected DF is ADF1 Expected value is {FF FF FF} 4- Applet1 is triggered Current selected DF is ADF1 UICCException. NO_EF_SELECTED UICCException.FILE_NOT_FOUND is thrown 5- Applet1 is selected No Exception shall be thrown Current selected DF is ADF1 Expected value is {FF FF FF} 6- Applet1 is selected Current selected DF is ADF1 UICCException. NO_EF_SELECTED	

Id	Description	API Expectation	APDU Expectation
		UICCException.FILE_NOT_FOUND is thrown	
3	**FileView context independency** 1- Envelope menu selection is sent to the UICC 2- Check that previous fileviews are different (FV1 != FV2 != FV3) 3- Select DF_{Test}/EF_{LARU} using FV1 4- Select DF_{Test}/EF_{CARU} using FV2 5- Select DF_{Test}/EF_{CARU} using FV3 6- Read record number 1 using FV1 (in absolute mode) 7- Read record number 2 using FV2 (in absolute mode)	1- Applet1 is triggered 3- No exception shall be thrown 4- No exception shall be thrown 5- A security exception shall be thrown 6- Expected value is “55 55 55 55” 7- Expected value is “AA AA AA”	
4	**ILLEGAL_TRANSIENT SystemException** 1- Envelope menu selection is sent to the UICC 2- Applet1 calls getTheFileView() method with buffer[20] = {ADF1,…} bOffset= 0 bLength= 16 with the event JCSystem.CLEAR_ON_DESELECT	1- Applet1 is triggered 2- SystemException. ILLEGAL_TRANSIENT is thrown	
5	**NO_TRANSIENT_SPACE SystemException with CLEAR_ON_RESET FileView object** 1 Get the available transient memory space using method length=JCSystem.getAvailableMemory(MEMORY_TYPE_TRANSIENT_RESET) 2- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR_ON_RESET) 3- Applet calls method getTheFileView() with buffer[20] = {ADF1,…} bOffset= 0 bLength= 16 with the event JCSystem.CLEAR_ON_RESET 4- Reset	1- No Exception shall be thrown 2- No Exception shall be thrown 3- SystemException. NO_TRANSIENT_SPACE is thrown	

Id	Description	API Expectation	APDU Expectation
6	**NO_TRANSIENT_SPACE SystemException with CLEAR_ON_DESELECT FileView object** 1 - Select the Applet by AID 2- Get the available transient memory space using method length=JCSystem.getAvailableMemory(MEMORY_TYPE_TRANSIENT_DESELECT) 3- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR_ON_DESELECT) } 4- Applet calls method getTheFileView() with buffer[20] = {ADF1,…} bOffset= 0 bLength= 16 with event: JCSystem.CLEAR_ON_DESELECT	1- Applet1 is triggered 2- No Exception shall be thrown 3- No Exception shall be thrown 4- SystemException. NO_TRANSIENT_SPACE is thrown	
7	**ILLEGAL_VALUE SystemException** 1- Invoke the method getTheFileView() with event different from 0,1,2	1- SystemException.ILLEGAL_VALUE is thrown	
8	**NullPointerException** Invoke the method getTheFileView() with buffer[20] = null bOffset= 0 bLength= 16 with event: 1 - JCSystem.CLEAR_ON_RESET	1- Shall be thrown java.lang.NullPointerException	
9	**ArrayIndexOutOfBoundsException** 1-Envelope menu selection is sent to the UICC Invoke the method getTheFileView() with buffer[20] = {ADF1,…} bOffset= 5 bLength= 16 event =JCSystem. CLEAR_ON_RESET Invoke the method getTheFileView() with buffer[20] = {ADF1,…} bOffset= -1 bLength= 16 event =JCSystem. CLEAR_ON_RESET	1- Applet1 is triggered Shall be thrown ArrayIndexOutOfBoundsException Shall be thrown ArrayIndexOutOfBoundsException	
10	**SystemException.ILLEGAL_VALUE** 1-Envelope menu selection is sent to the UICC Invoke the method getTheFileView() with buffer[20] = {ADF1,…} bOffset= 0 bLength= 4 event =JCSystem. CLEAR_ON_RESET Invoke the method getTheFileView() with buffer[20] = {ADF1,…} bOffset= 0 bLength= 17 event =JCSystem. CLEAR_ON_RESET	1- Applet1 is triggered SystemException.ILLEGAL_VALUE shall be thrown SystemException.ILLEGAL_VALUE shall be thrown	

5.1.4 Interface UICCException

5.1.4.1 Constructor

Test Area Reference: Api_1_Uex_Coor.

5.1.4.1.1 Conformance Requirement:

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

```
public UICCException(short reason)
                throws UICCException
```

5.1.4.1.1.1 Normal execution

- CRRN1: Construct an UICCException with the specified reason.

5.1.4.1.1.2 Parameter errors

No requirements.

5.1.4.1.1.3 Context errors

No requirements.

5.1.4.1.2 Test suite files

Test Source: Test_Api_2_Uex_Coor.java.

Test Applet: Api_2_Uex_Coor_1.java.

Cap File: api_2_uex_coor.cap.

5.1.4.1.3 Test Coverage

CRR number	Test case number
N1	1

5.1.4.1.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**UICCException with the specified reason** (The reason shall set with setReason and compare the Exception with getReason)	Reason (specified)	

5.1.4.2 Method throwIt

Test Area Reference: Api_1_Uex_Thit.

5.1.4.2.1 Conformance Requirement:

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

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

5.1.4.2.1.1 Normal execution

- CRRN1: Throws the JCRE owned instance of UICCException with the specified reason.
- CRRN2: extends javacard.framework.CardRuntimeException.

5.1.4.2.1.2 Parameter errors

No requirements.

5.1.4.2.1.3 Context errors

No requirements.

5.1.4.2.2 Test area files

Test Source: Test_Api_2_Uex_Thit.java.

Test Applet: Api_2_Uex_Thit_1.java.

Cap File: api_2_uex_thit.cap.

5.1.4.2.3 Test Coverage

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

5.1.4.2.4 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	Throws the JCRE instance of UICCException with the specified reason	Reason = 0	
2	Throws the JCRE instance of UICCException with the specified reason	Reason = 1	
3	Throws the JCRE instance of UICCException with the specified reason	Reason = 0xA55A	
4	UICCException extends javacard.framework.CardRuntimeException	Reason = 0	
5	UICCException extends javacard.framework.CardRuntimeException	Reason = 1	
6	UICCException extends javacard.framework.CardRuntimeException	Reason = 0xA55A	

5.1.4.3 Reason Codes

This part contain only constant defined for the available reason code. No test of constants will be performed.

5.1.5 Contexts

5.1.5.1 Context tests

Test Area Reference: Api_1_Cont

5.1.5.1.1 Conformance requirement

5.1.5.1.1.1 Normal execution

- CRRN1: A file (EF, DF or ADF) can be accessed (selected, read, updated, deleted, deactivated, activated, increased, searched, resized) concurrently by different UICC-based applications such as toolkit applications.
- CRRN2: A file (EF, DF or ADF) created by a FileView can be accessed by other applications and vice versa.
- CRRN3: If a file is indicated as shareable (in the file descriptor byte in the FCP), then applications may perform authorized operations (selected, read, updated, deleted, deactivated, activated, increased, searched, resized, deleted) on the file independently of whether or not the file is the current file of any other application. A consequence is that if changes to a shareable file are permitted by the file's security conditions, then the file can be changed by one application while it is currently selected and being used by a second application.
- CRRN4: If a file is indicated as non-shareable and is the current file of one application, then another application cannot perform any operation on the file regardless of authorization. A consequence is that an application acquires exclusive access to a not-shareable file by successfully selecting it. Access by any other application, including an attempt to select the file, shall return an error indication.
- CRRN5: Concurrent access to a file by two executing instances of a single application is considered to be accessed by two different applications.

5.1.5.1.1.2 Parameter errors

No requirements.

5.1.5.1.1.3 Context errors

No requirements.

5.1.5.1.2 Test area files

Test Source: Api_1_Cont.java.

Test Applet: Api_1_Cont _1.java.

Cap File: api_1_cont.cap.

5.1.5.1.3 Test coverage

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

5.1.5.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	1- Select DF $_{TELECOM}$ 2- Trigger applet 3- Applet gets object UICCFileView1 using getTheUICCView() 4- Applet gets object UICCFileView2 using getTheUICCView() 5- Applet gets object ADF1FileView1 using getTheFileView() 6- Applet gets object ADF1FileView2 using getTheFileView() 7- Applet gets object UICCAdminFileView1 using getTheUICCAdminFileView () 8- Applet gets object UICCAdminFileView2 using getTheUICCAdminFileView () 9- Applet gets object ADF1AdminFileView1 using getTheAdminFileView () 10- Applet gets object ADF1AdminFileView2 using getTheAdminFileView ()		
1	**Select and status** 1- UICCFileView1.select() DF_{TEST}, UICCFileView1.select() EF_{TARU}, 2- UICCFileView1.select() ADF1 3- UICCFileView2.select() EF_{UICC}, 4- UICCFileView2.select() ADF1 5- ADF1FileView1.select() $DF_{TELECOM}$, 6- ADF1FileView1.select() MF 7- ADF1FileView2.select() DF_{TEST}, ADF1FileView2.select() DF_{SUB_TEST} 8- ADF1FileView2.select() MF 9- UICCFileView1.status() 10- UICCFileView2. status() 11- ADF1FileView1. status() 12- ADF1FileView2. status() 13- Status	2- UICCException.FILE_NOT_FOUND is thrown 4- UICCException.FILE_NOT_FOUND is thrown 6- UICCException.FILE_NOT_FOUND is thrown 8- UICCException.FILE_NOT_FOUND is thrown 9- FCP corresponding to DF_{TEST} (under MF) is returned 10- FCP corresponding to MF is returned 11- FCP corresponding to $DF_{TELECOM}$ (under ADF1) is returned 12- FCP corresponding to DF_{SUB_TEST} (under ADF1) is returned Applet finalizes	13- FCP corresponding to DF $_{TELECOM}$ (under MF) is returned

Id	Description	API Expectation	APDU Expectation
2	**Select SFI** 1- UICCFileView1.select() DF_{TEST}, UICCFileView1.select() EF_{TNR} with SFI 01, UICCFileView1.deactivate() 2- UICCFileView1.select() SFI 06 3- UICCFileView2.select() DF_{TEST}, UICCFileView2.select() EF_{TARU} with SFI 03, UICCFileView2.updateBinary() 01 01 4- UICCFileView2.select() SFI 08 5- ADF1FileView1.select() DF_{TEST}, ADF1FileView1. select() EF_{CNR} with SFI 04, ADF1FileView1.deactivate() 6- ADF1FileView1.select() SFI 06 7- ADF1FileView2.select() DF_{TEST}, ADF1FileView2.select() EF_{TARU} with SFI 03, ADF1FileView2.updateBinary() 02 02 8- ADF1FileView2.select() SFI 08 9- select DF $_{TEST}$, select EF_{TNR} 10- activate 11- select DF $_{TEST}$, select EF_{TARU} 12- readBinary 13- updateBInary FF FF FF… 14- select AID of ADF1, select DF $_{TEST}$, select EF_{CNR} 15- activate 16- select EF_{TARU}, 17- readBinary 18- updateBInary FF FF FF…	1- No exception is thrown 2- UICCException.FILE_NOT_FOUND is thrown 3- No exception is thrown 4- UICCException.FILE_NOT_FOUND is thrown 5- No exception is thrown 6- UICCException.FILE_NOT_FOUND is thrown 7- No exception is thrown 8- UICCException.FILE_NOT_FOUND is thrown Applet finalizes	10- SW=90 00 12- Returns 01 01 15- SW=90 00 17- Returns 02 02
3	**ReadBinary and updateBinary** 1- Reset 2- Trigger the applet 3- UICCFileView1.select() DF_{TEST}, UICCFileView1.select() EF_{TARU}, UICCFileView1.updateBinary() 01 01 01 01, offset 0 4- UICCFileView1.readBinary(), offset 0 5- UICCFileView2.select() DF_{TEST}, UICCFileView2.select() EF_{TARU}, UICCFileView2.updateBinary() 02 02, offset 2 6- UICCFileView2.readBinary(), offset 0 7- ADF1FileView1.select() DF_{TEST}, ADF1FileView1.select() EF_{TARU}, ADF1FileView1.updateBinary() 03 03 03 03, offset 0 8- ADF1FileView1.readBinary(), offset 0	3- No exception is thrown 4- Returns 01 01 01 01 5- No exception is thrown 6- Returns 01 01 02 02 7- No exception is thrown	

Id	Description	API Expectation	APDU Expectation
	9- ADF1FileView2.select() DF_{TEST}, ADF1FileView2.select() EF_{TARU}, ADF1FileView2.updateBinary() 04 04, offset 2	8- Returns 03 03 03 03	
	10- ADF1FileView2.readBinary(), offset 0	9- No exception is thrown	
	11- select DF $_{TEST}$, select EF $_{TARU}$, readBinary	10- Returns 03 03 04 04	
	12- updateBInary FF FF FF…	Applet finalizes	11- Returns 01 01 02 02
	13- select AID of ADF1, select DF $_{TEST}$, select EF $_{TARU}$, readBinary		
	14- updateBInary FF FF FF…		13- Returns 03 03 04 04
4	**SearchRecord**		
	1- Reset		
	2- Trigger the applet		
	3- UICCFileView1.select() DF_{TEST}, UICCFileView1.select() EF_{LUPC}, UICCFileView1.searchRecord() 22 22 22 22 22 22 22 22 22 22	3- returns 2	
	4- UICCFileView1.updateRecord() current 33 33 33 33 33 33 33 33 33 33	4- No exception is thrown	
	5- UICCFileView2.select() DF_{TEST}, UICCFileView2.select() EF_{LUPC}, UICCFileView2. readRecord() current	5- UICCException.RECORD_NOT_FOUND is thrown	
	6- UICCFileView2.searchRecord() 22 22 22 22 22 22 22 22 22 22,	6- returns 0	
	7- UICCFileView2.searchRecord() 33 33 33 33 33 33 33 33 33 33	7- returns 2	
	8- ADF1FileView1.select() DF_{TEST}, ADF1FileView1.select() EF_{LUPC}, ADF1FileView1.searchRecord() 33 33 33 33 33 33 33 33 33 33	8- returns 0	
	9- ADF1FileView1.searchRecord() 22 22 22 22 22 22 22 22 22 22	9- returns 2	
	10- ADF1FileView1.updateRecord() current 11 11 11 11 11 11 11 11 11 11	10- No exception is thrown	
	11- ADF1FileView2.select() DF_{TEST}, ADF1FileView2.select() EF_{LUPC}, ADF1FileView2.searchRecord() 22 22 22 22 22 22 22 22 22 22	11- returns 0	
	12- ADF1FileView2.searchRecord() 11 11 11 11 11 11 11 11 11 11	12- returns 1 and 2 Applet finalizes	
	13- select DF $_{TEST}$, select EF_{LUPC}, readRecord 01		13- returns 11 11 11 11 11 11 11 11 11 11
	14- readRecord 02		14- returns 33 33 33 33 33 33 33 33 33 33
	15- updateRecord 01, 11 11 11 11 11 11 11 11 11 11		
	16- updateRecord 02, 22 22 22 22 22 22 22 22 22 22		
	17- select AID of ADF1, select DF $_{TEST}$, select EF_{LUPC}, readRecord 01		17- returns 11 11 11 11 11

Id	Description	API Expectation	APDU Expectation
	18- readRecord 02 19- updateRecord 01 11 11 11 11 11 11 11 11 11 11 20- updateRecord 02 22 22 22 22 22 22 22 22 22 22		11 11 11 11 11 18- returns 11 11 11 11 11 11 11 11 11 11
5	**readRecord and updateRecord** 1- Reset 2- Trigger the applet 3- UICCFileView1.select() DF_{TEST}, UICCFileView1.select() EF_{LARU}, UICCFileView1.updateRecord() 66 66 66 66, next 4- UICCFileView1.readRecord(), current 5- UICCFileView1.readRecord(), next 6- UICCFileView2.select() DF_{TEST}, UICCFileView2.select() EF_{LARU}, UICCFileView2.readRecord(), current 7- UICCFileView2.updateRecord() BB BB BB BB, record 2 8- UICCFileView2.readRecord(), next 9- UICCFileView2.readRecord(), next 10- ADF1FileView1.select() DF_{TEST}, ADF1FileView1.select() EF_{LARU}, ADF1FileView1.updateRecord() 44 44 44 44, next 11- ADF1FileView1.readRecord(), current 12- ADF1FileView1.readRecord(), next 13- ADF1FileView2.select() DF_{TEST}, ADF1FileView2.select() EF_{LARU}, ADF1FileView2.readRecord(), current 14- ADF1FileView2.updateRecord() 99 99 99 99, record 2 15- ADF1FileView2.readRecord(), next 16- ADF1FileView2.readRecord(), next 17- select DF $_{TEST}$, select EF $_{TARU}$, readRecord next 18- readRecord next 19- updateRecord record 1, 55 55 55 55 20- updateRecord record 2, AA AA AA AA 21- select AID of ADF1, select DF $_{TEST}$, select EF_{LARU}, readRecord next 22- readRecord next 23- updateRecord record 1, 55 55 55 55 24- updateRecord record 2, AA AA AA AA	3- No exception is thrown 4- returns 66 66 66 66 5- returns AA AA AA AA 6- UICCException.RECORD_NOT_FOUND is thrown 7- No exception is thrown 8- returns 66 66 66 66 9- returns BB BB BB BB 10- No exception is thrown 11- returns 44 44 44 44 12- returns AA AA AA AA 13- UICCException.RECORD_NOT_FOUND is thrown 14- No exception is thrown 15- returns 44 44 44 44 16- returns 99 99 99 99 Applet finalizes	17- returns 66 66 66 66 18- returns BB BB BB BB 21- returns 44 44 44 44 22- returns 99 99 99 99

Id	Description	API Expectation	APDU Expectation
6	**ActivateFile and deactivateFile** 1- Reset 2- Trigger the applet 3- UICCFileView1.select() DF_{TEST}, UICCFileView1.select() EF_{TNU}, UICCFileView1.deactivate() 4- UICCFileView2.select() DF_{TEST}, UICCFileView2.select() EF_{TNU}, UICCFileView2.readBinary() 5- UICCFileView2.activate(), UICCFileView2.readBinary() 6- UICCFileView2.deactivate() 7- ADF1FileView1.select() DF_{TEST}, ADF1FileView1.select() EF_{TNU}, ADF1FileView1.readBinary() 8- ADF1FileView1. deactivate() 9- ADF1FileView2.select() DF_{TEST}, ADF1FileView2.select() EF_{TNU}, ADF1FileView2.readBinary() 10- ADF1FileView2.activate(), ADF1FileView2.readBinary() 11- select DF $_{TEST}$, select EF_{TNU} 12- activate 13- select AID of ADF1, select DF $_{TEST}$, select EF_{TNU}	3- No exception is thrown 4- UICCException.REF_DATA_INVALIDATED is thrown 5- returns 55 55 55 6- No exception is thrown 7- returns 55 55 55 8- No exception is thrown 9- UICCException.REF_DATA_INVALIDATED is thrown 10- returns 55 55 55 Applet finalizes	11- SW = 62 83 12- SW = 90 00 13- SW = 90 00
7	**Increase** 1- Reset 2- Trigger the applet 3- UICCFileView1.select() DF_{TEST}, UICCFileView1.select() EF_{CARU}, UICCFileView1.updateRecord() 00 00 00, previous UICCFileView1.updateRecord() 00 00 00, previous 4- UICCFileView1.increase(), 00 00 01 5- UICCFileView1.readRecord(), current 6- UICCFileView1.readRecord(), previous 7- UICCFileView2.select() DF_{TEST}, UICCFileView2.select() EF_{CARU}, UICCFileView2.readRecord(), current 8- UICCFileView2.increase(),00 00 01 9- UICCFileView2.readRecord(), current 10- UICCFileView2.readRecord(), previous 11- ADF1FileView1.select() DF_{TEST}, ADF1FileView1.select() EF_{CARU}, ADF1FileView1.readRecord(), current 12- ADF1FileView1.updateRecord() 00 00 00, previous ADF1FileView1.updateRecord() 00 00 00, previous 13- ADF1FileView1.increase(),00 00 02 14- ADF1FileView1.readRecord(), current 15- ADF1FileView1.readRecord(), previous	3- No exception is thrown 4- resp[] = 00 00 01 5- returns 00 00 01 6- returns 00 00 00 7- UICCException.RECORD_NOT_FOUND is thrown 8- resp[] = 00 00 02 9- returns 00 00 02 10- returns 00 00 01 11- UICCException.RECORD_NOT_FOUND is thrown 12- No exception is thrown 13- resp[] = 00 00 02 14- returns 00 00 02	

Id	Description	API Expectation	APDU Expectation
	16- ADF1FileView2.select() DF_{TEST}, ADF1FileView2.select() EF_{CARU}, ADF1FileView2.readRecord(), current 17- ADF1FileView2.increase(),00 00 02 18- ADF1FileView2.readRecord(), current 19- ADF1FileView2.readRecord(), previous 20- select DF $_{TEST}$, select EF_{CARU}, readRecord current 21- readRecord previous 22- readRecord previous 23- updateRecord previous AA AA AA 24- updateRecord previous 55 55 55 25- select AID of ADF1, select DF $_{TEST}$, select EF_{CARU}, readRecord current 26- readRecord previous 27- readRecord previous 28- updateRecord previous AA AA AA 29- updateRecord previous 55 55 55	15- returns 00 00 00 16- UICCException.RECORD_NOT_FOUND is thrown 17- resp[] = 00 00 04 18- returns 00 00 04 19- returns 00 00 02 Applet finalizes	20- SW = 6A 83 21- returns 00 00 01 22- returns 00 00 02 23- SW = 90 00 24- SW = 90 00 25- SW = 6A 83 26- returns 00 00 02 27- returns 00 00 04 28- SW = 90 00 29- SW = 90 00
8	**CreateFile EF** 1- Reset 2- Trigger the applet 3- UICCAdminFileView1.select() DF_{TEST}, UICCAdminFileView1.select() 6F 29 4- UICCAdminFileView1.create() 6F 29 5- UICCAdminFileView1.select() 6F 29 6- UICCAdminFileView2.select() DF_{TEST}, UICCAdminFileView2.select() 6F 29 7- UICCAdminFileView2.select() 6F 2A 8- UICCAdminFileView2.create() 6F 2A 9- UICCAdminFileView1.select() 6F 2A 10- UICCAdminFileView1.select() 6F 29 11- ADF1AdminFileView1.select() DF_{TEST}, ADF1AdminFileView1.select() 6F 29 12- ADF1AdminFileView1.create() 6F 29 13- ADF1AdminFileView1.select() 6F 29 14- ADF1AdminFileView2.select() DF_{TEST}, ADF1AdminFileView2.select() 6F 29 15- ADF1AdminFileView2.select() 6F 2A 16- ADF1AdminFileView2.create() 6F 2A 17- ADF1AdminFileView1.select() 6F 2A 18- ADF1AdminFileView1.select() 6F 29 21- select DF $_{TEST}$, select 6F 29 22- select 6F 2A 23- select AID of ADF1, select DF $_{TEST}$, select 6F 29	3- UICCException.FILE_NOT_FOUND is thrown 4- No exception is thrown 5- No exception is thrown 6- No exception is thrown 7- UICCException.FILE_NOT_FOUND is thrown 8- No exception is thrown 9- No exception is thrown 10- No exception is thrown 11- UICCException.FILE_NOT_FOUND is thrown 12- No exception is thrown 13- No exception is thrown 14- No exception is thrown 15- UICCException.FILE_NOT_FOUND is thrown 16- No exception is thrown 17- No exception is thrown 18- No exception is thrown Applet finalizes	21- SW = 90 00 22- SW = 90 00 23- SW = 90 00

Id	Description	API Expectation	APDU Expectation
	24- select 6F 2A		24- SW = 90 00
9	**CreateFile DF** 1- Reset 2- Trigger the applet 3- UICCAdminFileView1.select() DF_{TEST}, UICCAdminFileView1.select() 5F 01 4- UICCAdminFileView1.create() 5F 01 5- UICCAdminFileView1.select() 5F 01 6- UICCAdminFileView2.select() DF_{TEST}, UICCAdminFileView2.select() 5F 01 7- UICCAdminFileView2.select() 5F 02 8- UICCAdminFileView2.create() 5F 02 9- UICCAdminFileView1.select() 5F 02 10- UICCAdminFileView1.select() 5F 01 11- ADF1AdminFileView1.select() 5F 01 12- ADF1AdminFileView1.create() 5F 01 13- ADF1AdminFileView1.select() 5F 01 14- ADF1AdminFileView2.select() 5F 01 15- ADF1AdminFileView2.select() 5F 02 16- ADF1AdminFileView2.create() 5F 02 17- ADF1AdminFileView1.select() 5F 02 18- ADF1AdminFileView1.select() 5F 01 21- select 5F 01 22- select 5F 02 23- select AID of ADF1, select 5F 01 24- select 5F 02	3- UICCException.FILE_NOT_FOUND is thrown 4- No exception is thrown 5- No exception is thrown 6- No exception is thrown 7- UICCException.FILE_NOT_FOUND is thrown 8- No exception is thrown 9- No exception is thrown 10- No exception is thrown 11- UICCException.FILE_NOT_FOUND is thrown 12- No exception is thrown 13- No exception is thrown 14- No exception is thrown 15- UICCException.FILE_NOT_FOUND is thrown 16- No exception is thrown 17- No exception is thrown 18- No exception is thrown Applet finalizes	21- SW = 90 00 22- SW = 90 00 23- SW = 90 00 24- SW = 90 00

Id	Description	API Expectation	APDU Expectation
10	**ResizeFile** 1- Reset 2- Trigger the applet 3- UICCAdminFileView1.select() DF_{TEST}, UICCAdminFileView1.select() EF_{TDAC}, UICCAdminFileView1.readBinary(), length 06 4- UICCAdminFileView1.resize(),add 3 bytes UICCAdminFileView1.readBinary(), length 06 5- UICCAdminFileView2.select() DF_{TEST}, UICCAdminFileView2.select() EF_{TDAC}, UICCAdminFileView2.readBinary(), length 06 6- UICCAdminFileView1.resize(), remove 2 bytes 7- ADF1AdminFileView1.select() DF_{TEST}, ADF1AdminFileView1.select() EF_{LNU}, ADF1AdminFileView1.readRecord(),record 4 8- ADF1AdminFileView1.resize(),add 2 records ADF1AdminFileView1.readRecord (),record 4 9- ADF1AdminFileVie2.select() DF_{TEST}, ADF1AdminFileView2.select() EF_{LNU}, ADF1AdminFileView2.readRecord (),record 4 10- ADF1AdminFileView2.resize(), remove 1 record 11- select DF $_{TEST}$, select EF_{TDAC}, readBinary length 06 12- readBinary length 04 13- resize, remove 1 byte 14- select AID of ADF1, select DF $_{TEST}$, select EF_{LNU}, readRecord record 4 15- readRecord record 3 16- resize, remove 1 record	3- UICCException.OUT_OF_FILE_BOUNDARIES is thrown 4- returns 00 00 00 FF FF FF 5- returns 00 00 00 FF FF FF 6- No exception is thrown 7- UICCException.RECORD_NOT_FOUND is thrown 8- returns FF FF FF FF 9- returns FF FF FF FF 10- No exception is thrown Applet finalizes	11- R-APDU = 00 00 00 FF 90 00 or 62 82 or 67 00 12- returns 00 00 00 FF 13- SW = 90 00 14- SW = 6A 83 15- returns FF FF FF FF 16- SW = 90 00
11	**Non-shareable files (UICCFileView - UICCFileView)** 1- Reset 2- Trigger the applet 3- UICCFileView1.select() DF_{TEST}, UICCFileView1.select() EF_{NOSH} 4- UICCFileView2.select() DF_{TEST}, UICCFileView2.select()EF_{NOSH} 5- UICCFileView1.select() DF_{TEST} 6- UICCFileView2.select()EF_{NOSH}	3- No exception is thrown 4- UICCException.INTERNAL_ERROR is thrown or UICCException.FILE_NOT_FOUND 5- No exception is thrown 6- No exception is thrown	

Id	Description	API Expectation	APDU Expectation
12	**Non-shareable files (FileView - FileView)** 1- Reset 2- Trigger the applet 3- ADF1FileView1.select() DF_{TEST}, ADF1FileView1.select()EF_{NOSH} 4- ADF1FileView2.select() DF_{TEST}, ADF1FileView2.select()EF_{NOSH} 5- ADF1FileView1.select() DF_{TEST} 6- ADF1FileView2.select()EF_{NOSH}	3- No exception is thrown 4- UICCException.INTERNAL_ERROR is thrown or UICCException.FILE_NOT_FOUND 5- No exception is thrown 6- No exception is thrown	
13	**Non-shareable files (UICCFileView - MF)** 1- Trigger the applet 2- UICCFileView1.select() DF_{TEST}, UICCFileView1.select()EF_{NOSH} 4- Select DF_{TEST}, select EF_{NOSH} 5- Fetch and terminal response	2- No exception is thrown 3- Applet sends a display text	4- SW = 69 85
14	**Non-shareable files (FileView - ADF)** 1- Trigger the applet 2- ADF1FileView1.select() DF_{TEST}, ADF1FileView1.select()EF_{NOSH} 4- Select AID of ADF1, select DF_{TEST}, select EF_{NOSH} 5- Fetch and terminal response	2- No exception is thrown 3- Applet sends a display text	4- SW = 69 85
15	**Non-shareable files (MF - UICCFileView)** 1- Reset 2- Select DF_{TEST}, select EF_{NOSH} 3- Trigger the applet 4- UICCFileView1.select() DF_{TEST}, UICCFileView1.select()EF_{NOSH}	4- UICCException.INTERNAL_ERROR is thrown or UICCException.FILE_NOT_FOUND	2- SW = 90 00

Id	Description	API Expectation	APDU Expectation
16	**Non-shareable files (ADF - FileView)** 1- Reset 2- Select AID of ADF1, select DF_{TEST}, select EF_{NOSH} 3- Trigger the applet 4- ADF1FileView1.select() DF_{TEST}, ADF1FileView1.select()EF_{NOSH}	4- UICCException.INTERNAL_ERROR is thrown or UICCException.FILE_NOT_FOUND	2- SW = 90 00
17	**Terminated EF/DF** 1- Reset 2- Trigger the applet 3- UICCAdminFileView1.select() DF_{TEST}, UICCAdminFileView1.select()EF_{TERM} 4- UICCAdminFileView1.select() DF_{TEST}, UICCAdminFileView1.select() DF_{TERM}	3- No exception is thrown 4- No exception is thrown	

5.2 Package uicc.toolkit

5.2.1 Interface EditHandler

Tests are done in inheriting interfaces EnvelopeResponseHandler and ProactiveHandler.

5.2.2 Interface EnvelopeHandler

5.2.2.1 Method getItemIdentifier

Test Area Reference: Api_2_Enh_Giid

5.2.2.1.1 Conformance requirement:

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

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

5.2.2.1.1.1 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.

5.2.2.1.1.2 Parameter errors

No requirements.

5.2.2.1.1.3 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 Comprehension TLV.

5.2.2.1.2 Test area files

Test Source: Test_Api_2_Enh_Giid.java.

Test Applet: Api_2_Enh_Giid_1.java.

Cap File: api_2_enh_giid.cap.

5.2.2.1.3 Test coverage

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

5.2.2.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	Send envelope Menu Selection with item identifier TLV and identifier value of 03	Returns 03	
2	Send envelope Menu Selection with two item identifier TLV with first value 02 and second 01	Returns 02	
3	Send envelope Menu Selection with two item identifier TLV with first value 04 and second 01, call twice the method getItemIdentifier()	Returns 04 Returns 04	
4	Send envelope Menu Selection with item identifier TLV and value of 66. FindTLV() with TAG 02. getItemIdentifier() and then getValueByte() with offset 0	getItemIdentifier()=getValueByte()	
5	Send unrecognized envelope without item identifier TLV and getItemIdentifier()	ToolkitException.UNAVAILABLE_ELEMENT	
6	Send Envelope Menu Selection with item identifier TLV (66), send proactive command. Then getItemIdentifier()	Returns 66	
7	Send Unrecognized Envelope with item identifier TLV but without item number	ToolkitException.OUT_OF_TLV_BOUNDARIES	

5.2.2.2 Method getLength

Test Area Reference: Api_2_Enh_Glen.

5.2.2.2.1 Conformance requirement

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

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

5.2.2.2.1.1 Normal execution

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

5.2.2.2.1.2 Parameter errors

No requirements.

5.2.2.2.1.3 Context errors

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

5.2.2.2.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_2_Enh_Glen.java.

Test Applet: Api_2_Enh_Glen_1.java.

Cap File: api_2_enh_glen.cap.

5.2.2.2.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4
C1	Not testable

5.2.2.2.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	Send an Unrecognized Envelope with BER length of 0x31	Result of getLength() is 0x0031	
2	Send an Unrecognized Envelope with BER length of 0x7F	Result of getLength() is 0x007Fh	
3	Send an Unrecognized Envelope with BER length of 81 80	Result of getLength() is 0x0080h	
4	Send an Unrecognized Envelope with BER length of 81 FC	Result of getLength() is 0x00FCh	

5.2.2.3 Method copy

Test Area Reference: Api_2_Enh_Copy.

5.2.2.3.1 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
```

5.2.2.3.1.1 Normal execution

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

5.2.2.3.1.2 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 Comprehension TLV List, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.OUT_OF_TLV_BOUNDARIES.

5.2.2.3.1.3 Context errors

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

5.2.2.3.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Enh_Copy.java.

Test Applet: Api_2_Enh_Copy_1.java.

Cap File: api_2_enh_copy.cap.

5.2.2.3.4 Test coverage

CRR number	Test case number
N1	9, 11, 13, 15
N2	8, 10, 12, 14, 16
P1	1
P2	2, 3, 4, 5, 6
P3	7
C1	Not testable

5.2.2.3.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**NULL as parameter to dstBuffer**	NullPointerException is thrown	
2	**dstOffset ≥ dstBuffer.length** copy() dstBuffer.length = 5 dstOffset = 5 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	
3	**dstOffset < 0** copy() dstBuffer.length = 5 dstOffset = -1 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**dstLength > dstBuffer.length** copy() dstBuffer.length = 5 dstOffset = 0 dstLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**DstOffset + dstLength > dstBuffer.length** copy() DstBuffer.length = 5 DstOffset = 3 DstLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength < 0** copy() dstBuffer.length = 5 dstOffset = 0 dstLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**DstLength > length of the Comprehension TLV list** copy() DstBuffer.length = 48 DstOffset = 0 DstLength = 48	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**Successful call, dstBuffer is the whole buffer** copy() 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** copy() 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** copy() dstBuffer.length = 252 dstOffset = 3 dstLength = 252	Result of copy() is 0X00FF	
13	**Compare the whole buffer**	Result of arrayCompare() is 0	
14	**Successful call, dstBuffer is part of a buffer** copy() dstBuffer.length = 260 dstOffset = 257 dstLength = 3	Result of copy() is 0X0104	
15	**Compare the whole buffer**	Result of arrayCompare() is 0	

Id	Description	API Expectation	APDU Expectation
16	**Successful call, copy() with length =0** dstBuffer.length = 260 dstOffset = 260 dstLength = 0	Result of copy() is 0x104	

5.2.2.4 Method findTLV

Test Area Reference: Api_2_Enh_Find.

5.2.2.4.1 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
```

5.2.2.4.1.1 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.

5.2.2.4.1.2 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.

5.2.2.4.1.3 Context errors

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

5.2.2.4.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Enh_Find.java.

Test Applet: Api_2_Enh_Find_1.java.

Cap File: api_2_enh_find.cap.

5.2.2.4.3 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	Not testable

5.2.2.4.4 Test procedure

Id	Description	API Expectation	APDU Expectation
	Trigger the applet with Unrecognized Envelope including: Tag 82, tag 86, tag 8B, tag 02 and tag 04		

1	**Invalid input parameter**
	`findTLV()`
	`Occurrence = 0`
		ToolkitException.BAD_INPUT_PARAMETER is thrown	

2	**Search 1st TLV**
	`findTLV()`
	`Tag = 02h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_SET	

3	**Call the getValueLength() method**	Result is 0x02	

4	**Search 2nd TLV**
	`findTLV()`
	`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
	`findTLV()`
	`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**
	`findTLV()`
	`Tag = 02h`
	`Occurrence = 3`
		Result is TLV_NOT_FOUND	

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

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

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

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

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

5.2.2.5 Method getValueLength

Test Area Reference: Api_2_Enh_Gvle.

5.2.2.5.1 Conformance requirement

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

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

5.2.2.5.1.1 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.

5.2.2.5.1.2 Parameter errors

No requirements.

5.2.2.5.1.3 Context errors

- CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.HANDLER_NOT_AVAILABLE.
- CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.UNAVAILABLE_ELEMENT.

5.2.2.5.2 Test area files

Specific triggering: Unrecognized Envelope:

Test source: Test_Api_2_Enh_Gvle.java.

Test Applet: Api_2_Enh_Gvle_1.java.

Cap File: api_2_enh_gvle.cap.

5.2.2.5.3 Test coverage

CRR number	Test case number
N1	2, 3, 4
C1	Not testable
C2	1

5.2.2.5.4 Test procedure

Id	Description	API Expectation	APDU Expectation
	Fill the Unrecognized envelope with TLV: Tag 02, length 02, Tag 06, length 05, Tag 0B, length 24, 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	

5.2.2.6 Method getValueByte

Test Area Reference: Api_2_Enh_Gvby.

5.2.2.6.1 Conformance requirement

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

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

5.2.2.6.1.1 Normal execution

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

5.2.2.6.1.2 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.

5.2.2.6.1.3 Context errors

- CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.HANDLER_NOT_AVAILABLE.
- CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.UNAVAILABLE_ELEMENT.

5.2.2.6.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Enh_Gvby.java.

Test Applet: Api_2_Enh_Gvby_1.java.

Cap File: api_2_enh_gvby.cap.

5.2.2.6.3 Test coverage

CRR number	Test case number
N1	3, 4, 5, 6, 7, 8
P1	2
C1	Not testable
C2	1

5.2.2.6.4 Test procedure

Id	Description	API Expectation	APDU Expectation
	Fill the Unrecognized envelope with TLV: Tag 02, length 02, value 83 81, Tag 06, length 06, Tag 0B, length 21, 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)	

Id	Description	API Expectation	APDU Expectation
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	

5.2.2.7 Method copyValue

Test Area Reference: Api_2_Enh_Cpyv.

5.2.2.7.1 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
```

5.2.2.7.1.1 Normal execution

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

5.2.2.7.1.2 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.

5.2.2.7.1.3 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.

5.2.2.7.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Enh_Cpyv.java.

Test Applet: Api_2_Enh_Cpyv_1.java.

Cap File: api_2_enh_cpyv.cap.

5.2.2.7.3 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	Not testable
C2	11

5.2.2.7.4 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
	`copyValue()`
	`dstBuffer.length = 5`
	`dstOffset = 5`
	`dstLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

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

4	**dstLength >dstBuffer.length**
	`copyValue()`
	`dstBuffer.length = 5`
	`dstOffset = 0`
	`dstLength = 6`
		ArrayIndexOutOfBoundsException is thrown	

5	**dstOffset + dstLength >dstBuffer.length**
	`copyValue()`
	`dstBuffer.length = 5`
	`dstOffset = 3`
	`dstLength = 3`
		ArrayIndexOutOfBoundsException is thrown	

6	**dstLength < 0**
	`copyValue()`
	`dstBuffer.length = 5`
	`dstOffset = 0`
	`dstLength = -1`
		ArrayIndexOutOfBoundsException is thrown	

7	**Search TLV 06h**		

	valueOffset ≥ TLV Length
	`copyValue()`
	`valueOffset = 6`
	`dstBuffer.length = 15`
	`dstOffset = 0`
	`dstLength = 1`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

8	**valueOffset < 0**
	`copyValue()`
	`valueOffset = -1`
	`dstBuffer.length = 15`
	`dstOffset = 0`
	`dstLength = 1`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

9	**dstLength > TLV length**
	`copyValue()`
	`valueOffset = 0`
	`dstBuffer.length = 15`
	`dstOffset = 0`
	`dstLength = 7`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

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

Id	Description	API Expectation	APDU Expectation
11	**Search TLV 01h**		
	copyValue()	ToolkitException.UNAVAILABLE_ELEMENT is thrown on the copyValue() method call.	
12	**Search TLV 06h**		
	Successful call copyValue() 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	**initialize dstBuffer** dstBuffer = 55 55 … 55		
	Successful call copyValue() 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	

5.2.2.8 Method compareValue

Test Area Reference: Api_2_Enh_Cprv.

5.2.2.8.1 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
```

5.2.2.8.1.1 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 Comprehension TLV List is less than that in compareBuffer.
- CRRN3: returns 1 if the first miscomparing byte in Comprehension TLV List is greater than that in compareBuffer.

5.2.2.8.1.2 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.

5.2.2.8.1.3 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.

5.2.2.8.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Enh_Cprv.java.

Test Applet: Api_2_Enh_Cprv_1.java.

Cap File: api_2_enh_cprv.cap.

5.2.2.8.3 Test coverage

CRR number	Test case number
N1	12, 15, 18
N2	13, 14, 17
N3	16
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Not testable
C2	11

5.2.2.8.4 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
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 5`
	`compareLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

3	**compareOffset < 0**
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = -1`
	`compareLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

4	**compareLength >compareBuffer.length**
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 0`
	`compareLength = 6`
		ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
5	**compareOffset + compareLength >compareBuffer.length** compareValue() compareBuffer.length = 5 compareOffset = 3 compareLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**compareLength < 0** compareValue() compareBuffer.length = 5 compareOffset = 0 compareLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Search TLV 06h**		
	valueOffset ≥ TLV Length compareValue() valueOffset = 6 compareBuffer.length = 15 compareOffset = 0 compareLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**valueOffset < 0** compareValue() valueOffset = -1 compareBuffer.length = 15 compareOffset = 0 compareLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
9	**compareLength > TLV length** compareValue() valueOffset = 0 compareBuffer.length = 15 compareOffset = 0 compareLength = 7	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
10	**valueOffset + compareLength > TLV length** compareValue() 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**		
	Initialize compareBuffer compareBuffer = 81 11 22 33 44 F5		
	Compare buffers compareValue() valueOffset = 0 compareOffset = 0 compareLength = 6	Result is 00h	
13	**Initialize compareBuffer** compareBuffer = 7F 11 22 33 44 F5		
	Compare buffers with same parameters	Result is -1	
14	**Initialize compareBuffer** compareBuffer = 83 11 22 33 44 F5		
	Compare buffers with same parameters	Result is -1	
15	**Initialize compareBuffer** compareBuffer = 55 55 55 81 11 22 33 44 F5 55 55 55 55 55		
	Compare buffers compareValue() valueOffset = 1 compareOffset = 4 compareLength = 5	Result is 00h	
16	**Initialize compareBuffer** compareBuffer = 55 55 55 81 10 23 33 44 F5 55 55 55 55 55		
	Compare buffers with same parameters	Result is +1	

Id	Description	API Expectation	APDU Expectation
17	**Initialize compareBuffer** compareBuffer = 55 55 55 81 12 21 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	

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

Test Area Reference: Api_2_Enh_Facyb_Bs.

5.2.2.9.1 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
```

5.2.2.9.1.1 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.

5.2.2.9.1.2 Parameter errors

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

5.2.2.9.1.3 Context errors

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

5.2.2.9.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Enh_Facyb_Bs.java.

Test Applet: Api_2_Enh_Facyb_Bs_1.java.

Cap File: api_2_enh_facyb_bs.cap.

5.2.2.9.3 Test coverage

CRR number	Test case number
N1	9, 11, 13
N2	6, 7
N3	8, 10, 12
N4	14, 15, 16, 17
P1	1
P2	2, 3, 4, 5
C1	Not testable

5.2.2.9.4 Test procedure

Id	Description	API Expectation	APDU Expectation
	Fill the Unrecognized Envelope with TLV: Tag 02, Value 83 81, Tag 06, Value 81 11 22 33 44 F5, 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** findAndCopyValue() tag = 06h dstBuffer.length = 06 dstOffset = 06	ArrayIndexOutOfBoundsException is thrown	
3	**dstOffset < 0** findAndCopyValue() dstBuffer.length = 06 dstOffset = -1	ArrayIndexOutOfBoundsException is thrown	
4	**length > dstBuffer.length** findAndCopyValue() dstBuffer.length = 05 dstOffset = 0	ArrayIndexOutOfBoundsException is thrown	
5	**DstOffset + length >dstBuffer.length** findAndCopyValue() 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** findAndCopyValue() 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	**Initialize dstBuffer** dstBuffer = 55 55 … 55		
	Successful call findAndCopyValue() 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** findAndCopyValue() tag = 02h dstBuffer.length = 2 dstOffset = 0	Result of findAndCopyValue () is 0002	
13	**Compare buffer** buffer = 83 81	Result is 00h	

Id	Description	API Expectation	APDU Expectation

14	**Successful call (with tag 82h)**
	`findAndCopyValue()`
	`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)**
	`findAndCopyValue()`
	`tag = B3h`
	`dstBuffer.length = C4`
	`dstOffset = 0`
		Result of findAndCopyValue () is 00C4	

17	**Compare buffer** `buffer = 01 02 … C4`	Result is 00h	

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

Test Area Reference: Api_2_Enh_Facybs_Bss.

5.2.2.10.1 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
```

5.2.2.10.1.1 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.

5.2.2.10.1.2 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.
- 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`.

5.2.2.10.1.3 Context errors

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

5.2.2.10.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Enh_Facybs_Bss.java.

Test Applet: Api_2_Enh_Facybs_Bss_1.java.

Cap File: api_2_enh_facybs_bss.cap.

5.2.2.10.3 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
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
P4	26
C1	Not testable

5.2.2.10.4 Test procedure

Id	Description	API Expectation	APDU Expectation
	Fill the Unrecognized Envelope with TLV: Tag 02, Value 83 81, Tag 06, Value 81 11 22 33 44 F5, 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** findAndCopyValue() tag = 06h, occurrence = 1 valueOffset = 0 dstBuffer.length = 5 dstOffset = 5 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	
3	**dstOffset < 0** findAndCopyValue() dstBuffer.length = 5 dstOffset = -1 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**dstLength >dstBuffer.length** findAndCopyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**dstOffset + dstLength >dstBuffer.length** findAndCopyValue() dstBuffer.length = 5 dstOffset = 3 dstLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength < 0** findAndCopyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**valueOffset ≥ Value Length** findAndCopyValue() tag = 06h, occurrence = 1 valueOffset = 6 dstBuffer.length = 15 dstOffset = 0 dstLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

Id	Description	API Expectation	APDU Expectation
8	**valueOffset < 0** findAndCopyValue() valueOffset = -1 dstBuffer.length = 15 dstOffset = 0 dstLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
9	**dstLength > Value length** findAndCopyValue() valueOffset = 0 dstBuffer.length = 15 dstOffset = 0 dstLength = 7	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
10	**valueOffset + dstLength > Text String length** findAndCopyValue() 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** findAndCopyValue() 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	**Initialize dstBuffer** dstBuffer = 55 55 … 55		
	Successful call findAndCopyValue() 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** findAndCopyValue() 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** findAndCopyValue() 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)** findAndCopyValue() 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	

Id	Description	API Expectation	APDU Expectation
23	**Successful call (with tag 82h)** findAndCopyValue() 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	
26	**Invalid parameter** findAndCopyValue() occurrence = 0	ToolkitException.BAD_INPUT_PARAMETER is thrown	

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

Test Area Reference: Api_2_Enh_Facrb_Bs.

5.2.2.11.1 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
```

5.2.2.11.1.1 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 Comprehension TLV is less than that in compareBuffer returns -1.
- CRRN5: if the first miscomparing byte in Comprehension TLV is greater than that in compareBuffer returns 1.
- CRRN6: The search method is comprehension required flag independent.

5.2.2.11.1.2 Parameter errors

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

5.2.2.11.1.3 Context errors

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

5.2.2.11.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Enh_Facrb_Bs.java.

Test Applet: Api_2_Enh_Facrb_Bs_1.java.

Cap File: api_2_enh_facrb_bs.cap.

5.2.2.11.3 Test coverage

CRR number	Test case number
N1	6,7
N2	9
N3	8, 12, 13
N4	11, 15
N5	10, 14
N6	16, 17
P1	1
P2	2, 3, 4, 5
C1	Not testable

5.2.2.11.4 Test procedure

Id	Description	API Expectation	APDU Expectation
	Fill the Unrecognized Envelope with TLV: Tag 02, Value 83 81, Tag 06, Value 81 11 22 33 44 F5, 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**
	`findAndCompareValue()`
	`tag = 06h`
	`compareBuffer.length = 12`
	`compareOffset = 12`
		ArrayIndexOutOfBoundsException is thrown	

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

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

5	**compareOffset + length > compareBuffer.length**
	`findAndCompareValue()`
	`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	**Initialize compareBuffer** `compareBuffer = 81 11 22 33 44 F5`		

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

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

Id	Description	API Expectation	APDU Expectation
11	**Initialize compareBuffer** compareBuffer = 81 11 22 33 44 F6		
	Compare buffers with same parameters	Result is -1	
12	**Initialize compareBuffer** compareBuffer = 55 55 81 11 22 33 44 F5 55 55 55 55		
	Compare buffers findAndCompareValue() compareOffset = 2	Result is 00h	
13	**Initialize compareBuffer** compareBuffer = 55 55 83 81 55 55 55 55 55 55 55 55		
	Compare buffers findAndCompareValue() compareOffset = 2	Result is 00h	
14	**Initialize compareBuffer** compareBuffer = 55 55 83 80 55 55 55 55 55 55 55 55		
	Compare buffers findAndCompareValue() compareOffset = 2	Result is +1	
15	**Initialize compareBuffer** compareBuffer = 55 55 83 82 55 55 55 55 55 55 55 55		
	Compare buffers findAndCompareValue() compareOffset = 2	Result is -1	
16	**Initialize compareBuffer** compareBuffer = 83 81 55 55 55 55 55 55 55 55 55 55		
	Successful call (with tag 02h) findAndCompareValue() tag = 02h compareBuffer.length = 12 compareOffset = 0	Result is 00h	
17	**Initialize compareBuffer** CompareBuffer = 01 02 … C4		
	Successful call (with tag B3h) findAndCompareValue() Tag = B3h CompareBuffer.length = C4 CompareOffset = 0	Result is 00h	

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

Test Area Reference: Api_2_Enh_Facrbbs_Bss.

5.2.2.12.1 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
```

5.2.2.12.1.1 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 Comprehension TLV is less than that in compareBuffer -1 is returned.
- CRRN5: if the first miscomparing byte in Comprehension TLV is greater than that in compareBuffer 1 is returned
- CRRN6: The search method is comprehension required flag independent.

5.2.2.12.1.2 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.

5.2.2.12.1.3 Context errors

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

5.2.2.12.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Enh_Facrbbs_Bss.java.

Test Applet: Api_2_Enh_Facrbbs_Bss_1.java.

Cap File: api_2_enh_facrbbs_bss.cap.

5.2.2.12.3 Test coverage

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

5.2.2.12.4 Test procedure

Id	Description	API Expectation	APDU Expectation
	Fill the Unrecognized Envelope with TLV: Tag 02, Value 83 81, Tag 06, Value 81 11 22 33 44 F5, 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** findAndCompareValue() tag = 06h, occurrence = 1 valueOffset = 0 compareBuffer.length = 6 compareOffset = 6 compareLength = 1	ArrayIndexOutOfBoundsException is thrown	
3	**compareOffset < 0** findAndCompareValue() compareBuffer.length = 6 compareOffset = -1 compareLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**compareLength >compareBuffer.length** findAndCompareValue() compareBuffer.length = 5 compareOffset = 0 compareLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**compareOffset + compareLength >compareBuffer.length** findAndCompareValue() compareBuffer.length = 5 compareOffset = 3 compareLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**compareLength < 0** findAndCompareValue() compareBuffer.length = 5 compareOffset = 0 compareLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**valueOffset ≥ Value Length** findAndCompareValue() tag = 06h, occurrence = 1 valueOffset = 6 compareBuffer.length = 15 compareOffset = 0 compareLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**valueOffset < 0** findAndCompareValue() valueOffset = -1 compareBuffer.length = 15 compareOffset = 0 compareLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
9	**compareLength > Value length** findAndCompareValue() valueOffset = 0 compareBuffer.length = 15 compareOffset = 0 compareLength = 7	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
10	**valueOffset + compareLength > Value length** findAndCompareValue() valueOffset = 2 compareBuffer.length = 15 compareOffset = 0 compareLength = 5	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
11	**Invalid parameter** findAndCompareValue() 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.	

Id	Description	API Expectation	APDU Expectation
14	**Initialize 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	**Initialize compareBuffer** compareBuffer = 81 11 22 33 44 F4		
	Compare buffers with same parameters	Result is +1	
17	**Initialize compareBuffer** compareBuffer = 81 11 22 33 44 F6		
	Compare buffers with same parameters	Result is -1	
18	**Initialize compareBuffer** compareBuffer = 55 55 55 22 33 44 F5 55 55 55 55		
	Compare buffers findAndCompareValue() valueOffset = 2 compareOffset = 3 compareLength = 4	Result is 00h	
19	**Initialize compareBuffer** compareBuffer = 55 55 55 22 33 45 F5 55 55 55 55		
	Compare buffers with same parameters	Result is -1	
20	**Initialize compareBuffer** compareBuffer = 55 55 55 22 33 43 F5 55 55 55 55		
	Compare buffers with same parameters	Result is +1	
21	**Initialize 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	**Initialize 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	**Initialize 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	**Initialize compareBuffer** compareBuffer = 83 81 55 55 55 55 55 55 55 55 55 55		
	Successful call (with tag 02h) findAndCompareValue() tag = 02h, occurrence = 1 valueOffset = 0 compareBuffer.length = 12 compareOffset = 0 compareLength = 2	Result is 00h	

Id	Description	API Expectation	APDU Expectation
25	**Initialize compareBuffer** `compareBuffer = 01 02 … C4`		

	Successful call (with tag B3h)
	`findAndCompareValue()`
	`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	

5.2.2.13 Method getCapacity

Test Area Reference: Api_2_Enh_Gcap.

5.2.2.13.1 Conformance requirement

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

```
public byte getCapacity()
```

5.2.2.13.1.1 Normal execution

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

5.2.2.13.1.2 Parameter errors

No requirements

5.2.2.13.1.3 Context errors

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

5.2.2.13.2 Test area files

Test Source: Test_Api_2_Enh_Gcap.java.

Test Applet: Api_2_Enh_Gcap_1.java.

Cap File: api_2_enh_gcap.cap.

5.2.2.13.3 Test coverage

CRR number	Test case number
N1	1
C1	Not testable

5.2.2.13.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**EnvelopeHandler available** 1 - Send envelope Menu Selection 2 - The applet calls the getLength() method 3 - The applet calls the 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	

5.2.2.14 Method getChannelIdentifier

Test Area Reference: Api_2_Enh_Gcid.

5.2.2.14.1 Conformance requirement

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

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

5.2.2.14.1.1 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.

5.2.2.14.1.2 Parameter errors

No requirements

5.2.2.14.1.3 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 Comprehension TLV Channel Status length is equal to 0.

5.2.2.14.2 Test area files

Test Source: Test_Api_2_Enh_Gcid.java.

Test Applet: Api_2_Enh_Gcid_1.java.

Cap File: api_2_enh_gcid.cap.

5.2.2.14.3 Test coverage

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

5.2.2.14.4 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() method is called.		2- OPEN CHANNEL proactive command is fetched **TERMINAL RESPONSE is issued with Channel Id from 01 to 07**
1	**Successful Call** 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	**Two channel status elements** 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 twice 0x04	
3	**Verify current TLV** 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	**UNAVAILABLE_ELEMENT exception** 1- Send envelope Menu Selection without Channel Status TLV. 2- Call EnvelopeHandler.getChannelIdentifier() method.	2- A Toolkit exception.UNAVAILABLE_ELEMENT is thrown.	
5	**Successful Call** 1- Send Envelope Event Download Channel Status with Channel Status TLV: Channel status value = 0x0600 2- Call EnvelopeHandler.getChannelIdentifier() method.	1- Returns 0x06	

Id	Description	API Expectation	APDU Expectation
6	**OUT_OF_TLV_BOUNDARIES exception** 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.	

5.2.2.15 Method getChannelStatus

Test Area Reference: Api_2_Enh_Gcst.

5.2.2.15.1 Conformance requirement

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

```
public short getChannelStatus(byte channelIdentifier)
                      throws ToolkitException
```

5.2.2.15.1.1 Normal execution

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

5.2.2.15.1.2 Parameter errors

No requirements.

5.2.2.15.1.3 Context errors

- CRRC1: The method shall throw ToolkitException.UNAVAILABLE_ELEMENT if no Channel Status TLV element with the right identifier could be found.
- CRRC2: The method shall throw ToolkitException.OUT_OF_TLV_BOUNDARIES if a Channel Status TLV element with the right identifier could be found but its value is less than 2 bytes long.

5.2.2.15.2 Test area files

Test Source: Test_Api_2_Enh_Gcst.java.

Test Applet: Api_2_Enh_Gcst_1.java.

Cap File: api_2_enh_gcst.cap.

5.2.2.15.3 Test coverage

CRR number	Test case number
N1	6
N2	5
N3	7
N4	8
C1	1, 2
C2	3, 4

5.2.2.15.4 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 a channel. ProactiveHandler.send() method is called.		2- OPEN CHANNEL proactive command is fetched TERMINAL RESPONSE is issued with channel status value = 0x8100
1	**Channel status TLV is not present** 1- Send envelope Event Download Channel Status with no Channel status TLV 2- Call EnvelopeHandler.getChannelStatus(0x01) method.	2- UNAVAILABLE_ELEMENT ToolkitException is thrown	
2	**Channel status TLV with the identifier is not present** 1- Send envelope Event Download Channel Status with Channel status Value = 0x8200 2- Call EnvelopeHandler.getChannelStatus(0x01) method.	2- UNAVAILABLE_ELEMENT ToolkitException is thrown	
3	**Channel status TLV with a length equal to 0** 1- Send envelope Event Download Channel Status with Channel status length equal to 0. 2- Call EnvelopeHandler.getChannelStatus(0x01) method.	2- UNVAILABLE_ELEMENT ToolkitException is thrown	
4	**Channel status TLV with a length equal to 1** 1- Send envelope Event Download Channel Status with Channel status length equal to 1. 2- Call EnvelopeHandler.getChannelStatus(0x01) method.	2- OUT_OF_TLV_BOUNDARIES ToolkitException is thrown	
5	**Get channel status value** 1- Send envelope Event Download Channel Status with Channel status value=0x8100. 2- Call EnvelopeHandler.getChannelStatus(0x01) method.	2- Returns 0x8100	
6	**Get channel status value with 2 TLV** 1- Send envelope Event Download Channel Status with 2 channel status value: 0x8100 and 0x8101. 2- Call EnvelopeHandler.getChannelStatus(0x01) method.	2- Returns 0x8100	

Id	Description	API Expectation	APDU Expectation
7	**Channel status TLV is currently selected TLV** 1- Send envelope Event Download Channel Status with channel status value 0x8100. Call ViewHandler.FindTLV() method with Device Identity Tag. 2- Call EnvelopeHandler.getChannelStatus(0x01) method. 3- Compare EnvelopeHandler.getChannelStatus(0x01) and ViewHandler.getValueShort(0) method results.	2- Returns 0x8100 3- Check getChannelStatus() =getValueShort(0)	
8	**Get channel status value after a proactive command** 1- Send envelope Event Download Channel Status with Channel status value=0x8100. 2- Call EnvelopeHandler.getChannelStatus(0x01) method. 3- Send a proactive command display text 4- Call EnvelopeHandler.getChannelStatus(0x01) method.	2- Returns 0x8100 4- Returns 0x8100	3- DISPLAY TEXT proactive command is fetched TERMINAL RESPONSE is issued

5.2.2.16 Method getValueShort

Test Area Reference: Api_2_Enh_Gvsh.

5.2.2.16.1 Conformance requirement

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

```
public short getValueShort(short valueOffset)
                    throws ToolkitException
```

5.2.2.16.1.1 Normal execution

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

5.2.2.16.1.2 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.

5.2.2.16.1.3 Context errors

- CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.HANDLER_NOT_AVAILABLE.
- CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.UNAVAILABLE_ELEMENT.

5.2.2.16.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Enh_Gvsh.java.

Test Applet: Api_2_Enh_Gvsh_1.java.

Cap File: api_2_enh_gvsh.cap.

5.2.2.16.3 Test coverage

CRR number	Test case number
N1	3, 4, 5, 6, 7, 8
P1	2
C1	Not testable
C2	1

5.2.2.16.4 Test procedure

Id	Description	API Expectation	APDU Expectation
	Fill the Unrecognized envelope with TLVs: Tag 02, Length 02 Value 83 81 Tag 06, Length 06 Value 81 11 22 33 44 F5 Tag 33, Length C9 Value 01 02 …		
1	getValueShort(0)	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	Search TLV 02h		
	getValueShort(2)	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
3	Search TLV 02h		
	getValueShort(0)	Result is 0x83 0x81	
4	Search TLV 06h		
	getValueShort(1)	Result is 0x11 0x22	
5	Search TLV 33h		
	getValueShort(7E)	Result is 0x7F 0x80	
6	Search TLV 33h		
	getValueShort(80)	Result is 0x81 0x82	
7	getValueShort(7F)	Result is 0x80 0x81	
8	Search TLV B3h		
	getValueShort(C7)	Result is 0xC8 0xC9	

5.2.2.17 Method getSize

Test Area Reference: Api_2_Enh_Gtsz.

5.2.2.17.1 Conformance requirement

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

```
public short getSize()
```

5.2.2.17.1.1 Normal execution

- CRRN1: Returns the BER TLV size, this includes the tag and the length.

5.2.2.17.1.2 Parameter errors

No requirements.

5.2.2.17.1.3 Context errors

No requirements.

5.2.2.17.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Enh_Gtsz.java.

Test Applet: Api_2_Enh_Gtsz_1.java.

Cap File: api_2_enh_gtsz.cap.

5.2.2.17.3 Test coverage

CRR number	Test case number
1	1, 2

5.2.2.17.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	`Send an unrecognized envelope of length 0x33 (including tag and length)`		
1	`Call getSize() method just after triggering of the application.`	Returns 0x33	
2	`Call getSize() method after a proactive command.`	Returns 0x33	

5.2.2.18 Method getTag

Test Area Reference: Api_2_Enh_Gttg.

5.2.2.18.1 Conformance requirement

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

```
public short getTag()
```

5.2.2.18.1.1 Normal execution

- CRRN1: Returns the BER Tag of the BER TLV list.

5.2.2.18.1.2 Parameter errors

No requirements.

5.2.2.18.1.3 Context errors

No requirements.

5.2.2.18.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Enh_Gttg.java.

Test Applet: Api_2_Enh_Gttg_1.java.

Cap File: api_2_enh_gttg.cap.

5.2.2.18.3 Test coverage

CRR number	Test case number
1	1, 2

5.2.2.18.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Send an unrecognized envelope		
1	Call getTag() method just after triggering of the application.	Returns 0xXX	
2	Call getTag() method after a proactive command.	Returns 0xXX	

5.2.3 Interface EnvelopeResponseHandler

5.2.3.1 Method post

Test Area Reference: Api_2_Erh_Post.

5.2.3.1.1 Conformance requirement

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

```
public void post(boolean value)
          throws ToolkitException
```

5.2.3.1.1.1 Normal execution

- CRRN1: The toolkit applet can continue its processing after the call to post() method.
- CRRN2: In case of CALL_CONTROL, the CAT Runtime Environment has to set the boolean value always to true.
- CRRN3: The CAT Runtime Environment shall send the response before the emission of the next proactive command or when all the Toolkit Applets triggered by the event have finished their processing.
- CRRN4: The CAT Runtime Environment has to map the boolean value to the correct status word. If the value is true it corresponds to a successful ending of the command status word "9000". If the value is false it corresponds to a warning status word "6200".

5.2.3.1.1.2 Parameter errors

No requirements.

5.2.3.1.1.3 Context errors

- CRRC1: The method shall throw ToolkitException.HANDLER_NOT_AVAILABLE if the handler is busy.
- CRRC2: The method shall throw ToolkitException.BAD_LENGTH if the resulting response length is greater than 256 and the response data has to be retrieved by the GET RESPONSE command.

5.2.3.1.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_2_Erh_Post.java.

Test Applet: Api_2_Erh_Post_1.java.

Cap File: api_2_erh_post.cap.

5.2.3.1.3 Test coverage

CRR number	Test case number
N1	3, 4, 7
N2	8, 9
N3	4, 5
N4	1, 2, 4, 7
C1	3, 6, 7
C2	10 (see note)
NOTE: This test is conditional and automatically performed if the capacity of the envelopeResponseHandler is greater than 256 bytes.	

5.2.3.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
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 value TRUE		FD data with status word 90 00 are returned
3	**Verify that after a post the handler is no more available** appendTLV(), then post() and then appendTLV()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown on the second appendTLV	
4	**construct the response (appendTLV() with 0x10 data) and post it with value FALSE and then send a display text** 1- Send an unrecognizedEnvelope() to trigger the applet. 2- Send a envelopeEventDownloadUserActivity() to get the 91 XX status word.		1- 12 data with status SW = 62 00 2- SW = 91 15 is returned and display text is 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 1- Send an unrecognizedEnvelope() to trigger the applet. 2- Send a envelopeEventDownloadUserActivity() to get the 91 XX status word.		1- SW = 62 00 2- SW = 91 15 is returned 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 and then post().	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	
7	**Verify that the handler is no more available after a post()** getTheHandler(), appendTLV(), then post() with value FALSE and then post() with value TRUE	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	12 data with status word 62 00 are returned
8	**CALL_CONTROL, the CAT Runtime Environment set the boolean value always to true** Trigger the applet with CALL CONTROL getTheHandler(), appendTLV(), post() with value FALSE		12 data with status word 90 00 are returned
9	**CALL_CONTROL, the CAT Runtime Environment set the boolean value always to true** Trigger the applet with CALL CONTROL getTheHandler(), appendTLV(), post() with value TRUE		12 data with status word 90 00 are returned
10	**Resulting response length greater than 256** getTheHandler(), appendTLV()(data length 252 bytes), appendTLV()(data length 1 byte), post() with value TRUE	javacard.framework.APDUException.BAD_LENGTH is thrown	

5.2.3.2 Method postAsBERTLV

Test Area Reference:Api_2_Erh_Poab.

5.2.3.2.1 Conformance requirement

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

```
public void postAsBERTLV( boolean value,
                          byte tag)
                  throws ToolkitException
```

5.2.3.2.1.1 Normal execution

- CRRN1: The toolkit applet can continue its processing after the call to postAsBERTLV() method.
- CRRN2: In case of CALL_CONTROL, the CAT Runtime Environment has to set the boolean value always to true.
- CRRN3: The CAT Runtime Environment shall send the response before the emission of the next proactive command or when all the Toolkit Applets triggered by the event have finished their processing.
- CRRN4: The byte tag is the BER Tag at the beginning of the Comprehension TLV list.
- CRRN5: The CAT Runtime Environment has to map the Boolean value to the correct status word. If the value is true it corresponds to a successful ending of the command status word "9000". If the value is false it corresponds to a warning status word "6200".

5.2.3.2.1.2 Parameter errors

No requirements.

5.2.3.2.1.3 Context errors

- CRRC1: The method shall thrown ToolkitException.HANDLER_NOT_AVAILABLE if the handler is busy.
- CRRC2: The method shall throw ToolkitException.BAD_LENGTH if the resulting response length is greater than 256 and the response data has to be retrieved by the GET RESPONSE command.

5.2.3.2.2 Test area files

Specific triggering: Unrecognized envelope:

Test Source: Test_Api_2_Erh_Poab.java.

Test Applet: Api_2_Erh_Poab_1.java.

Cap File: api_2_erh_poab.cap.

5.2.3.2.3 Test coverage

CRR number	Test case number
N1	3, 4, 7
N2	8, 9
N3	4, 5
N4	2, 4, 7
N5	1, 2, 4, 7
C1	3, 6, 7
C2	10

5.2.3.2.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**getTheHandler() and then postAsBERTLV()** (the handler is empty)		02 data with status word 90 00 are returned, the tag shall be 33 and the length is 00
2	**Fill the handler and then postAsBERTLV() the data with value TRUE, and tag 33**		FF data with status word 90 00 are returned, the tag shall be 33
3	**appendTLV(), postAsBERTLV() and then appendTLV()**	ToolkitException.HANDLER_NOT_AVAILABLE is thrown on the second appendTLV	
4	**construct the response (appendTLV() with 0x10 data) and postAsBERTLV() with value FALSE, tag 75 and then send a display text** 1- Send an unrecognizedEnvelope() to trigger the applet. 2- Send a envelopeEventDownloadUserActivity() to get the 91 XX status word.		1- 14 data with status SW = 62 00 2- SW = 91 15 is returned and display text is retrieved by a FETCH
5	**getTheHandler() and postAsBERTLV() with value FALSE, then send a display text** 1- Send an unrecognizedEnvelope() to trigger the applet. 2- Send a envelopeEventDownloadUserActivity() to get the 91 XX status word.		1- 02 data are returned with status SW = 62 00 2- SW = 91 15 is returned 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 and then 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, then postAsBERTLV() with value FALSE, tag 56 and then postAsBERTLV() with value TRUE, tag 28	ToolkitException.HANDLER_NOT_AVAILABLE is thrown on the second postAsBERTLV	14 data with status word 62 00 are returned, the tag shall be 56
8	**CALL_CONTROL, the CAT Runtime Environment set the boolean value always to true** Trigger the applet with CALL CONTROL getTheHandler(), appendTLV(), postAsBERTLV() with value FALSE		12 data with status word 90 00 are returned
9	**CALL_CONTROL, the CAT Runtime Environment set the boolean value always to true** Trigger the applet with CALL CONTROL getTheHandler(), appendTLV(), postAsBERTLV() with value TRUE		12 data with status word 90 00 are returned
10	**Resulting response length greater than 256** getTheHandler(), appendArray()(255 bytes), postAsBERTLV() with value TRUE	javacard.framework.APDUException.BAD_LENGTH is thrown	

5.2.3.3 Method getLength

Test Area Reference: Api_2_Erh_Glen.

5.2.3.3.1 Conformance requirement

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

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

5.2.3.3.1.1 Normal execution

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

5.2.3.3.1.2 Parameter errors

No requirements.

5.2.3.3.1.3 Context errors

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

5.2.3.3.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Glen.java.

Test Applet: Api_2_Erh_Glen_1.java.

Cap File: api_2_erh_glen.cap.

5.2.3.3.3 Test coverage

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

5.2.3.3.4 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 getCapacity()-3** `getLength()`	Result of getLength() is getCapacity()-3	
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	**HANDLER_NOT_AVAILABLE exception** `Call post() method, then getLength()`	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.4 Method copy

Test Area Reference: Api_2_Erh_Copy.

5.2.3.4.1 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
```

5.2.3.4.1.1 Normal execution

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

5.2.3.4.1.2 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 Comprehension TLV List, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.OUT_OF_TLV_BOUNDARIES.

5.2.3.4.1.3 Context errors

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

5.2.3.4.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Copy.java.

Test Applet: Api_2_Erh_Copy_1.java.

Cap File: api_2_erh_copy.cap.

5.2.3.4.3 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

5.2.3.4.4 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 copy() dstBuffer.length = 5 dstOffset = 5 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	
3	**dstOffset < 0** copy() dstBuffer.length = 5 dstOffset = -1 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**dstLength > dstBuffer.length** copy() dstBuffer.length = 5 dstOffset = 0 dstLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**dstOffset + dstLength > dstBuffer.length** copy() dstBuffer.length = 5 dstOffset = 3 dstLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength < 0** copy() dstBuffer.length = 5 dstOffset = 0 dstLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**dstLength > length of the Comprehension TLV list** copy() dstBuffer.length = 10 dstOffset = 0 dstLength = 10	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**Successful call, dstBuffer is the whole buffer** copy() 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** copy() dstBuffer.length = 0x8F dstOffset = 3 dstLength = 0x8C	Result of copy() is 0x8F	
11	**Compare the whole buffer**	Result of arrayCompare() is 0	
12	**Successful call, dstBuffer is part of a buffer** copy() 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** copy() dstBuffer.length = 15 dstOffset = 15 dstLength = 0	Result of copy() is 15	
15	**HANDLER_NOT_AVAILABLE exception** Call post() method, then copy()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.5 Method findTLV

Test Area Reference: Api_2_Erh_Find.

5.2.3.5.1 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
```

5.2.3.5.1.1 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.

5.2.3.5.1.2 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.

5.2.3.5.1.3 Context errors

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

5.2.3.5.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Find.java.

Test Applet: Api_2_Erh_Find_1.java.

Cap File: api_2_erh_find.cap.

5.2.3.5.3 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

5.2.3.5.4 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 findTLV() Occurrence = 0	ToolkitException.BAD_INPUT_PARAMETER is thrown	
2			
	Search 1st TLV findTLV() Tag = 01h Occurrence = 1	Result is TLV_FOUND_CR_SET	
3	**Call the getValueLength() method**	Result is 03h	
4	**Search 2nd TLV** findTLV() 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 findTLV() 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** findTLV() 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 findTLV() Tag = 02h Occurrence = 2	Result is TLV_FOUND_CR_NOT_SET	
11	**Append a TLV with tag=04h**		
	Search the TLV findTLV() Tag = 04h Occurrence = 1	Result is TLV_FOUND_CR_NOT_SET	
12	**Search tag 81h** findTLV() Tag = 81h Occurrence = 1	Result is TLV_FOUND_CR_SET	
13	**Search tag 84h** findTLV() Tag = 84h Occurrence = 1	Result is TLV_FOUND_CR_NOT_SET	
14	**HANDLER_NOT_AVAILABLE exception** Call post() method, then findTLV()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.6 Method getValueLength

Test Area Reference: Api_2_Erh_Gvle.

5.2.3.6.1 Conformance requirement

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

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

5.2.3.6.1.1 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.

5.2.3.6.1.2 Parameter errors

No requirements.

5.2.3.6.1.3 Context errors

- CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.HANDLER_NOT_AVAILABLE.
- CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.UNAVAILABLE_ELEMENT.

5.2.3.6.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Gvle.java.

Test Applet: Api_2_Erh_Gvle_1.java.

Cap File: api_2_erh_gvle.cap.

5.2.3.6.3 Test coverage

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

5.2.3.6.4 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 appendTLV() with TAG 0D and length 0x7F**		
	Search TLV 0Dh		
	getValueLength()	Result is 7Fh	
5	**Clear the handler and appendTLV() with TAG 0D and length 0x80**		
	Search TLV 0Dh		
	`getValueLength()`	Result is 80h	

Id	Description	API Expectation	APDU Expectation
6	**Clear the handler and appendTLV() with TAG 0D and length 0xF1**		
	Search TLV 0Dh		
	`getValueLength()`	Result is F1h	
7	**HANDLER_NOT_AVAILABLE exception** `Call post() method, then getValueLength()`	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.7 Method getValueByte

Test Area Reference: Api_2_Erh_Gvby.

5.2.3.7.1 Conformance requirement

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

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

5.2.3.7.1.1 Normal execution

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

5.2.3.7.1.2 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.

5.2.3.7.1.3 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.

5.2.3.7.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Gvby.java.

Test Applet: Api_2_Erh_Gvby_1.java.

Cap File: api_2_erh_gvby.cap.

5.2.3.7.3 Test coverage

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

5.2.3.7.4 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(0)`	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	**Search TLV 01h**		
	`getValueByte(3)`	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
3	**Search TLV 01h**		
	`getValueByte(2)`	Result is FEh	
4	**Search TLV 02h**		
	`getValueByte(0)`	Result is 81h	
5	**appendTLV() with TAG 0D, Length 0x7E, Value: 00, 01, ..., 7D**		
	`getValueByte(7D)`	Result is 7Dh	
6	**clear the handler, appendTLV() with TAG 0D, Length 0x80, Value: 00, 01, ..., 7F**		
	`getValueByte(7E)`	Result is 7Eh	
7	`getValueByte(7F)`	Result is 7Fh	
8	**clear the handler, appendTLV() with TAG 0D, Length 0xF1, Value: 00, 01, ..., F0**		
	`getValueByte(F0)`	Result is F0h	
9	**HANDLER_NOT_AVAILABLE exception** `Call post() method, then getValueByte()`	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.8 Method copyValue

Test Area Reference: Api_2_Erh_Cpyv.

5.2.3.8.1 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
```

5.2.3.8.1.1 Normal execution

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

5.2.3.8.1.2 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.

5.2.3.8.1.3 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.

5.2.3.8.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Cpyv.java.

Test Applet: Api_2_Erh_Cpyv_1.java.

Cap File: api_2_erh_cpyv.cap.

5.2.3.8.3 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

5.2.3.8.4 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
	`copyValue()`
	`dstBuffer.length = 5`
	`dstOffset = 5`
	`dstLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

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

4	**dstLength >dstBuffer.length**
	`copyValue()`
	`dstBuffer.length = 5`
	`dstOffset = 0`
	`dstLength = 6`
		ArrayIndexOutOfBoundsException is thrown	

5	**dstOffset + dstLength >dstBuffer.length**
	`copyValue()`
	`dstBuffer.length = 5`
	`dstOffset = 3`
	`dstLength = 3`
		ArrayIndexOutOfBoundsException is thrown	

6	**dstLength < 0**
	`copyValue()`
	`dstBuffer.length = 5`
	`dstOffset = 0`
	`dstLength = -1`
		ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
7	**clear the handler, appendTLV() with TAG: 0D and length 6 Select Text String TLV**		
	valueOffset ≥ Text String Length copyValue() valueOffset = 6 dstBuffer.length = 15 dstOffset = 0 dstLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**valueOffset < 0** copyValue() valueOffset = -1 dstBuffer.length = 15 dstOffset = 0 dstLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
9	**dstLength > Text String length** copyValue() valueOffset = 0 dstBuffer.length = 15 dstOffset = 0 dstLength = 7	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
10	**valueOffset + dstLength > Text String length** copyValue() valueOffset = 2 dstBuffer.length = 15 dstOffset = 0 dstLength = 5	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
11	**Initialize 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 copyValue() 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	**Initialize dstBuffer** dstBuffer = 55 55 … 55 clear the handler AppendTLV with TAG 0x0D and value 0x01 0x02 … 0x83		
	Successful call copyValue() valueOffset = 2 dstBuffer.length = 0x86 dstOffset = 3 dstLength = 0x81	Result of copyValue() is 0x84	
15	**Compare buffer** buffer = 55 55 55 01 02 03 04 05 06 07 08 09 … 81 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	**HANDLER_NOT_AVAILABLE exception** Call post() method, then copyValue()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.9 Method compareValue

Test Area Reference: Api_2_Erh_Cprv.

5.2.3.9.1 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
```

5.2.3.9.1.1 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 Comprehension TLV List is less than that in compareBuffer.
- CRRN3: returns 1 if the first miscomparing byte in Comprehension TLV List is greater than that in compareBuffer.

5.2.3.9.1.2 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.

5.2.3.9.1.3 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.

5.2.3.9.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Cprv.java.

Test Applet: Api_2_Erh_Cprv_1.java.

Cap File: api_2_erh_cprv.cap.

5.2.3.9.3 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

5.2.3.9.4 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
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 5`
	`compareLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

3	**compareOffset < 0**
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = -1`
	`compareLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

4	**compareLength >compareBuffer.length**
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 0`
	`compareLength = 6`
		ArrayIndexOutOfBoundsException is thrown	

5	**compareOffset + compareLength >compareBuffer.length**
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 3`
	`compareLength = 3`
		ArrayIndexOutOfBoundsException is thrown	

6	**compareLength < 0**
	`compareValue()`
	`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
	`compareValue()`
	`valueOffset = 6`
	`compareBuffer.length = 15`
	`compareOffset = 0`
	`compareLength = 1`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

8	**valueOffset < 0**
	`compareValue()`
	`valueOffset = -1`
	`compareBuffer.length = 15`
	`compareOffset = 0`
	`compareLength = 1`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

9	**compareLength > Text String length**
	`compareValue()`
	`valueOffset = 0`
	`compareBuffer.length = 15`
	`compareOffset = 0`
	`compareLength = 7`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

Id	Description	API Expectation	APDU Expectation
10	**valueOffset + compareLength > Text String length** compareValue() valueOffset = 2 compareBuffer.length = 15 compareOffset = 0 compareLength = 5	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
11	**Initialize the handler**		
	compareValue()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
12	**appendTLV with TAG: 0D and value: 04 00 01 … 0F** **Select Text String TLV**		
	Initialize compareBuffer compareBuffer = 04 00 01 … 0F		
	Compare buffers compareValue() valueOffset = 0 compareOffset = 0 compareLength = 17	Result is 00h	
13	**Initialize 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	**Initialize compareBuffer** compareBuffer = 03 00 01 … 0F		
	Compare buffers with same parameters	Result is +1	
15	**Initialize compareBuffer** compareBuffer = 55 55 55 02 03 04 05 06 07 08 09 0A 0B … 81 55 55 55 55 55 clear the handler AppendTLV with TAG 0x0D and value 0x01 0x02 … 0x83		
	Compare buffers compareValue() valueOffset = 2 compareOffset = 3 compareLength = 0x81	Result is 00h	
16	**Initialize compareBuffer** compareBuffer = 55 55 55 02 03 04 05 06 07 08 09 0A 0B … 84 55 55 55 55 55		
	Compare buffers with same parameters	Result is -1	
17	**Initialize compareBuffer** compareBuffer = 55 55 55 02 03 04 05 06 07 08 09 0A 0B … 82 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	
19	**HANDLER_NOT_AVAILABLE exception** Call post() method, then compareValue()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

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

Test Area Reference: Api_2_Erh_Facyb_Bs.

5.2.3.10.1 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
```

5.2.3.10.1.1 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.

5.2.3.10.1.2 Parameter errors

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

5.2.3.10.1.3 Context errors

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

5.2.3.10.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Facyb_Bs.java.

Test Applet: Api_2_Erh_Facyb_Bs_1.java.

Cap File: api_2_erh_facyb_bs.cap.

5.2.3.10.3 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

5.2.3.10.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialize 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 findAndCopyValue() tag = 0Dh dstBuffer.length = 20 dstOffset = 20	ArrayIndexOutOfBoundsException is thrown	
3	**dstOffset < 0** findAndCopyValue() dstBuffer.length = 20 dstOffset = -1	ArrayIndexOutOfBoundsException is thrown	
4	**dstOffset + length >dstBuffer.length** findAndCopyValue() dstBuffer.length = 20 dstOffset = 5	ArrayIndexOutOfBoundsException is thrown	
5	**length > dstBuffer.length** findAndCopyValue() dstBuffer.length = 15 dstOffset = 0	ArrayIndexOutOfBoundsException is thrown	
6	**clear the handler, appendTLV() with TAG 02 and Length 02**		
	Select a TLV (tag 02h)		
	findAndCopyValue() tag = 03h	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the getValueLength() method	ToolkitException.UNAVAILABLE_ELEMENT is thrown.	
7	**appendTLV() with TAG: 0D and value: 04 00 01 … 0F**		
	Successful call findAndCopyValue() Tag = 0Dh dstBuffer.length = 17 dstOffset = 0	Result of findAndCopyValue() is 17	
8	**Compare buffer** buffer = 04 00 01 … 0F	Result is 00h	
9	**Initialize dstBuffer** dstBuffer = 55 55 … 55		
	Successful call findAndCopyValue() 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 findAndCopyValue() tag = 0Dh dstBuffer.length = 17 dstOffset = 0	Result of findAndCopyValue() is 17	
12	**Compare buffer** buffer = 04 00 01 … 0F	Result is 00h	

Id	Description	API Expectation	APDU Expectation
13	**clear the handler, appendTLV() with TAG: 0D and value: 04 00 01 … 0F**		

	Successful call (with tag 8Dh)
	`findAndCopyValue()`
	`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**
	
	`clear the Handler.`
	`AppendTLV() with tag 0x0F and value 01 02 … 80`		

	Successful call (with tag 8Fh)
	`findAndCopyValue()`
	`tag = 8Fh`
	`dstBuffer.length = 0x83`
	`dstOffset = 3`
		Result of findAndCopyValue() is 0x83	

16	**Compare buffer** `buffer = 55 55 55 00 01 … 80`	Result is 00h	
17	**HANDLER_NOT_AVAILABLE exception** Call post() method, then findAndCopyValue()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

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

Test Area Reference: Api_2_Erh_Facybbs_Bss.

5.2.3.11.1 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
```

5.2.3.11.1.1 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.

5.2.3.11.1.2 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.
- 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.

5.2.3.11.1.3 Context errors

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

5.2.3.11.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Facybbs_Bss.java.

Test Applet: Api_2_Erh_Facybbs_Bss_1.java.

Cap File: api_2_erh_facybbs_bss.cap.

5.2.3.11.3 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
P4	26
C1	25

5.2.3.11.4 Test procedure

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

	dstOffset ≥ dstBuffer.length
	`findAndCopyValue()`
	`tag = 0Dh, occurrence = 1`
	`valueOffset = 0`
	`dstBuffer.length = 5`
	`dstOffset = 5`
	`dstLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

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

4	**dstLength >dstBuffer.length**
	`findAndCopyValue()`
	`dstBuffer.length = 5`
	`dstOffset = 0`
	`dstLength = 6`
		ArrayIndexOutOfBoundsException is thrown	

5	**dstOffset + dstLength >dstBuffer.length**
	`findAndCopyValue()`
	`dstBuffer.length = 5`
	`dstOffset = 3`
	`dstLength = 3`
		ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
6	**dstLength < 0** findAndCopyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**appendTLV() with TAG: 0D and length 6**		
	valueOffset ≥ Text String Length findAndCopyValue() tag = 0Dh, occurrence = 1 valueOffset = 6 dstBuffer.length = 15 dstOffset = 0 dstLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**valueOffset < 0** findAndCopyValue() valueOffset = -1 dstBuffer.length = 15 dstOffset = 0 dstLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
9	**dstLength > Text String length** findAndCopyValue() valueOffset = 0 dstBuffer.length = 15 dstOffset = 0 dstLength = 7	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
10	**valueOffset + dstLength > Text String length** findAndCopyValue() 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	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the getValueLength() method	ToolkitException.UNAVAILABLE_ELEMENT is thrown.	
12	**clear the handler and appendTLV() with TAG: 0D and value: 04 00 01 … 0F**		
	Successful call findAndCopyValue() 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	**Initialize dstBuffer** dstBuffer = 55 55 … 55		
	Successful call findAndCopyValue() 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	

Id	Description	API Expectation	APDU Expectation
16	**Append a Text String TLV** tag = 0D buffer = 00 11 22 33 44 55 (no specific DCS byte)		
	Successful call findAndCopyValue() 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** findAndCopyValue() 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) findAndCopyValue() 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 AppendTLV() with tag 0x0F and value O1 02 … 80		
	Successful call (with tag 8Fh) findAndCopyValue() tag = 8Fh occurrence = 1 valueOffset = 0 dstBuffer.length = 0x83 dstOffset = 0 dstLength = 0x80	Result of findAndCopyValue() is 0x80	
23	**Compare buffer** buffer = 00 01 … 80 55 55 55	Result is 00h	
24	**Successful call, findAndCopyValue() with length =0** dstBuffer.length = 16 dstOffset = 16 dstLength = 0	Result of findAndCopyValue() is 16	
25	**Invalid parameter** findAndCopyValue() **occurrence = 0**	ToolkitException.BAD_INPUT_PARAMETER is thrown	
26	**HANDLER_NOT_AVAILABLE exception** Call post() method, then findAndCopyValue()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

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

Test Area Reference: Api_2_Erh_Facrb_Bs.

5.2.3.12.1 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
```

5.2.3.12.1.1 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 Comprehension TLV is less than that in compareBuffer returns -1.
- CRRN5: if the first miscomparing byte in Comprehension TLV is greater than that in compareBuffer returns 1.
- CRRN6: The search method is comprehension required flag independent.

5.2.3.12.1.2 Parameter errors

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

5.2.3.12.1.3 Context errors

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

5.2.3.12.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Facrb_Bs.java.

Test Applet: Api_2_Erh_Facrb_Bs_1.java.

Cap File: api_2_erh_facrb_bs.cap.

5.2.3.12.3 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

5.2.3.12.4 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
	`findAndCompareValue()`
	`tag = 0Dh`
	`compareBuffer.length = 20`
	`compareOffset = 20`
		ArrayIndexOutOfBoundsException is thrown	

3	**compareOffset < 0**
	`findAndCompareValue()`
	`compareBuffer.length = 20`
	`compareOffset = -1`
		ArrayIndexOutOfBoundsException is thrown	

4	**compareOffset + length > compareBuffer.length**
	`findAndCompareValue()`
	`compareBuffer.length = 20`
	`compareOffset = 5`
		ArrayIndexOutOfBoundsException is thrown	

5	**length > compareBuffer.length**
	`findAndCompareValue()`
	`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**		

	Initialize compareBuffer
	`compareBuffer =`
	`04 00 01 … 0F`		

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

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

10	**Initialize compareBuffer**
	`compareBuffer =`
	`04 00 01 … 10`		

	Compare buffers with same parameters	Result is -1	

11	**Initialize compareBuffer**
	`compareBuffer =`
	`03 00 01 … 0F`		

	Compare buffers with same parameters	Result is +1	

12	**Initialize 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
13	**Successful call with 2 TLVs** clear() the Handler AppendTLV() with tag 0x0D and value set to 00 01 … 80 81 Append a second TLV with tag 0x0D and value set to 00 11 22 33 44 55		
	Initialize compareBuffer compareBuffer = 55 55 55 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F .. 81		
	Compare buffers findAndCompareValue() the first TLV compareOffset = 3	Result is 00h	
14	**Initialize compareBuffer** CompareBuffer = 55 55 55 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E .. 7F 81		
	Compare buffers findAndCompareValue() the first TLV compareOffset = 3	Result is -1	
15	**Initialize compareBuffer** compareBuffer = 55 55 55 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E .. 80 80		
	Compare buffers compareOffset = 3	Result is +1	
16	**clear the handler and appendTLV() with TAG: 0D and value: 04 00 01 … 0F**		
	Initialize compareBuffer compareBuffer = 04 00 01 … 0F		
	Successful call (with tag 8Dh) findAndCompareValue() tag = 8Dh compareBuffer.length = 17 compareOffset = 0	Result is 00h	
17	**Append tag 0Fh** buffer = 00 01 … 0F		
	Initialize compareBuffer compareBuffer = 00 01 … 0F		
	Successful call (with tag 8Fh) findAndCompareValue() tag = 8Fh compareBuffer.length = 16 compareOffset = 0	Result is 00h	
18	**HANDLER_NOT_AVAILABLE exception** Call post() method, then findAndCompareValue()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

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

Test Area Reference: Api_2_Erh_Facrbbs_Bss.

5.2.3.13.1 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
```

5.2.3.13.1.1 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 Comprehension TLV is less than that in compareBuffer -1 is returned.
- CRRN5: if the first miscomparing byte in Comprehension TLV is greater than that in compareBuffer 1 is returned.
- CRRN6: The search method is comprehension required flag independent.

5.2.3.13.1.2 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, 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.

5.2.3.13.1.3 Context errors

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

5.2.3.13.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Facrbbs_Bss.java.

Test Applet: Api_2_Erh_Facrbbs_Bss_1.java.

Cap File: api_2_erh_facrbbs_bss.cap.

5.2.3.13.3 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

5.2.3.13.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialize the handler**		
	findAndCompareValue() with a null compareBuffer	NullPointerException is thrown	
2	**clear the handler and appendTLV() with TAG: 0D and value: 04 00 01 … 0F**		

	compareOffset ≥ compareBuffer.length
	`findAndCompareValue()`
	`tag = 0Dh, occurrence = 1`
	`valueOffset = 0`
	`compareBuffer.length = 5`
	`compareOffset = 5`
	`compareLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

3	**compareOffset < 0**
	`findAndCompareValue()`
	`compareBuffer.length = 5`
	`compareOffset = -1`
	`compareLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

4	**compareLength >compareBuffer.length**
	`findAndCompareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 0`
	`compareLength = 6`
		ArrayIndexOutOfBoundsException is thrown	

5	**compareOffset + compareLength >compareBuffer.length**
	`findAndCompareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 3`
	`compareLength = 3`
		ArrayIndexOutOfBoundsException is thrown	

6	**compareLength < 0**
	`findAndCompareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 0`
	compareLength = -1
		ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
7	**clear the handler and appendTLV() with TAG and length of 6**		

	valueOffset ≥ Text String Length
	`findAndCompareValue()`
	`tag = 0Dh, occurrence = 1`
	`valueOffset = 6`
	`compareBuffer.length = 15`
	`compareOffset = 0`
	`compareLength = 1`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

8	**valueOffset < 0**
	`findAndCompareValue()`
	`valueOffset = -1`
	`compareBuffer.length = 15`
	`compareOffset = 0`
	`compareLength = 1`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

9	**compareLength > Text String length**
	`findAndCompareValue()`
	`valueOffset = 0`
	`compareBuffer.length = 15`
	`compareOffset = 0`
	`compareLength = 7`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

10	**valueOffset + compareLength > Text String length**
	`findAndCompareValue()`
	`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**		

	Initialize 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	

16	**Initialize compareBuffer**
	`compareBuffer =`
	`04 00 01 … 10`		

	Compare buffers with same parameters	Result is -1	

17	**Initialize compareBuffer**
	`compareBuffer =`
	`03 00 01 … 0F`		

	Compare buffers with same parameters	Result is +1	

Id	Description	API Expectation	APDU Expectation
18	**Initialize compareBuffer** clear() the handler. AppendTLV with tag 0x0D and data = 01 02 03 … 80 81 compareBuffer = 55 55 55 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E … 80 81		
	Compare buffers findAndCompareValue() tag = 0x0D occurrence = 1 valueOffset = 0 compareOffset = 3 compareLength = 0x81	Result is 00h	
19	**Initialize compareBuffer** compareBuffer = 55 55 55 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E … 7F 81		
	findAndCompareValue() with same parameters	Result is -1	
20	**Initialize compareBuffer** compareBuffer = 55 55 55 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E … 80 80		
	findAndCompareValue() with same parameters	Result is +1	
21	**append a second Text String TLV** tag = 0Dh buffer = 00 11 22 33 44 55		
	Initialize compareBuffer compareBuffer = 55 55 55 01 … 80 81		
	findAndCompareValue() tag = 0Dh, occurrence = 1 valueOffset = 0 compareOffset = 3 compareLength = 0x81	Result is 00h	
22	**Initialize compareBuffer** compareBuffer = 00 11 22 33 44 55		
	findAndCompareValue() tag = 0Dh, occurrence = 2 valueOffset = 0 compareOffset = 0 compareLength = 6	Result is 00h	
23	**Initialize 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**		
	Initialize compareBuffer compareBuffer = 04 00 01 … 0F		
	Successful call (with tag 8Dh) findAndCompareValue() 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		
	Initialize compareBuffer compareBuffer = 00 01 … 0F		
	Successful call (with tag 8Fh) findAndCompareValue() 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	**HANDLER_NOT_AVAILABLE exception** Call post() method, then findAndCompareValue()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.14 Method appendArray

Test Area Reference: Api_2_Erh_Apda.

5.2.3.14.1 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
```

5.2.3.14.1.1 Normal execution

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

5.2.3.14.1.2 Parameter errors

- 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.

5.2.3.14.1.3 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.

5.2.3.14.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Apda_Bss.java.

Test Applet: Api_2_Erh_Apda_Bss_1.java.

Cap File: api_2_erh_apda_bss.cap.

5.2.3.14.3 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

5.2.3.14.4 Test procedure

Id	Description	API Expectation	APDU Expectation
	Initialize the envelope response handler with a TLV of length 1		
1	**Null buffer**	NullPointerException is thrown	
2	**offset ≥ buffer.length** appendArray() buffer.length = 5 offset = 5 length = 1	ArrayIndexOutOfBoundsException is thrown	
3	**offset < 0** appendArray() buffer.length = 5 offset = -1 length = 1	ArrayIndexOutOfBoundsException is thrown	
4	**length > buffer.length** appendArray() buffer.length = 5 offset = 0 length = 6	ArrayIndexOutOfBoundsException is thrown	
5	**offset + length > buffer.length** appendArray() buffer.length = 5 offset = 3 length = 3	ArrayIndexOutOfBoundsException is thrown	
6	**length < 0** appendArray() buffer.length = 5 offset = 0 length = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Handler overflow exception** appendArray() buffer.length = getCapacity()+1 offset = 0 length = getCapacity()+1	ToolkitException.HANDLER_OVERFLOW is thrown	
8	**append the handler with TLVs:** 81 03 11 22 33 82 02 99 77		
	findTLV() 0x81		
	Successful call appendArray() buffer = FF FE … F8 offset = 0 length = 8		
	Verify Current TLV: Call getValueLength()	Result is 03h	

Id	Description	API Expectation	APDU Expectation
9	**Clear the handler**		
	Successful call appendArray() buffer = FF FE … F8 offset = 0 length = 8		
	Call copy() method		
	Compare handler compareBuffer = FF FE … F8	Result is 00h	
10	**Successful call** appendArray() 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** appendArray() 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	**HANDLER_NOT_AVAILABLE exception** Call post() method, then appendArray()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.15 Method appendTLV(byte tag, byte value)

Test Area Reference: Api_2_Erh_Aptlbb.

5.2.3.15.1 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
```

5.2.3.15.1.1 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.

5.2.3.15.1.2 Parameter errors

No requirements.

5.2.3.15.1.3 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.

5.2.3.15.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Aptlbb.java.

Test Applet: Api_2_Erh_Aptlbb_1.java.

Cap File: api_2_erh_aptlbb.cap.

5.2.3.15.3 Test coverage

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

5.2.3.15.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call appendArray()** `length = getCapacity()-1`		
	Handler Overflow exception: Call 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() method	Result is 03h	
3	**Clear the handler**		

	Successful call
	`appendTLV()`
	`tag = 84h`
	`value = 00h`		

	Call copy() method		
	Compare handler `compareBuffer = 84 01 00`	Result is 00h	

4	**Successful call**
	`appendTLV()`
	`tag = 01h`
	`value = Feh`		

	Call copy() method		
	Compare handler `compareBuffer = 84 01 00 01 01 FE`	Result is 00h	
5	**HANDLER_ NOT_AVAILABLE exception** `Call post() method, then appendTLV()`	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

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

Test Area Reference: Api_2_Erh_Aptlbbb.

5.2.3.16.1 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
```

5.2.3.16.1.1 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.

5.2.3.16.1.2 Parameter errors

No requirements.

5.2.3.16.1.3 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.

5.2.3.16.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Aptl Bbb.java.

Test Applet: Api_2_Erh_Aptl Bbb_1.java.

Cap File: api_2_erh_aptl bbb.cap.

5.2.3.16.3 Test coverage

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

5.2.3.16.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the appendArray() with length of getCapacity()-1**		
	Handler Overflow exception: Call 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 appendTLV() method		
	Verify Current TLV: Call getValueLength() method	Result is 03h	
3	**Clear the handler**		
	Successful call appendTLV() tag = 84h value1 = 00h value2 = 01h		
	Call copy() method		
	Compare handler compareBuffer = 84 02 00 01	Result is 00h	

Id	Description	API Expectation	APDU Expectation
4	**Successful call** appendTLV() tag = 01h value1 = FEh value2 = FDh		
	Call copy() method		
	Compare handler compareBuffer = 84 02 00 01 01 02 FE FD	Result is 00h	
5	**HANDLER_NOT_AVAILABLE exception** Call post() method, then appendTLV()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

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

Test Area Reference: Api_2_Erh_Aptlb_Bss.

5.2.3.17.1 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
```

5.2.3.17.1.1 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.

5.2.3.17.1.2 Parameter errors

- 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.

5.2.3.17.1.3 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.
- CRRC3: if valuelength is greater than 255, a ToolkitException is thrown with reason code BAD_INPUT_PARAMETER.

5.2.3.17.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Aptlb_Bss.java.

Test Applet: Api_2_Erh_Aptlb_Bss_1.java.

Cap File: api_2_erh_aptlb_bss.cap.

5.2.3.17.3 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

5.2.3.17.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Null value**	NullPointerException is thrown	
2	**valueOffset ≥ value.length** appendTLV() value.length = 5 valueOffset = 5 valueLength = 1	ArrayIndexOutOfBoundsException is thrown	
3	**valueOffset < 0** appendTLV() value.length = 5 valueOffset = -1 valueLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**valueLength > value.length** appendTLV() value.length = 5 valueOffset = 0 valueLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**ValueOffset + valueLength > value.length** appendTLV() value.length = 5 valueOffset = 3 valueLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**valueLength < 0** appendTLV() value.length = 5 valueOffset = 0 valueLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Handler overflow exception** Call the appendArray() with length of getCapacity()-1, appendTLV() value.length = 256 valueOffset = 0 valueLength = 254	ToolkitException.HANDLER_OVERFLOW is thrown	
8	**Bad parameter exception** Clear the handler, appendTLV() 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 appendTLV() tag = 04 value = FF FE … F8 valueOffset = 0 valueLength = 8		
	Verify Current TLV: Call getValueLength()	Result is 03h	

Id	Description	API Expectation	APDU Expectation
10	**Clear the handler**		
	Successful call appendTLV() 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** appendTLV() 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	
12	**Successful call** appendTLV() 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 appendTLV() 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	**HANDLER_NOT_AVAILABLE exception** Call post() method, then appendTLV()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

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

Test Area Reference: Api_2_Erh_Aptlbb_Bss.

5.2.3.18.1 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
```

5.2.3.18.1.1 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.

5.2.3.18.1.2 Parameter errors

- 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.

5.2.3.18.1.3 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 254, a ToolkitException is thrown with reason code BAD_INPUT_PARAMETER.

5.2.3.18.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Aptlbb_Bss.java.

Test Applet: Api_2_Erh_Aptlbb_Bss_1.java.

Cap File: api_2_erh_aptlbb_bss.cap.

5.2.3.18.3 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

5.2.3.18.4 Test procedure

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

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

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

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

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

Id	Description	API Expectation	APDU Expectation
6	**value2Length < 0** appendTLV() value2.length = 5 value2Offset = 0 value2Length = -1	ArrayIndexOutOfBoundsExceptio n is thrown	
7	**Handler overflow exception** Call the appendArray() with length of getCapacity()-1 appendTLV() value2.length = 256 value2Offset = 0 value2Length = 254	ToolkitException.HANDLER_OVE RFLOW is thrown	
8	**Bad parameter exception** Clear the handlerappendTLV() value2.length = 256 value2Offset = 0 value2Length = 256	ToolkitException.BAD_INPUT_PA RAMETER 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 appendTLV() 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 appendTLV() 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** appendTLV() 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** appendTLV() 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	

Id	Description	API Expectation	APDU Expectation
13	**Clear the handler**		
	Successful call appendTLV() 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	**HANDLER_NOT_AVAILABLE exception** Call post() method, then appendTLV()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.19 Method clear

Test Area Reference: Api_2_Erh_Cler.

5.2.3.19.1 Conformance requirement

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

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

5.2.3.19.1.1 Normal execution

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

5.2.3.19.1.2 Parameter errors

No requirements.

5.2.3.19.1.3 Context errors

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

5.2.3.19.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Cler.java.

Test Applet: Api_2_Erh_Cler_1.java.

Cap File: api_2_erh_cler.cap.

5.2.3.19.3 Test coverage

CRR number	Test case number
N1	1, 2
C1	3

5.2.3.19.4 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`
		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	
3	**HANDLER_NOT_AVAILABLE exception** `Call appendTLV() method, then post() and then clear()`	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.20 Method getCapacity

Test Area Reference: Api_2_Erh_Gcap.

5.2.3.20.1 Conformance requirement

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

```
public byte getCapacity()
```

5.2.3.20.1.1 Normal execution

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

5.2.3.20.1.2 Parameter errors

No requirements

5.2.3.20.1.3 Context errors

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

5.2.3.20.2 Test area files

Test Source: Test_Api_2_Erh_Gcap.java.

Test Applet: Api_2_Erh_Gcap_1.java.

Cap File: api_2_erh_gcap.cap.

5.2.3.20.3 Test coverage

CRR number	Test case number
N1	1
C1	Tested in CAT Runtime Environment part: FWK_MHA_ERHD

5.2.3.20.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**EnvelopeResponseHandler available** 1- Send envelope Menu Selection 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	

5.2.3.21 Method getValueShort

Test Area Reference: Api_2_Erh_Gvsh.

5.2.3.21.1 Conformance requirement

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

```
public short getValueShort(short valueOffset)
                    throws ToolkitException
```

5.2.3.21.1.1 Normal execution

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

5.2.3.21.1.2 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.

5.2.3.21.1.3 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.

5.2.3.21.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Gvsh.java.

Test Applet: Api_2_Erh_Gvsh_1.java.

Cap File: api_2_erh_gvsh.cap.

5.2.3.21.3 Test coverage

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

5.2.3.21.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**appendTLV() 82 02 81 82, appendTLV() 81 03 11 22 FE** findTLV() with TAG 03		
	getValueShort(0)	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	Search TLV 01h		
	getValueShort(3)	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
3	Search TLV 01h		
	getValueShort(1)	Result is 22h FEh	
4	Search TLV 02h		
	getValueShort(0)	Result is 81h 82h	
5	appendTLV() with TAG 0D, Length 0x7E, Value: 00, 01, ..., 7D		
	getValueShort(7C)	Result is 7Ch 7Dh	
6	clear the handler, appendTLV() with TAG 0D, Length 0x80, Value: 00, 01, ..., 7F		
	getValueShort(7D)	Result is 7Dh 7Eh	
7	getValueShort(7E)	Result is 7Eh 7Fh	
8	clear the handler, appendTLV() with TAG 0D, Length 0xF1, Value: 00, 01, ..., F0		
	getValueShort(EF)	Result is EFh F0h	
9	**HANDLER_NOT_AVAILABLE exception** Call post() method, then getValueShort()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.22 Method appendTLV(byte tag, byte value1, short value2)

Test Area Reference: Api_2_Erh_Aptlbbs.

5.2.3.22.1 Conformance requirements

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

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

5.2.3.22.1.1 Normal execution

- CRRN1: Appends a TLV element to the current TLV list (3-byte element(1-byte,1-short)).
- CRRN2: A successful append does not modify the TLV selected.

5.2.3.22.1.2 Parameter errors

No requirements.

5.2.3.22.1.3 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.

5.2.3.22.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Aptlbbs.java.

Test Applet: Api_2_Erh_Aptlbbs_1.java.

Cap File: api_2_erh_aptlbbs.cap.

5.2.3.22.3 Test coverage

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

5.2.3.22.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the appendArray() with length of getCapacity()-1**		
	Handler Overflow Exception: Call appendTLV() method	ToolkitException.HANDLER_OVERFLOW is thrown	
2	**clear the handler, append the handler with TLVs:** 81 03 11 22 33 82 03 99 77 00		
	Select Command Details TLV		
	Call appendTLV() method		
	Verify Current TLV: Call getValueLength() method	Result is 03h	
3	**Clear the handler**		
	Successful call appendTLV() tag = 84h value1 = 00h value2 = 01h 02h		
	Call copy() method		
	Compare handler compareBuffer = 84 03 00 01 02	Result is 00h	
4	**Successful call** appendTLV() tag = 01h value1 = FEh value2 = FDh FCh		
	Call copy() method		
	Compare handler compareBuffer = 84 03 00 01 02 01 03 FE FD FC	Result is 00h	
5	**HANDLER_NOT_AVAILABLE exception** Call post() method, then AppendTLV()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.23 Method appendTLV(byte tag, short value)

Test Area Reference: Api_2_Erh_Aptlbs.

5.2.3.23.1 Conformance requirements

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

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

5.2.3.23.1.1 Normal execution

- CRRN1: Appends a TLV element to the current TLV list (2-byte or 1-short element).
- CRRN2: A successful append does not modify the TLV selected.

5.2.3.23.1.2 Parameter errors

No requirements.

5.2.3.23.1.3 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.

5.2.3.23.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Aptlbs.java.

Test Applet: Api_2_Erh_Aptlbs_1.java.

Cap File: api_2_erh_aptlbs.cap.

5.2.3.23.3 Test coverage

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

5.2.3.23.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call appendArray()** `length = getCapacity()-1`		
	Handler Overflow Exception: Call appendTLV() method	ToolkitException.HANDLER_OVERFLOW is thrown	

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() method	Result is 03h	
3	**Clear the handler**		

	Successful call
	`appendTLV()`
	`tag = 84h`
	`value = 00h 01h`		

	Call copy() method		
	Compare handler `compareBuffer = 84 02 00 01`	Result is 00h	

Id	Description	API Expectation	APDU Expectation
4	**Successful call** appendTLV() tag = 01h value = FEh FFh		
	Call copy() method		
	Compare handler compareBuffer = 84 02 00 01 01 02 FE FF	Result is 00h	
5	**HANDLER_NOT_AVAILABLE exception** Call post() method, then AppendTLV()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.24 Method appendTLV(byte tag, short value1, short value2)

Test Area Reference: Api_2_Erh_Aptlbss.

5.2.3.24.1 Conformance requirements

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

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

5.2.3.24.1.1 Normal execution

- CRRN1: Appends a TLV element to the current TLV list (4-byte element(2-short)).
- CRRN2: A successful append does not modify the TLV selected.

5.2.3.24.1.2 Parameter errors

No requirements.

5.2.3.24.1.3 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.

5.2.3.24.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Aptlbss.java.

Test Applet: Api_2_Erh_Aptlbss_1.java.

Cap File: api_2_erh_aptlbss.cap.

5.2.3.24.3 Test coverage

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

5.2.3.24.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the appendArray() method with length equal getCapacity()-1**		
	Handler Overflow Exception: Call 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 appendTLV() method		
	Verify Current TLV: Call getValueLength() method	Result is 03h	
3	**Clear the handler**		
	Successful call appendTLV() tag = 84h value1 = 00h 01h value2 = 02h 03h		
	Call copy() method		
	Compare handler compareBuffer = 84 04 00 01 02 03	Result is 00h	
4	**Successful call** appendTLV() tag = 01h value1 = FEh FDh value2 = FCh FBh		
	Call copy() method		
	Compare handler compareBuffer = 84 04 00 01 02 03 01 04 FE FD FC FB	Result is 00h	
5	**HANDLER_NOT_AVAILABLE exception** Call post() method, then appendTLV()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.3.25 Method appendTLV(byte tag, byte[] value1, short value1Offset, short value1Length, byte[] value2, short value2Offset, short value2Length)

Test Area Reference: Api_2_Erh_Aptlb_Bss_Bss.

5.2.3.25.1 Conformance requirements

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

```
public void appendTLV(byte tag,
                      byte[] value1,
                      short value1Offset,
                      short value1Length,
                      byte[] value2,
                      short value2Offset,
                      short value2Length)
               throws java.lang.NullPointerException,
                      java.lang.ArrayIndexOutOfBoundsException,
                      ToolkitException
```

5.2.3.25.1.1 Normal execution

- CRRN1: Appends a TLV element to the current TLV list (2 byte arrays format).
- CRRN2: A successful append does not modify the TLV selected.

5.2.3.25.1.2 Parameter errors

- CRRP1: If value1 or value2 is null, a NullPointerException is thrown.
- CRRP2: If value1Offset or value1Length or both would cause access outside value1 array bounds, or if value1Length is negative, an ArrayIndexOutOfBoundsException is thrown.
- CRRP3: If value2Offset or value2Length or both would cause access outside value2 array bounds, or if value2Length is negative, an ArrayIndexOutOfBoundsException is thrown.

5.2.3.25.1.3 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 value1Length or value2Length is greater than 255, a ToolkitException is thrown with reason code BAD_INPUT_PARAMETER.

5.2.3.25.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Erh_Aptlb_Bss_Bss.java.

Test Applet: Api_2_Erh_Aptlb_Bss_Bss_1.java.

Cap File: api_2_erh_aptlb_bss_bss.cap.

5.2.3.25.3 Test coverage

CRR number	Test case number
N1	18, 19, 20
N2	17
P1	1, 2
P2	3, 4, 5, 6, 7
P3	8, 9, 10, 11, 12
C1	13
C2	21
C3	14, 15

5.2.3.25.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Null value1**	NullPointerException is thrown	
2	**Null value2**	NullPointerException is thrown	

3	**Value1Offset ≥ value1.length**
	`appendTLV()`
	`value1.length = 5`
	`value1Offset = 5`
	`value1Length = 1`
	`value2.length = 5`
	`value2Offset = 0`
	`value2Length = 1`
		ArrayIndexOutOfBoundsException is thrown	

4	**Value1Offset < 0**
	`appendTLV()`
	`value1.length = 5`
	`value1Offset = -1`
	`value1Length = 1`
	`value2.length = 5`
	`value2Offset = 0`
	`value2Length = 1`
		ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
5	**Value1Length > value1.length** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 6 value2.length = 5 value2Offset = 0 value2Length = 1	ArrayIndexOutOfBoundsException is thrown	
6	**Value1Offset + value1Length > value1.length** appendTLV() Value1.length = 5 value1Offset = 3 value1Length = 3 value2.length = 5 value2Offset = 0 value2Length = 1	ArrayIndexOutOfBoundsException is thrown	
7	**Value1Length < 0** appendTLV() value1.length = 5 value1Offset = 0 value1Length = -1 value2.length = 5 value2Offset = 0 value2Length = 1	ArrayIndexOutOfBoundsException is thrown	
8	**Value2Offset ≥ value2.length** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 value2.length = 5 value2Offset = 5 value2Length = 1	ArrayIndexOutOfBoundsException is thrown	
9	**Value2Offset < 0** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 value2.length = 5 value2Offset = -1 value2Length = 1	ArrayIndexOutOfBoundsException is thrown	
10	**Value2Length > value2.length** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 value2.length = 5 value2Offset = 0 value2Length = 6	ArrayIndexOutOfBoundsException is thrown	
11	**Value2Offset + value2Length > value2.length** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 Value2.length = 5 Value2Offset = 3 Value2Length = 3	ArrayIndexOutOfBoundsException is thrown	
12	**Value2Length < 0** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 value2.length = 5 value2Offset = 0 value2Length = -1	ArrayIndexOutOfBoundsException is thrown	
13	**Handler overflow Exception** Call the appendArray() with length of getCapacity()-1 appendTLV() Value1.length = 256 Value1Offset = 0 Value1Length = 1 Value2.length = 256 Value2Offset = 0 Value2Length = 1	ToolkitException.HANDLER_OVERFLOW is thrown	

Id	Description	API Expectation	APDU Expectation
14	**Bad parameter Exception** Clear the handler appendTLV() Value1.length = 256 Value1Offset = 0 Value1Length = 256 Value2.length = 256 Value2Offset = 0 Value2Length = 1	ToolkitException.BAD_INPUT_PARAMETER is thrown	
15	**Bad parameter Exception** appendTLV() Value1.length = 256 Value1Offset = 0 Value1Length = 1 Value2.length = 256 Value2Offset = 0 Value2Length = 256	ToolkitException.BAD_INPUT_PARAMETER is thrown	
	clear the handler, append the handler with TLVs: 81 03 11 22 33 82 02 99 77		
	Select Command Details TLV		
16	**Successful call** appendTLV() tag = 04 value1 = FF FE … F8 value1Offset = 0 value1Length = 8 value2 = F7 F6 … F0 value2Offset = 0 value2Length = 8		
	Verify Current TLV: Call getValueLength()	Result is 03h	
	Clear the handler		
17	**Successful call** appendTLV() tag = 04 value1 = FF FE … F8 value1Offset = 0 value1Length = 8 value2 = F7 F6 … F0 value2Offset = 0 value2Length = 8		
	Call copy() method		
	Compare handler CompareBuffer = 04 10 FF FE … F0	Result is 00	
18	**Successful call** appendTLV() tag = 85h value1 = 00 01 … 07 value1Offset = 2 value1Length = 6 value2 = 08 09 … 0F value2Offset = 2 value2Length = 6		
	Call copy() method		
	Compare handler compareBuffer = 04 10 FF FE … F0 85 0C 02 03 04 05 06 07 0A 0B 0C 0D 0E 0F	Result is 00	
19	**Successful call** appendTLV() tag = 01 value1 = 11 22 … 88 value1Offset = 2 value1Length = 4 value2 = 99 AA … FF 00 value2Offset = 2 value2Length = 4		
	Call copy() method		

Id	Description	API Expectation	APDU Expectation
	Compare handler compareBuffer = 04 10 FF FE … F0 85 0C 02 03 04 05 06 07 0A 0B 0C 0D 0E 0F 01 08 33 44 55 66 BB CC DD EE	Result is 00	
	Clear the handler		
20	**Successful call** appendTLV() tag = 04 value1 = 00 01 … 7F value1Offset = 0 value1Length = 80h value2 = 80 81 … FC value2Offset = 0 value2Length = 7Dh		
	Call copy() method		
	Compare handler compareBuffer = 04 81 FD 00 01…FC	Result is 00	
21	**HANDLER_NOT_AVAILABLE exception** Call post() method, then appendTLV()	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.4 Interface ProactiveHandler

5.2.4.1 Method init

Test Area Reference: Api_2_Pah_Init.

5.2.4.1.1 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)
```

5.2.4.1.1.1 Normal execution

- CRRN1: The init() method initializes the next Proactive command in the ProactiveHandler, with Command details and Device Identities TLV. The source device is always the UICC 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.

5.2.4.1.1.2 Parameter errors

No requirements.

5.2.4.1.1.3 Context errors

No requirements.

5.2.4.1.2 Test area files

Test Source: Test_Api_2_Pah_Init.java.

Test Applet: Api_2_Pah_Init_1.java.

Cap File: api_2_pah_init.cap.

5.2.4.1.3 Test coverage

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

5.2.4.1.4 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()	

5.2.4.2 Method initDisplayText

Test Area Reference: Api_2_Pah_Indt.

5.2.4.2.1 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
```

5.2.4.2.1.1 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 initializes 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 ETSI TS 101 267 [11].

5.2.4.2.1.2 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.

5.2.4.2.1.3 Context errors

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

5.2.4.2.2 Test area files

Test Source: Test_Api_2_Pah_Indt.java.

Test Applet: Api_2_Pah_Indt_1.java.

Cap File: api_2_pah_indt.cap.

5.2.4.2.3 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

5.2.4.2.4 Test procedure

Id	Description	API Expectation	APDU Expectation

1	**NULL as parameter to buffer**
	`initDisplayText()`
	`buffer = NULL`
		NullPointerException is thrown	

2	**offset > buffer.length**
	`initDisplayText()`
	`buffer = "Text"`
	`offset = 5`
	`length = 0`
		ArrayIndexOutOfBoundsException is thrown	

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

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

Id	Description	API Expectation	APDU Expectation
5	**offset + length > buffer.length** initDisplayText() buffer = "Text" offset = 3 length = 2	ArrayIndexOutOfBoundsException is thrown	
6	**length < 0** initDisplayText() buffer = "Text" offset = 3 length = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Successful call, buffer is the whole buffer** initDisplayText() 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** initDisplayText() 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** initDisplayText() 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** initDisplayText() 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** initDisplayText() qualifier = 81h dcs = 4 buffer = "TextE" offset = 0 length = 5		DISPLAY TEXT Proactive command qualifier = 81h dcs = 4 Text = "TextE"
13	**Successful call, DCS=0 (7 bits)** **Send the command** initDisplayText() qualifier = 0 dcs = 0 buffer = "TextF" offset = 0 length = 5		DISPLAY TEXT Proactive command qualifier = 00h dcs = 0 Text = "TextF"

Id	Description	API Expectation	APDU Expectation
14	**Successful call, DCS=8 (UCS2)** **Send the command** initDisplayText() 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** initDisplayText() 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** initDisplayText() 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** initDisplayText() 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** initDisplayText() 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'

5.2.4.3 Method initGetInkey

Test Area Reference: Api_2_Pah_Ingk.

5.2.4.3.1 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
```

5.2.4.3.1.1 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 initializes 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 ETSI TS 101 267 [11].

5.2.4.3.1.2 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.

5.2.4.3.1.3 Context errors

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

5.2.4.3.2 Test area files

Test Source: Test_Api_2_Pah_Ingk.java.

Test Applet: Api_2_Pah_Ingk_1.java.

Cap File: api_2_pah_ingk.cap.

5.2.4.3.3 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

5.2.4.3.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**NULL as parameter to buffer** initGetInkey() buffer = NULL	NullPointerException is thrown	
2	**offset > buffer.length** initGetInkey() buffer = "Text" offset = 5	ArrayIndexOutOfBoundsException is thrown	
3	**offset < 0** initGetInkey() buffer = "Text" offset = -1	ArrayIndexOutOfBoundsException is thrown	
4	**length > buffer.length** initGetInkey() buffer = "Text" offset = 0 length = 5	ArrayIndexOutOfBoundsException is thrown	
5	**offset + length > buffer.length** initGetInkey() buffer = "Text" offset = 3 length = 2	ArrayIndexOutOfBoundsException is thrown	
6	**length < 0** initGetInkey() buffer = "Text" offset = 3 length = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Successful call, buffer is the whole buffer** initGetInkey() 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** initGetInkey() 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** initGetInkey() 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** initGetInkey() qualifier = 0 dcs = 4 buffer = "12TextD34" offset = 2 length = 5		GET INKEY Proactive command qualifier = 00h dcs = 4 Text = "TextD"

Id	Description	API Expectation	APDU Expectation
12	**Successful call, qualifier = 81h** initGetInkey() qualifier = 81h dcs = 4 buffer = "TextE" offset = 0 length = 5		GET INKEY Proactive command qualifier = 81h dcs = 4 Text = "TextE"
13	**Successful call, DCS=0 (7 bits)** initGetInkey() 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)** initGetInkey() 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** initGetInkey() 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** initGetInkey() 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** initGetInkey() 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** initGetInkey() 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'

5.2.4.4 Method initGetInput

Test Area Reference: Api_2_Pah_Ingp.

5.2.4.4.1 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
```

5.2.4.4.1.1 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 initializes 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 ETSI TS 101 267 [11].

5.2.4.4.1.2 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.

5.2.4.4.1.3 Context errors

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

5.2.4.4.2 Test area files

Test Source: Test_Api_2_Pah_Ingp.java.

Test Applet: Api_2_Pah_Ingp_1.java.

Cap File: api_2_pah_ingp.cap.

5.2.4.4.3 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

CRR number	Test case number
P2	2, 3, 4, 5, 6
C1	21

5.2.4.4.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**NULL as parameter to buffer** initGetInput() buffer = NULL	NullPointerException is thrown	
2	**offset > buffer.length** initGetInkey() buffer = "Text" offset = 5	ArrayIndexOutOfBoundsException is thrown	
3	**offset < 0** initGetInkey() buffer = "Text" offset = -1	ArrayIndexOutOfBoundsException is thrown	
4	**length > buffer.length** initGetInkey() buffer = "Text" offset = 0 length = 5	ArrayIndexOutOfBoundsException is thrown	
5	**offset + length > buffer.length** initGetInkey() buffer = "Text" offset = 3 length = 2	ArrayIndexOutOfBoundsException is thrown	
6	**length < 0** initGetInkey() buffer = "Text" offset = 3 length = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Successful call, buffer is the whole buffer** initGetInkey() 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** initGetInkey() 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

Id	Description	API Expectation	APDU Expectation
10	**Successful call, buffer is part of a buffer with the first part** **Send the command** initGetInkey() 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
11	**Successful call, buffer is part of a buffer** **Send the command** initGetInkey() 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** initGetInkey() 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)** initGetInkey() 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)** initGetInkey() 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** initGetInkey() 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()	

Id	Description	API Expectation	APDU Expectation
18	**Successful call, buffer length = 7Eh** initGetInkey()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
19	**Successful call, buffer length = 7Fh** initGetInkey() 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** initGetInkey() 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'

5.2.4.5 Method send

Test Area Reference: Api_2_Pah_Send.

5.2.4.5.1 Conformance requirement

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

```
public byte send()
```

5.2.4.5.1.1 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.

5.2.4.5.1.2 Parameter errors

No requirements.

5.2.4.5.1.3 Context errors

- CRRC1: A ToolkitException.UNAVAILABLE_ELEMENT shall be thrown is the Result Comprehension 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 Comprehension 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 CAT Runtime Environment.
- CRRC4: A ToolkitException.COMMAND_NOT_ALLOWED shall be thrown if one parameter of the proactive command to be sent is not allowed by the CAT Runtime Environment.

5.2.4.5.2 Test area files

Test Source: Test_Api_2_Pah_Send.java.

Test Applet: Api_2_Pah_Send_1.java.

Cap File: api_2_pah_send.cap.

5.2.4.5.3 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 the CAT Runtime Environment test : Cre_Api_Tran (test case 1)
C1	15
C2	16
C3	checked in the CAT Runtime Environment test : Fwk_Pcs_Pcco (test case 1)
C4	checked in the CAT Runtime Environment test : Fwk_Pcs_Pcco (test cases 2 to 3)

5.2.4.5.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Build and send a DISPLAY TEXT command** qualifier = 00h dcs = 04h buffer = 'Text'		DISPLAY TEXT 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	

Id	Description	API Expectation	APDU Expectation
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 information)	Result of send() is 02h	
9	**Build and send a 7Fh byte command (DISPLAY TEXT)** qualifier = 00h dcs = 04h buffer = "UUUUU…" length = 73h		DISPLAY TEXT 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 Comprehension 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 Comprehension TLV Result TLV = 03 00	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown by send()	

5.2.4.6 Method getLength

Test Area Reference Api_2_Pah_Glen.

5.2.4.6.1 Conformance requirement

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

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

5.2.4.6.1.1 Normal execution

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

5.2.4.6.1.2 Parameter errors

No requirements.

5.2.4.6.1.3 Context errors

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

5.2.4.6.2 Test area files

Test Source: Test_Api_2_Pah_Glen.java

Test Applet: Api_2_Pah_Glen_1.java.

Cap File: api_2_pah_glen.cap.

5.2.4.6.3 Test coverage

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

5.2.4.6.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Clear the handler** `getLength()`	Result of getLength() is 0	
2	**Call the init() method** `getLength()`	Result of getLength() is 9	
3	**Call the initDisplayText() method, with buffer length = 240** `getLength()`	Result of getLength() is 253	
4	**Build a 7Fh Proactive Handler** `getLength()`	Result of getLength() is 7Fh	
5	**Build a 80h Proactive Handler** `getLength()`	Result of getLength() is 80h	

5.2.4.7 Method copy

Test Area Reference Api_2_Pah_Copy.

5.2.4.7.1 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
```

5.2.4.7.1.1 Normal execution

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

5.2.4.7.1.2 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 Comprehension TLV List, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.OUT_OF_TLV_BOUNDARIES.

5.2.4.7.1.3 Context errors

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

5.2.4.7.2 Test area files

Test Source: Test_Api_2_Pah_Copy.java.

Test Applet: Api_2_Pah_Copy_1.java.

Cap File: api_2_pah_copy.cap.

5.2.4.7.3 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

5.2.4.7.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**NULL as parameter to dstBuffer**	NullPointerException is thrown	
2	**Call the init() method**		

	DstOffset > dstBuffer.length
	`copy()`
	`dstBuffer.length = 5`
	`dstOffset = 6`
	`dstLength = 0`
		ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
3	**dstOffset < 0** copy() dstBuffer.length = 5 dstOffset = -1 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**DstLength > dstBuffer.length** copy() dstBuffer.length = 5 dstOffset = 0 dstLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**dstOffset + dstLength > dstBuffer.length** copy() dstBuffer.length = 5 dstOffset = 3 dstLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength < 0** copy() dstBuffer.length = 5 dstOffset = 0 dstLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**dstLength > length of the Comprehension TLV list** copy() dstBuffer.length = 10 dstOffset = 0 dstLength = 10	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**Successful call, dstBuffer is the whole buffer** copy() 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** copy() 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** copy() dstBuffer.length = 15 dstOffset = 3 dstLength = 6	Result of copy() is 9	
13	**Compare the whole buffer**	Result of arrayCompare() is 0	

5.2.4.8 Method findTLV

Test Area Reference Api_2_Pah_Find.

5.2.4.8.1 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
```

5.2.4.8.1.1 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.

5.2.4.8.1.2 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.

5.2.4.8.1.3 Context errors

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

5.2.4.8.2 Test area files

Test Source: Test_Api_2_Pah_Find.java.

Test Applet: Api_2_Pah_Find_1.java.

Cap File: api_2_pah_find.cap.

5.2.4.8.4 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

5.2.4.8.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialize the handler**		

	Invalid input parameter
	`findTLV()`
	`Occurrence = 0`
		ToolkitException.BAD_INPUT_PARAMETER is thrown	

2	**Call the init() method**		

	Search 1st TLV
	`findTLV()`
	`Tag = 01h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_SET	

3	**Call the getValueLength() method**	Result is 03h	

4	**Search 2nd TLV**
	`findTLV()`
	`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
	`findTLV()`
	`Tag = 03h`
	`Occurrence = 1`
		Result is TLV_NOT_FOUND	

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

Id	Description	API Expectation	APDU Expectation

8	**Search a tag with wrong occurrence**
	`findTLV()`
	`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
	`findTLV()`
	`Tag = 02h`
	`Occurrence = 2`
		Result is TLV_FOUND_CR_NOT_SET	

11	**Append a TLV with tag=04h**		

	Search the TLV
	`findTLV()`
	`Tag = 04h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_NOT_SET	

12	**Search tag 81h**
	`findTLV()`
	`Tag = 81h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_SET	

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

5.2.4.9 Method getValueLength

Test Area Reference Api_2_Pah_Gvle.

5.2.4.9.1 Conformance requirement

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

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

5.2.4.9.1.1 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.

5.2.4.9.1.2 Parameter errors

No requirements.

5.2.4.9.1.3 Context errors

- CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.HANDLER_NOT_AVAILABLE.
- CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.UNAVAILABLE_ELEMENT.

5.2.4.9.2 Test area files

Test Source: Test_Api_2_Pah_Gvle.java.

Test Applet: Api_2_Pah_Gvle_1.java.

Cap File: api_2_pah_gvle.cap.

5.2.4.9.3 Test coverage

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

5.2.4.9.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the init() method**		
	`getValueLength()`	ToolkitException.UNAVAILABLE_ELEMENT is thrown	

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	

5.2.4.10 Method getValueByte

Test Area Reference Api_2_Pah_Gvby.

5.2.4.10.1 Conformance requirement

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

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

5.2.4.10.1.1 Normal execution

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

5.2.4.10.1.2 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.

5.2.4.10.1.3 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.

5.2.4.10.2 Test area files

Test Source: Test_Api_2_Pah_Gvby.java.

Test Applet: Api_2_Pah_Gvby_1.java.

Cap File: api_2_pah_gvby.cap.

5.2.4.10.3 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

5.2.4.10.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the init() method** type = FFh qualifier = FEh destination = FDh		
	getValueByte(0)	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	**Search TLV 01h (Command Details TLV)**		
	getValueByte(3)	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
3	**Search TLV 01h (Command Details TLV)**		
	getValueByte(2)	Result is FEh (qualifier)	
4	**Search TLV 02h (Device Identities TLV)**		
	getValueByte(0)	Result is 81h (Source)	
5	**initDisplayText()** buffer = 00 01 … 7D length = 7Eh Search TLV 0Dh (Text String TLV)		
	getValueByte(7E)	Result is 7Dh	
6	**initDisplayText()** buffer = 00 01 … 7D 7E length = 7Fh Search TLV 0Dh (Text String TLV)		
	getValueByte(7E)	Result is 7Dh	
7	**getValueByte(7F)**	Result is 7Eh	
8	**initDisplayText()** buffer = 00 01 … EF length = F0h Search TLV 0Dh (Text String TLV)		
	getValueByte(F0)	Result is EFh	

5.2.4.11 Method copyValue

Test Area Reference Api_2_Pah_Cpyv

5.2.4.11.1 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
```

5.2.4.11.1.1 Normal execution

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

5.2.4.11.1.2 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.

5.2.4.11.1.3 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.

5.2.4.11.2 Test area files

Test Source: Test_Api_2_Pah_Cpyv.java.

Test Applet: Api_2_Pah_Cpyv_1.java.

Cap File: api_2_pah_cpyv.cap.

5.2.4.11.3 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

5.2.4.11.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialize 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 copyValue() dstBuffer.length = 5 dstOffset = 6 dstLength = 0	ArrayIndexOutOfBoundsException is thrown	
3	**dstOffset < 0** copyValue() dstBuffer.length = 5 dstOffset = -1 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**dstLength >dstBuffer.length** copyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**dstOffset + dstLength >dstBuffer.length** copyValue() dstBuffer.length = 5 dstOffset = 3 dstLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength < 0** copyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**initDisplayText() with length = 5** Select Text String TLV		
	valueOffset > Text String Length copyValue() valueOffset = 7 dstBuffer.length = 15 dstOffset = 0 dstLength = 0	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**[Select Text String TLV]** valueOffset < 0 copyValue() 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 copyValue() 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 copyValue() valueOffset = 2 dstBuffer.length = 15 dstOffset = 0 dstLength = 5	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
11	**Initialize the handler**		
	copyValue()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	

Id	Description	API Expectation	APDU Expectation
12	**initDisplayText()** dcs = 4 buffer = 00 01 … 0F Select Text String TLV		
	Successful call copyValue() 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	**Initialize dstBuffer** dstBuffer = 55 55 … 55		
	Successful call copyValue() 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	

5.2.4.12 Method compareValue

Test Area Reference Api_2_Pah_Cprv.

5.2.4.12.1 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
```

5.2.4.12.1.1 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 Comprehension TLV List is less than that in compareBuffer.
- CRRN3: returns 1 if the first miscomparing byte in Comprehension TLV List is greater than that in compareBuffer.

5.2.4.12.1.2 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.

5.2.4.12.1.3 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.

5.2.4.12.2 Test area files

Test Source: Test_Api_2_Pah_Cprv.java.

Test Applet: Api_2_Pah_Cprv_1.java.

Cap File: api_2_pah_cprv.cap.

5.2.4.12.3 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

5.2.4.12.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialize 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
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 6`
	`compareLength = 0`
		ArrayIndexOutOfBoundsException is thrown	

3	**compareOffset < 0**
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = -1`
	`compareLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

4	**compareLength >compareBuffer.length**
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 0`
	`compareLength = 6`
		ArrayIndexOutOfBoundsException is thrown	

5	**compareOffset + compareLength >compareBuffer.length**
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 3`
	`compareLength = 3`
		ArrayIndexOutOfBoundsException is thrown	

6	**compareLength < 0** `compareValue()` compareBuffer.length = 5 compareOffset = 0 compareLength = -1	ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
7	**initDisplayText() with length = 5** **Select Text String TLV**		
	valueOffset > Text String Length compareValue() valueOffset = 7 compareBuffer.length = 15 compareOffset = 0 compareLength = 0	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**[Select Text String TLV]** valueOffset < 0 compareValue() 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 compareValue() 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 compareValue() valueOffset = 2 compareBuffer.length = 15 compareOffset = 0 compareLength = 5	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
11	**Initialize the handler**		
	compareValue()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
12	**initDisplayText()** dcs = 4 buffer = 00 01 … 0F Select Text String TLV		
	Initialize compareBuffer compareBuffer = 04 00 01 … 0F		
	Compare buffers compareValue() valueOffset = 0 compareOffset = 0 compareLength = 17	Result is 00h	
13	**Initialize 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	**Initialize compareBuffer** compareBuffer = 03 00 01 … 0F		
	Compare buffers with same parameters	Result is +1	
15	**Initialize compareBuffer** compareBuffer = 55 55 55 01 02 03 04 05 06 07 08 09 0A 0B 0C 55 55 55 55 55		
	Compare buffers compareValue() valueOffset = 2 compareOffset = 3 compareLength = 12	Result is 00h	

Id	Description	API Expectation	APDU Expectation
16	**Initialize 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	**Initialize 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	**Initialize 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	

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

Test Area Reference Api_2_Pah_Facyb_Bs.

5.2.4.13.1 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
```

5.2.4.13.1.1 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.

5.2.4.13.1.2 Parameter errors

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

5.2.4.13.1.3 Context errors

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

5.2.4.13.2 Test area files

Test Source: Test_Api_2_Pah_Facyb_Bs.java.

Test Applet: Api_2_Pah_Facyb_Bs_1.java.

Cap File: api_2_pah_facyb_bs.cap.

5.2.4.13.3 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

5.2.4.13.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialize the handler**		
	FindAndCopyValue() with a null dstBuffer	NullPointerException is thrown	
2	**InitDisplayText() with length = 15**		

	dstOffset > dstBuffer.length
	`findAndCopyValue()`
	`tag = 0Dh`
	`dstBuffer.length = 20`
	`dstOffset = 21`
		ArrayIndexOutOfBoundsException is thrown	

3	**dstOffset < 0**
	`findAndCopyValue()`
	`dstBuffer.length = 20`
	`dstOffset = -1`
		ArrayIndexOutOfBoundsException is thrown	

4	**length > dstBuffer.length**
	`findAndCopyValue()`
	`dstBuffer.length = 15`
	`dstOffset = 0`
		ArrayIndexOutOfBoundsException is thrown	

5	**DstOffset + length >dstBuffer.length**
	`findAndCopyValue()`
	`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
	`findAndCopyValue()`
	`Tag = 0Dh`
	`DstBuffer.length = 17`
	`DstOffset = 0`
		Result of findAndcopyValue() is 17	

8	**Compare buffer** `buffer = 04 00 01 … 0F`	Result is 00h	
9	**Initialize dstBuffer** `dstBuffer = 55 55 … 55`		

	Successful call
	`findAndCopyValue()`
	`dstBuffer.length = 20`
	`dstOffset = 2`
		Result of findAndcopyValue() is 19	

Id	Description	API Expectation	APDU Expectation
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 findAndCopyValue() 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) findAndCopyValue() 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) findAndCopyValue() tag = 8Fh dstBuffer.length = 16 dstOffset = 0	Result of findAndcopyValue() is 16	
16	**Compare buffer** buffer = 00 01 … 0F	Result is 00h	

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

Test Area Reference Api_2_Pah_Facybbs_Bss.

5.2.4.14.1 Conformance requirement

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

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

5.2.4.14.1.1 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.

5.2.4.14.1.2 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.
- 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.

5.2.4.14.1.3 Context errors

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

5.2.4.14.2 Test area files

Test Source: Test_Api_2_Pah_Facybbs_Bss.java.

Test Applet: Api_2_Pah_Facybbs_Bss_1.java.

Cap File: api_2_pah_facybbs_bss.cap.

5.2.4.14.3 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
P4	24
C1	Does not apply for ProactiveHandler

5.2.4.14.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialize the handler**		
	findAndCopyValue() with a null dstBuffer	NullPointerException is thrown	
2	**initDisplayText() with length = 15**		
	dstOffset > dstBuffer.length findAndCopyValue() tag = 0Dh, occurrence = 1 valueOffset = 0 dstBuffer.length = 5 dstOffset = 6 dstLength = 0	ArrayIndexOutOfBoundsException is thrown	
3	**dstOffset < 0** findAndCopyValue() dstBuffer.length = 5 dstOffset = -1 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
4	**dstLength >dstBuffer.length** findAndCopyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**dstOffset + dstLength >dstBuffer.length** findAndCopyValue() dstBuffer.length = 5 dstOffset = 3 dstLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength < 0** findAndCopyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**initDisplayText() with length = 5**		
	valueOffset > Text String Length findAndCopyValue() tag = 0Dh, occurrence = 1 valueOffset = 7 dstBuffer.length = 15 dstOffset = 0 dstLength = 0	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**valueOffset < 0** findAndCopyValue() valueOffset = -1 dstBuffer.length = 15 dstOffset = 0 dstLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
9	**dstLength > Text String length** findAndCopyValue() valueOffset = 0 dstBuffer.length = 15 dstOffset = 0 dstLength = 7	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
10	**valueOffset + dstLength > Text String length** findAndCopyValue() valueOffset = 2 dstBuffer.length = 15 dstOffset = 0 dstLength = 5	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
11	**InitDisplayText()**		
	Select a TLV (tag 02h)		
	findAndCopyValue() tag = 0Dh occurrence = 2	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the getValueLength() method	ToolkitException.UNAVAILABLE_ELEMENT is thrown.	
12	**initDisplayText()** dcs = 4 buffer = 00 01 … 0F		
	Successful call findAndCopyValue() 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	**Initialize dstBuffer** dstBuffer = 55 55 … 55		
	Successful call findAndCopyValue() tag = 0Dh, occurrence = 1 valueOffset = 2 dstBuffer.length = 20 dstOffset = 3 dstLength = 12	Result of findAndcopyValue() is 15	

Id	Description	API Expectation	APDU Expectation
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 findAndCopyValue() 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** findAndCopyValue() 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) findAndCopyValue() 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) findAndCopyValue() 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	**Invalid parameter** findAndCopyValue() occurrence = 0	ToolkitException.BAD_INPUT_PARAMETER is thrown	

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

Test Area Reference Api_2_Pah_Facrb_Bs.

5.2.4.15.1 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
```

5.2.4.15.1.1 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 Comprehension TLV is less than that in compareBuffer returns -1.
- CRRN5: if the first miscomparing byte in Comprehension TLV is greater than that in compareBuffer returns 1.
- CRRN6: The search method is comprehension required flag independent.

5.2.4.15.1.2 Parameter errors

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

5.2.4.15.1.3 Context errors

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

5.2.4.15.2 Test area files

Test Source: Test_Api_2_Pah_Facrb_Bs.java.

Test Applet: Api_2_Pah_Facrb_Bs_1.java.

Cap File: api_2_pah_facrb_bs.cap.

5.2.4.15.3 Test coverage

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

5.2.4.15.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialize the handler**		
	findAndCompareValue() with a null dstBuffer	NullPointerException is thrown	
2	**initDisplayText() with length = 15**		
	compareOffset > compareBuffer.length findAndCompareValue() tag = 0Dh compareBuffer.length = 20 compareOffset = 21	ArrayIndexOutOfBoundsException is thrown	
3	**compareOffset < 0** findAndCompareValue() compareBuffer.length = 20 compareOffset = -1	ArrayIndexOutOfBoundsException is thrown	
4	**length > compareBuffer.length** findAndCompareValue() compareBuffer.length = 15 compareOffset = 0	ArrayIndexOutOfBoundsException is thrown	
5	**compareOffset + length > compareBuffer.length** findAndCompareValue() 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		
	Initialize compareBuffer compareBuffer = 04 00 01 … 0F		
	Compare buffers findAndCompareValue() tag = 0Dh compareOffset = 0	Result is 00h	
8	**Verify current TLV** getValueLength()	Result is 17	
9	**Initialize compareBuffer** compareBuffer = 04 00 01 … 10		
	Compare buffers with same parameters	Result is -1	
10	**Initialize compareBuffer** compareBuffer = 03 00 01 … 0F		
	Compare buffers with same parameters	Result is +1	
11	**Initialize compareBuffer** compareBuffer = 55 55 04 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 55		
	Compare buffers findAndCompareValue() compareOffset = 2	Result is 00h	
12	**append a Text String TLV** tag = 0Dh buffer = 00 11 22 33 44 55		
	Initialize compareBuffer compareBuffer = 55 55 04 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 55		

Id	Description	API Expectation	APDU Expectation
	Compare buffers findAndCompareValue() compareOffset = 2	Result is 00h	
13	**Initialize compareBuffer** compareBuffer = 55 55 04 01 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 55		
	Compare buffers findAndCompareValue() compareOffset = 2	Result is -1	
14	**Initialize compareBuffer** compareBuffer = 55 55 04 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0D 10 55		
	Compare buffers findAndCompareValue() compareOffset = 2	Result is +1	
15	**initDisplayText()** dcs = 4 buffer = 00 01 … 0F		
	Initialize compareBuffer CompareBuffer = 04 00 01 … 0F		
	Successful call (with tag 8Dh) findAndCompareValue() tag = 8Dh compareBuffer.length = 17 compareOffset = 0	Result is 00h	
16	**Append tag 0Fh** buffer = 00 01 … 0F		
	Initialize compareBuffer compareBuffer = 00 01 … 0F		
	Successful call (with tag 8Fh) findAndCompareValue() tag = 8Fh compareBuffer.length = 16 compareOffset = 0	Result is 00h	
17	**Initialize compareBuffer** compareBuffer = 00 99 01 03 … 0F		
	Successful call (with tag 8Fh) findAndCompareValue() tag = 8Fh compareBuffer.length = 16 compareOffset = 0	Result is +1	

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

Test Area Reference Api_2_Pah_Facrbbs_Bss.

5.2.4.16.1 Conformance requirement

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

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

5.2.4.16.1.1 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 Comprehension TLV is less than that in compareBuffer -1 is returned.
- CRRN5: if the first miscomparing byte in Comprehension TLV is greater than that in compareBuffer 1 is returned.
- CRRN6: The search method is comprehension required flag independent.

5.2.4.16.1.2 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.

5.2.4.16.1.3 Context errors

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

5.2.4.16.2 Test area files

Test Source: Test_Api_2_Pah_Facrbbs_Bss.java.

Test Applet: Api_2_Pah_Facrbbs_Bss_1.java.

Cap File: api_2_pah_facrbbs_bss.cap.

5.2.4.16.3 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

5.2.4.16.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialize the handler**		
	findAndCompareValue() with a null compareBuffer	NullPointerException is thrown	
2	**initDisplayText() with length = 15**		
	compareOffset > compareBuffer.length findAndCompareValue() tag = 0Dh, occurrence = 1 valueOffset = 0 compareBuffer.length = 5 compareOffset = 6 compareLength = 0	ArrayIndexOutOfBoundsException is thrown	
3	**compareOffset < 0** findAndCompareValue() compareBuffer.length = 5 compareOffset = -1 compareLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**compareLength >compareBuffer.length** findAndCompareValue() compareBuffer.length = 5 compareOffset = 0 compareLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**compareOffset + compareLength >compareBuffer.length** findAndCompareValue() compareBuffer.length = 5 compareOffset = 3 compareLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**compareLength < 0** findAndCompareValue() compareBuffer.length = 5 compareOffset = 0 compareLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**initDisplayText() with length = 5**		
	valueOffset > Text String Length findAndCompareValue() tag = 0Dh, occurrence = 1 valueOffset = 7 compareBuffer.length = 15 compareOffset = 0 compareLength = 0	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**valueOffset < 0** findAndCompareValue() valueOffset = -1 compareBuffer.length = 15 compareOffset = 0 compareLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
9	**compareLength > Text String length** findAndCompareValue() valueOffset = 0 compareBuffer.length = 15 compareOffset = 0 compareLength = 7	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
10	**valueOffset + compareLength > Text String length** findAndCompareValue() valueOffset = 2 compareBuffer.length = 15 compareOffset = 0 compareLength = 5	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
11	**Invalid parameter** findAndCompareValue() 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		
	Initialize 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	**Initialize compareBuffer** compareBuffer = 04 00 01 … 10		
	Compare buffers with same parameters	Result is -1	
16	**Initialize compareBuffer** compareBuffer = 03 00 01 … 0F		
	Compare buffers with same parameters	Result is +1	
17	**Initialize compareBuffer** compareBuffer = 55 55 55 01 02 03 04 05 06 07 08 09 0A 0B 0C 55 55 55 55 55		
	Compare buffers findAndCompareValue() valueOffset = 2 compareOffset = 3 compareLength = 12	Result is 00h	
18	**Initialize 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	**Initialize 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		
	Initialize 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	**Initialize compareBuffer** compareBuffer = 00 11 22 33 44 55		
	findAndCompareValue() tag = 0Dh, occurrence = 2 valueOffset = 0 compareOffset = 0 compareLength = 6	Result is 00h	
22	**Initialize 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		
	Initialize compareBuffer CompareBuffer = 04 00 01 … 0F		
	Successful call (with tag 8Dh) findAndCompareValue() tag = 8Dh, occurrence = 1 valueOffset = 0 compareBuffer.length = 17 compareOffset = 0 compareLength = 17	Result is 00h	
24	**Append tag 0Fh** buffer = 00 01 … 0F		
	Initialize compareBuffer compareBuffer = 00 01 … 0F		
	Successful call (with tag 8Fh) findAndCompareValue() tag = 8Fh, occurrence = 1 valueOffset = 0 compareBuffer.length = 16 compareOffset = 0 compareLength = 16	Result is 00h	
25	**Initialize compareBuffer** compareBuffer =0099 02 … 0F		
	findAndCompareValue() findAndCompareValue() tag = 0Dh, occurrence = 1 valueOffset = 0 compareOffset = 0 compareLength = 17	Result is +1	

5.2.4.17 Method appendArray

Test Area Reference: Api_2_Pah_Apda.

5.2.4.17.1 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
```

5.2.4.17.1.1 Normal execution

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

5.2.4.17.1.2 Parameter errors

- 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.

5.2.4.17.1.3 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.

5.2.4.17.2 Test area files

Test Source: Test_Api_2_Pah_Apda.java.

Test Applet: Api_2_Pah_Apda_1.java.

Cap File: api_2_pah_apda.cap.

5.2.4.17.3 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

5.2.4.17.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Null buffer** `appendArray()`	NullPointerException is thrown	

2	**offset > buffer.length**
	`appendArray()`
	`buffer.length = 5`
	`offset = 6`
	`length = 0`
		ArrayIndexOutOfBoundsException is thrown	

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

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

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

Id	Description	API Expectation	APDU Expectation
6	**length < 0** appendArray() buffer.length = 5 offset = 0 length = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Handler overflow exception** appendArray() buffer.length = getCapacity()+1 offset = 0 length = getCapacity()+1	ToolkitException.HANDLER_OVERFLOW is thrown	
8	**Initialize handler**		
	Select Command Details TLV		
	Successful call appendArray() buffer = FF FE … F8 offset = 0 length = 8		
	Verify Current TLV: Call getValueLength()	Result is 03h	
9	**Clear the handler**		
	Successful call appendArray() 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** appendArray() 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** appendArray() 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 appendArray() 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	

5.2.4.18 Method appendTLV(byte tag, byte value)

Test Area Reference: Api_2_Pah_Aptlbb.

5.2.4.18.1 Conformance requirement:

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

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

5.2.4.18.1.1 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.

5.2.4.18.1.2 Parameter errors

No requirements.

5.2.4.18.1.3 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.

5.2.4.18.2 Test area files

Test Source: Test_Api_2_Pah_Aptlbb.java.

Test Applet: Api_2_Pah_Aptlbb_1.java.

Cap File: api_2_pah_aptlbb.cap.

5.2.4.18.3 Test coverage

CRR number	Test case number
N1	3, 4, 5
N2	2
C1	1
C2	Does not apply for Proactive Handler

5.2.4.18.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call appendArray()** `length = getCapacity()-1`		
	Handler Overflow exception: Call the appendTLV() method	ToolkitException.HANDLER_OVERFLOW is thrown	
2	**Initialize handler**		
	Select Command Details TLV		
	Call the appendTLV() method		
	Verify Current TLV: Call getValueLength()	Result is 03h	

Id	Description	API Expectation	APDU Expectation
3	**Clear the handler**		
	Successful call appendTLV() 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** appendTLV() 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	
5	**Clear the handler** **Call appendArray()** length = 250 buffer = 00 81 F7 03 04 … F9 **Successful call** appendTLV() 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	

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

Test Area Reference: Api_2_Pah_Aptlbbb.

5.2.4.19.1 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
```

5.2.4.19.1.1 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.

5.2.4.19.1.2 Parameter errors

No requirements.

5.2.4.19.1.3 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.

5.2.4.19.2 Test area files

Test Source: Test_Api_2_Pah_Aptlbbb.java.

Test Applet: Api_2_Pah_Aptlbbb_1.java.

Cap File: api_2_pah_aptlbbb.cap.

5.2.4.19.3 Test coverage

CRR number	Test case number
N1	3, 4, 5
N2	2
C1	1
C2	Does not apply for Proactive Handler

5.2.4.19.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the appendArray()** `length = getCapacity()-1`		
	Handler Overflow exception: **Call the appendTLV() method**	ToolkitException.HANDLER_OVERFLOW is thrown	
2	**Initialize handler**		
	Select Command Details TLV		
	Call the appendTLV() method		
	Verify Current TLV: Call getValueLength()	Result is 03h	
3	**Clear the handler**		

	Successful call
	`appendTLV()`
	`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**
	`appendTLV()`
	`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
	`appendTLV()`
	`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	

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

Test Area Reference: Api_2_Pah_Aptlb_Bss.

5.2.4.20.1 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
```

5.2.4.20.1.1 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.

5.2.4.20.1.2 Parameter errors

- 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.

5.2.4.20.1.3 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.
- CRRC3: if valuelength is greater than 255, a ToolkitException is thrown with reason code BAD_INPUT_PARAMETER.

5.2.4.20.2 Test area files

Test Source: Test_Api_2_Pah_Aptlb_Bss.java.

Test Applet: Api_2_Pah_Aptlb_Bss_1.java.

Cap File: api_2_pah_aptlb_bss.cap.

5.2.4.20.3 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

5.2.4.20.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Null value** appendTLV()	NullPointerException is thrown	
2	**valueOffset > value.length** appendTLV() value.length = 5 valueOffset = 6 valueLength = 0	ArrayIndexOutOfBoundsException is thrown	
3	**valueOffset < 0** appendTLV() value.length = 5 valueOffset = -1 valueLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**valueLength > value.length** appendTLV() value.length = 5 valueOffset = 0 valueLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**valueOffset + valueLength > value.length** appendTLV() value.length = 5 valueOffset = 3 valueLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**valueLength < 0** appendTLV() value.length = 5 valueOffset = 0 valueLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Handler overflow exception** Call the appendArray() method, length = getCapacity()-1 appendTLV() value.length = 254 valueOffset = 0 valueLength = 254	ToolkitException.HANDLER_OVERFLOW is thrown	
8	**Bad parameter exception** Clear the handler appendTLV() value.length = 256 valueOffset = 0 valueLength = 256	ToolkitException.BAD_INPUT_PARAMETER is thrown	
9	**Initialize handler**		
	Select Command Details TLV		
	Successful call appendTLV() tag = 04 value = FF FE … F8 valueOffset = 0 valueLength = 8		
	Verify Current TLV: Call getValueLength()	Result is 03h	
10	**Clear the handler**		
	Successful call appendTLV() 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** appendTLV() 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	

Id	Description	API Expectation	APDU Expectation
12	**Successful call** appendTLV() 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 appendTLV() 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	
14	**Clear the handler** **Successful call** appendTLV() 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	

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

Test Area Reference: Api_2_Pah_Aptlbb_Bss.

5.2.4.21.1 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
```

5.2.4.21.1.1 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.

5.2.4.21.1.2 Parameter errors

- 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.

5.2.4.21.1.3 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.

5.2.4.21.2 Test area files

Test Source: Test_Api_2_Pah_Aptlbb_Bss.java.

Test Applet: Api_2_Pah_Aptlbb_Bss_1.java.

Cap File: api_2_pah_aptlbb_bss.cap.

5.2.4.21.3 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

5.2.4.21.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Null value2** `appendTLV()`	NullPointerException is thrown	

2	**value2Offset > value2.length**
	`appendTLV()`
	`value2.length = 5`
	`value2Offset = 6`
	`value2Length = 0`
		ArrayIndexOutOfBoundsException is thrown	

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

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

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

Id	Description	API Expectation	APDU Expectation
6	**value2Length < 0** appendTLV() value2.length = 5 value2Offset = 0 value2Length = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Handler overflow exception** Call the appendArray() method, length = getCapacity()-1 appendTLV() value2.length = 254 value2Offset = 0 value2Length = 254	ToolkitException.HANDLER_OVERFLOW is thrown	
8	**Bad parameter exception** Clear the handler appendTLV() value2.length = 256 value2Offset = 0 value2Length = 256	ToolkitException.BAD_INPUT_PARAMETER is thrown	
9	**Initialize handler**		
	Select Command Details TLV		
	Successful call appendTLV() 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 appendTLV() 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	
11	**Successful call** appendTLV() 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** appendTLV() 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	

Id	Description	API Expectation	APDU Expectation
13	**Clear the handler**		
	Successful call appendTLV() 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 appendTLV() 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	

5.2.4.22 Method clear

Test Area Reference: Api_2_Pah_Cler.

5.2.4.22.1 Conformance requirement

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

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

5.2.4.22.1.1 Normal execution

- CRRN1: Clears the TLV list of an EditHandler.
- CRRN2: Resets the current TLV selected.

5.2.4.22.1.2 Parameter errors

No requirements.

5.2.4.22.1.3 Context errors

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

5.2.4.22.2 Test area files

Test Source: Test_Api_2_Pah_Cler.java.

Test Applet: Api_2_Pah_Cler_1.java.

Cap File: api_2_pah_cler.cap.

5.2.4.22.3 Test coverage

CRR number	Test case number
N1	1
N2	2
C1	Does not apply for Proactive Handler

5.2.4.22.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Initialize 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	

5.2.4.23 Method getCapacity

Test Area Reference: Api_2_Pah_Gcap.

5.2.4.23.1 Conformance requirement

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

```
public byte getCapacity()
```

5.2.4.23.1.1 Normal execution

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

5.2.4.23.1.2 Parameter errors

No requirements.

5.2.4.23.1.3 Context errors

No requirements.

5.2.4.23.2 Test area files

Test Source: Test_Api_2_Pah_Gcap.java.

Test Applet: Api_2_Pah_Gcap_1.java.

Cap File: api_2_pah_gcap.cap.

5.2.4.23.3 Test coverage

CRR number	Test case number
N1	1

5.2.4.23.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**ProactiveHandler available** 1- Send envelope Menu Selection 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	

5.2.4.24 Method initCloseChannel

Test Area Reference: Api_2_Pah_Icch.

5.2.4.24.1 Conformance requirement

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

```
public void initCloseChannel(byte bChannelIdentifier)
```

5.2.4.24.1.1 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 initializes 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.

5.2.4.24.1.2 Parameter errors

No requirements.

5.2.4.24.1.3 Context errors

No requirements.

5.2.4.24.2 Test area files

Test Source: Test_Api_2_Pah_Icch.java.

Test Applet: Api_2_Pah_Icch_1.java.

Cap File: api_2_pah_icch.cap.

5.2.4.24.3 Test coverage

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

5.2.4.24.4 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 ProactiveHandler.send() method. 2- Send an EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS Envelope. 3- Call ProactiveHandler.initCloseChannel() method with Channel Id = 01. 4- Call 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 UICC with Channel Id = 01. 4- CLOSE CHANNEL proactive command is fetched. TERMINAL RESPONSE of CLOSE CHANNEL is sent to the UICC.
2	**Call the initCloseChannel () method with any value then build and send a CLOSE CHANNEL command** 1- Call ProactiveHandler.init() method to open a Channel and ProactiveHandler.send() method. 2- Call ProactiveHandler.initCloseChannel() method with Channel Id = 2 3- Call ProactiveHandler.initCloseChannel() method with the Channel Id = 1. 4- Call 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 UICC with Channel Id = 01. 4- CLOSE CHANNEL proactive command is fetched. TERMINAL RESPONSE of CLOSE CHANNEL is sent to the UICC.
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 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 UICC with Channel Id = 01. 4- CLOSE CHANNEL proactive command is fetched. TERMINAL RESPONSE of CLOSE CHANNEL is sent to the UICC.

Id	Description	API Expectation	APDU Expectation
4	**Call the initCloseChannel() without sending the command** 1- Call ProactiveHandler.init() method to open a Channel and call the ProactiveHandler.send() method. 2- Call ProactiveHandler.initCloseChannel() method with Channel Id = 01 without ProactiveHandler.send(). 3- Send an EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS Envelope.	3- Applet1 is triggered.	1- OPEN CHANNEL proactive command is fetched. TERMINAL RESPONSE of OPEN CHANNEL is sent to the UICC with Channel Id = 01. No proactive command shall be sent. Expected status is '9000'

5.2.4.25 Method getValueShort

Test Area Reference: Api_2_Pah_Gvsh.

5.2.4.25.1 Conformance requirement

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

```
public short getValueShort(short valueOffset)
                    throws ToolkitException
```

5.2.4.25.1.1 Normal execution

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

5.2.4.25.1.2 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.

5.2.4.25.1.3 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.

5.2.4.25.2 Test area files

Specific triggering: None

Test Source: Test_Api_2_Pah_Gvsh.java.

Test Applet: Api_2_Pah_Gvsh_1.java.

Cap File: api_2_pah_gvsh.cap.

5.2.4.25.3 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

5.2.4.25.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the init() method** type = FFh qualifier = FEh destination = FDh		
	getValueShort(0)	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	**Search TLV 01h (Command Details TLV)**		
	getValueShort(3)	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
3	**Search TLV 01h (Command Details TLV)**		
	getValueShort(1)	Result is FFh FEh (type, qualifier)	
4	**Search TLV 02h (Device Identities TLV)**		
	getValueShort(0)	Result is 81h FDh (Source, Destination)	
5	**initDisplayText()** buffer = 00 01 … 7D length = 7Eh **Search TLV 0Dh (Text String TLV)**		
	getValueShort(7D)	Result is 7Ch 7Dh	
6	**initDisplayText()** buffer = 00 01 … 7D 7E length = 7Fh **Search TLV 0Dh (Text String TLV)**		
	getValueShort(7D)	Result is 7Ch 7Dh	
7	**getValueShort(7E)**	Result is 7Dh 7Eh	
8	**initDisplayText()** buffer = 00 01 … EF length = F0h **Search TLV 0Dh (Text String TLV)**		
	getValueShort(EF)	Result is EEh EFh	

5.2.4.26 Method appendTLV(byte tag, byte value1, short value2)

Test Area Reference: Api_2_Pah_Aptlbbs.

5.2.4.26.1 Conformance requirements

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

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

5.2.4.26.1.1 Normal execution

- CRRN1: Appends a TLV element to the current TLV list (3-byte element(1-byte,1-short)).
- CRRN2: A successful append does not modify the TLV selected.

5.2.4.26.1.2 Parameter errors

No requirements.

5.2.4.26.1.3 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.

5.2.4.26.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Pah_Aptlbbs.java.

Test Applet: Api_2_Pah_Aptlbbs_1.java.

Cap File: api_2_pah_aptlbbs.cap.

5.2.4.26.3 Test coverage

CRR number	Test case number
N1	3, 4, 5
N2	2
C1	1
C2	Does not apply for Proactive Handler

5.2.4.26.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the appendArray()** `length = getCapacity()-1`		
	Handler Overflow exception: Call the appendTLV() method	ToolkitException.HANDLER_OVERFLOW is thrown	
2	**Initialize handler**		
	Select Command Details TLV		
	Call the appendTLV() method		
	Verify Current TLV: Call getValueLength()	Result is 03h	
3	**Clear the handler**		

	Successful call
	`appendTLV()`
	`tag = 84h`
	`value1 = 00h`
	`value2 = 01h 02h`		

	Call copy() method		
	Compare the arrays `compareBuffer = 84 03 00 01 02`	Result of javacard.framework.Util.arrayCompare() is 00h	

4	**Successful call**
	`appendTLV()`
	`tag = 01h`
	`value1 = FEh`
	`value2 = FDh FCh`		

	Call copy() method		
	Compare the arrays `compareBuffer = 84 03 00 01 02 01 03 FE FD FC`	Result of javacard.framework.Util.arrayCompare() is 00h	

Id	Description	API Expectation	APDU Expectation
5	**Clear the handler**		
	Call appendArray() length = 248 buffer = 00 81 F5 03 04 … F7		
	Successful call appendTLV() tag = 84h value1 = 00h value2 = 01h 02h		
	Call getLength() method	result = 253	
	Call copy() method		
	Compare handler compareBuffer = 00 81 F5 03 04 … F7 84 03 00 01 02	Result of javacard.framework.Util.arrayCompare() is 00h	

5.2.4.27 Method appendTLV(byte tag, short value)

Test Area Reference: Api_2_Pah_Aptlbs.

5.2.4.27.1 Conformance requirements

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

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

5.2.4.27.1.1 Normal execution

- CRRN1: Appends a TLV element to the current TLV list (2-byte or 1-short element).
- CRRN2: A successful append does not modify the TLV selected.

5.2.4.27.1.2 Parameter errors

No requirements.

5.2.4.27.1.3 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.

5.2.4.27.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Pah_Aptlbs.java.

Test Applet: Api_2_Pah_Aptlbs_1.java.

Cap File: api_2_pah_aptlbs.cap.

5.2.4.27.3 Test coverage

CRR number	Test case number
N1	3, 4, 5
N2	2
C1	1
C2	Does not apply for Proactive Handler

5.2.4.27.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call appendArray()** length = getCapacity()-1		
	Handler Overflow exception: **Call the appendTLV() method**	ToolkitException.HANDLER_OVERFLOW is thrown	
2	**Initialize handler**		
	Select Command Details TLV		
	Call the appendTLV() method		
	Verify Current TLV: Call getValueLength()	Result is 03h	
3	**Clear the handler**		
	Successful call appendTLV() tag = 84h value = 00h 01h		
	Call copy() method		
	Compare the arrays compareBuffer = 84 02 00 01	Result of javacard.framework.Util.arrayCompare() is 00h	
4	**Successful call** appendTLV() tag = 01h value = FEh FFh		
	Call copy() method		
	Compare the arrays compareBuffer = 84 02 00 01 01 02 FE FF	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** appendTLV() tag = 84h value = 00h 01h **Call getLength() method** **Call copy() method** **Compare the array** compareBuffer = 00 81 F6 03 04 … F8 84 02 00 01	result = 253 Result of javacard.framework.Util.arrayCompare() is 00h	

5.2.4.28 Method appendTLV(byte tag, short value1, short value2)

Test Area Reference: Api_2_Pah_Aptlbss.

5.2.4.28.1 Conformance requirements

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

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

5.2. 4.28.1.1 Normal execution

- CRRN1: Appends a TLV element to the current TLV list (4-byte element(2-short)).
- CRRN2: A successful append does not modify the TLV selected.

5.2.4.28.1.2 Parameter errors

No requirements.

5.2.4.28.1.3 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.

5.2.4.28.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Pah_Aptlbss.java.

Test Applet: Api_2_Pah_Aptlbss_1.java.

Cap File: api_2_pah_aptlbss.cap.

5.2.4.28.3 Test coverage

CRR number	Test case number
N1	3, 4, 5
N2	2
C1	1
C2	Does not apply for Proactive Handler

5.2.4.28.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call the appendArray()** `length = getCapacity()-1`		
	Handler Overflow exception: **Call the appendTLV() method**	ToolkitException.HANDLER_OVERFLOW is thrown	
2	**Initialize handler**		
	Select Command Details TLV		
	Call the appendTLV() method		
	Verify Current TLV: Call getValueLength()	Result is 03h	

Id	Description	API Expectation	APDU Expectation
3	**Clear the handler**		
	Successful call appendTLV() tag = 84h value1 = 00h 01h value2 = 02h 03h		
	Call copy() method		
	Compare the arrays compareBuffer = 84 04 00 01 02 03	Result of javacard.framework.Util.arrayCompare() is 00h	
4	**Successful call** appendTLV() tag = 01h value1 = FEh FDh value2 = FCh FBh		
	Call copy() method		
	Compare the arrays compareBuffer = 84 04 00 01 02 03 01 04 FE FD FC FB	Result of javacard.framework.Util.arrayCompare() is 00h	
5	**Clear the handler**		
	Call appendArray() length = 247 buffer = 00 81 F4 03 04 … F6		
	Successful call appendTLV() tag = 84h value1 = 00h 01h value2 = 02h 03h		
	Call getLength() method	result = 253	
	Call copy() method		
	Compare handler compareBuffer = 00 81 F4 03 04 … F6 84 04 00 01 02 03	Result of javacard.framework.Util.arrayCompare() is 00h	

5.2.4.29 Method appendTLV(byte tag, byte[] value1, short value1Offset, short value1Length, byte[] value2, short value2Offset, short value2Length)

Test Area Reference: Api_2_Pah_Aptlb_Bss_Bss.

5.2.4.29.1 Conformance requirements

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

```
public void appendTLV(byte tag,
                      byte[] value1,
                      short value1Offset,
                      short value1Length,
                      byte[] value2,
                      short value2Offset,
                      short value2Length)
               throws java.lang.NullPointerException,
                      java.lang.ArrayIndexOutOfBoundsException,
                      ToolkitException
```

5.2.4.29.1.1 Normal execution

- CRRN1: Appends a TLV element to the current TLV list (2 byte arrays format).
- CRRN2: A successful append does not modify the TLV selected.

5.2.4.29.1.2 Parameter errors

- CRRP1: If value1 or value2 is null, a NullPointerException is thrown.
- CRRP2: If value1Offset or value1Length or both would cause access outside value1 array bounds, or if value1Length is negative, an ArrayIndexOutOfBoundsException is thrown.
- CRRP3: If value2Offset or value2Length or both would cause access outside value2 array bounds, or if value2Length is negative, an ArrayIndexOutOfBoundsException is thrown.

5.2.4.29.1.3 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 value1Length or value2Length is greater than 255, a ToolkitException is thrown with reason code BAD_INPUT_PARAMETER.

5.2.4.29.2 Test area files

Test Source: Test_Api_2_Pah_Aptlb_Bss_Bss.java.

Test Applet: Api_2_Pah_Aptlb_Bss_Bss_1.java.

Cap File: api_2_pah_aptlb_bss_bss.cap.

5.2.4.29.3 Test coverage

CRR number	Test case number
N1	18, 19, 20, 21
N2	16
P1	1, 2
P2	3, 4, 5, 6, 7
P3	8, 9, 10, 11, 12
C1	13
C2	Does not apply for ProactiveHandler
C3	14, 15

5.2.4.29.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Null value1** `appendTLV()`	NullPointerException is thrown	
2	**Null value2** `appendTLV()`	NullPointerException is thrown	

3	**Value1Offset ≥ value1.length**
	`appendTLV()`
	`value1.length = 5`
	`value1Offset = 5`
	`value1Length = 1`
	`value2.length = 5`
	`value2Offset = 0`
	`value2Length = 1`
		ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
4	**Value1Offset < 0** appendTLV() value1.length = 5 value1Offset = -1 value1Length = 1 value2.length = 5 value2Offset = 0 value2Length = 1	ArrayIndexOutOfBoundsExceptio n is thrown	
5	**Value1Length > value1.length** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 6 value2.length = 5 value2Offset = 0 value2Length = 1	ArrayIndexOutOfBoundsExceptio n is thrown	
6	**Value1Offset + value1Length > value1.length** appendTLV() Value1.length = 5 value1Offset = 3 value1Length = 3 value2.length = 5 value2Offset = 0 value2Length = 1	ArrayIndexOutOfBoundsExceptio n is thrown	
7	**Value1Length < 0** appendTLV() value1.length = 5 value1Offset = 0 value1Length = -1 value2.length = 5 value2Offset = 0 value2Length = 1	ArrayIndexOutOfBoundsExceptio n is thrown	
8	**Value2Offset ≥ value2.length** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 value2.length = 5 value2Offset = 5 value2Length = 1	ArrayIndexOutOfBoundsExceptio n is thrown	
9	**Value2Offset < 0** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 value2.length = 5 value2Offset = -1 value2Length = 1	ArrayIndexOutOfBoundsExceptio n is thrown	
10	**Value2Length > value2.length** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 value2.length = 5 value2Offset = 0 value2Length = 6	ArrayIndexOutOfBoundsExceptio n is thrown	
11	**Value2Offset + value2Length > value2.length** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 Value2.length = 5 Value2Offset = 3 Value2Length = 3	ArrayIndexOutOfBoundsExceptio n is thrown	
12	**Value2Length < 0** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 value2.length = 5 value2Offset = 0 value2Length = -1	ArrayIndexOutOfBoundsExceptio n is thrown	

Id	Description	API Expectation	APDU Expectation
13	**Handler overflow exception** Call the appendArray() method, length = getCapacity()-1 appendTLV() Value1.length = 256 Value1Offset = 0 Value1Length = 253 Value2.length = 256 Value2Offset = 0 Value2Length = 1	ToolkitException.HANDLER_OVERFLOW is thrown	
14	**Bad parameter exception** Clear the handler appendTLV() Value1.length = 256 Value1Offset = 0 Value1Length = 256 Value2.length = 256 Value2Offset = 0 Value2Length = 1	ToolkitException.BAD_INPUT_PARAMETER is thrown	
15	**Bad parameter exception** appendTLV() Value1.length = 256 Value1Offset = 0 Value1Length = 1 Value2.length = 256 Value2Offset = 0 Value2Length = 256	ToolkitException.BAD_INPUT_PARAMETER is thrown	
16	**Successful call does not modify the current TLV** 1- clear the handler, append the handler with TLVs: 81 03 11 22 33 82 02 99 77 2- Select Command Details TLV by using the findTLV() method 3- Successful call of the AppendTLV() method tag = 04 value1 = FF FE … F8 value1Offset = 0 value1Length = 8 value2 = F7 F6 … F0 value2Offset = 0 value2Length = 8 **Verify Current TLV: Call getValueLength()** **Clear the handler**	Result is 03h	
17	**Successful call** appendTLV() tag = 04 value1 = FF FE … F8 value1Offset = 0 value1Length = 8 value2 = F7 F6 … F0 value2Offset = 0 value2Length = 8 **Call copy() method** **Compare handler** **CompareBuffer = 04 10 FF FE … F0**	Result is 00	
18	**Successful call** appendTLV() tag = 85h value1 = 00 01 … 07 value1Offset = 2 value1Length = 6 value2 = 08 09 … 0F value2Offset = 2 **value2Length = 6** **Call copy() method** **Compare handler** **compareBuffer = 04 10 FF FE … F0 85 0C 02 03 04 05 06 07 0A 0B 0C 0D 0E 0F**	Result is 00	

Id	Description	API Expectation	APDU Expectation
19	**Successful call** appendTLV() tag = 01 value1 = 11 22 … 88 value1Offset = 2 value1Length = 4 value2 = 99 AA … FF 00 value2Offset = 2 value2Length = 4 **Call copy() method** **Compare handler** **compareBuffer = 04 10 FF FE … F0 85 0C 02 03 04 05 06 07 0A 0B 0C 0D 0E 0F 01 08 33 44 55 66 BB CC DD EE** **Clear the handler**	Result is 00	
20	**Successful call** appendTLV() tag = 04 value1 = 00 01 … 7F value1Offset = 0 value1Length = 80h value2 = 80 81 … FB value2Offset = 0 value2Length = 7Ch **Call copy() method** **Compare handler** **compareBuffer = 04 81 FC 00 01…FB**	Result is 00	

5.2.4.30 Method initMoreTime

Test Area Reference: Api_2_Pah_Inmt.

5.2.4.30.1 Conformance requirement

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

```
public void initMoreTime()
```

5.2.4.30.1.1 Normal execution

- CRRN1: Builds a More Time Proactive command without sending the command. The Comprehension Required flags are all set to 1.
- CRRN2: After the method invocation no TLV is selected.

5.2.4.30.1.2 Parameter errors

No requirements.

5.2.4.30.1.3 Context errors

No requirements.

5.2.4.30.2 Test area files

Test Source: Test_Api_2_Pah_Inmt.java.

Test Applet: Api_2_Pah_Inmt_1.java.

Cap File: api_2_pah_inmt.cap.

5.2.4.30.3 Test coverage

CRR number	Test case number
N1	1
N2	2

5.2.4.30.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call initMoreTime() method** 1- Call ProactiveHandler.initMoreTime(). 2- Call ProactiveHandler.send() method.		1- MORE TIME proactive command is fetched. TERMINAL RESPONSE of MORE TIME is sent to the UICC.
2	**Select a TLV in the ProactiveHandler** **Call the initMoreTime() method** 1- Select 1st TLV of the Proactive Handler. 2- Call ProactiveHandler.initMoreTime(). 3- Call ViewHandler.getValueLength() method.	3- UNAVAILABLE_ELEMENT ToolkitException is thrown by getValueLength() method.	

5.2.5 Interface ProactiveResponseHandler

5.2.5.1 Method copyAdditionalInformation

Test Area Reference: Api_2_Prh_Cpai.

5.2.5.1.1 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
```

5.2.5.1.1.1 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.

5.2.5.1.1.2 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.

5.2.5.1.1.3 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.

5.2.5.1.2 Test area files

Test Source: Test_Api_2_Prh_Cpai.java.

Test Applet: Api_2_Prh_Cpai_1.java.

Cap File: api_2_prh_cpai.cap.

5.2.5.1.3 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

5.2.5.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation

1	**Build and send a DISPLAY TEXT command**
	`qualifier = 0`
	`dcs = 4`
	`buffer = "Text"`
			DISPLAY TEXT 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
	`copyAdditionalInformation()`
	`dstBuffer = NULL`
		NullPointerException is thrown	

2	**dstOffset > dstBuffer.length**
	`copyAdditionalInformation()`
	`dstBuffer.length = 10`
	`dstOffset = 11`
	`dstLength = 0`
		ArrayIndexOutOfBoundsException is thrown	

3	**dstOffset < 0**
	`copyAdditionalInformation()`
	`dstBuffer.length = 10`
	`dstOffset = -1`
	`dstLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

4	**dstLength > dstBuffer.length**
	`copyAdditionalInformation()`
	`dstBuffer.length = 10`
	`dstOffset = 0`
	`dstLength = 11`
		ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
5	**dstOffset + dstLength > dstBuffer.length** copyAdditionalInformation() dstBuffer.length = 10 dstOffset = 6 dstLength = 5	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength < 0** copyAdditionalInformation() 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 copyAdditionalInformation() dstBuffer.length = 5 dstOffset = 0 dstLength = 5	result of copyAdditionalInformation() is 05h.	
8	**Compare dstBuffer using arrayCompare() method** 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 Result TLV = 03 07 01 AB CD EF FE DC BA		
	Successful call, dstBuffer is part of a buffer copyAdditionalInformation() dstBuffer.length = 7 dstOffset = 2 dstLength = 5	result of copyAdditionalInformation() is 07h.	
11	**Compare dstBuffer using arrayCompare() method** 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 copyAdditionalInformation() dstBuffer.length = 7 dstOffset = 0 dstLength = 5	result of copyAdditionalInformation() is 05h.	
13	**Compare dstBuffer using arrayCompare() method** src = {FE, DC, BA, 98, 76} srcOffset = 00 dest = dstBuffer destOffset = 0 length = 5	result of arrayCompare() is 00h.	

Id	Description	API Expectation	APDU Expectation
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
	`copyAdditionalInformation()`
	`dstBuffer.length = 9`
	`dstOffset = 2`
	`dstLength = 5`
		result of copyAdditionalInformation() is 07h.	

15	**Compare dstBuffer using arrayCompare() method**
	`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
	`copyAdditionalInformation()`
	`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
	`copyAdditionalInformation()`
	`dstBuffer.length = 6`
	`dstOffset = 0`
	`dstLength = 6`
		OUT_OF_TLV_BOUNDARIES 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`		
	Initialize dstBuffer `dstBuffer = {00h, 01h, 02h, 03h…}`		

	Call the copyAdditionalInformation() method
	`dstBuffer.length = 20`
	`dstOffset = 5`
	`dstLength = 5`		

	Compare dstBuffer using arrayCompare() method
	`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	

Id	Description	API Expectation	APDU Expectation
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() method
	`dstBuffer.length = 5`
	`dstOffset = 0`
	`dstLength = 5`
		result of copyAdditionalInformation() is 05h.	

22	**Compare dstBuffer using arrayCompare() method**
	
	`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 Comprehension TLV	ToolkitException.UNAVAILABLE_ELEMENT is thrown by send()	
	ProactiveResponseHandler, getTheHandler call copyAdditionalInformation()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	

5.2.5.2 Method copyTextString

Test Area Reference: Api_2_Prh_Cpts.

5.2.5.2.1 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
```

5.2.5.2.1.1 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).

5.2.5.2.1.2 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.

5.2.5.2.1.3 Context errors

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

5.2.5.2.2 Test area files

Test Source: Test_Api_2_Prh_Cpts.java.

Test Applet: Api_2_Prh_Cpts_1.java.

Cap File: api_2_prh_cpts.cap.

5.2.5.2.3 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

5.2.5.2.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Build and send a GET INPUT command** qualifier = 00h dcs = 04h buffer = 'Text' minRespLength = 00h maxRespLength = FFh		GET INPUT 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	
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 copyTextString() dstBuffer.length = 04h dstOffset = 02h	ArrayIndexOutOfBoundsException is thrown	
3	**dstOffset < 0** copyTextString() 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	

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

Id	Description	API Expectation	APDU Expectation
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		
	Initialize 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		
	Initialize 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 2nd Text String TLV = 0D 02 04 44		
	Initialize 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	

5.2.5.3 Method getAdditionalInformationLength

Test Area Reference: Api_2_Prh_Gtil.

5.2.5.3.1 Conformance requirement

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

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

5.2.5.3.1.1 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.

5.2.5.3.1.2 Parameter errors

No requirements.

5.2.5.3.1.3 Context errors

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

5.2.5.3.2 Test area files

Test Source: Test_Api_2_Prh_Gtil.java.

Test Applet: Api_2_Prh_Gtil_1.java.

Cap File: api_2_prh_gtil.cap.

5.2.5.3.3 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

5.2.5.3.4 Test procedure

Id	Description	API Expectation	APDU Expectation

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

	Terminal Response without additional information		
	ProactiveResponseHandler.getTheHandler() **;** call the getAdditionalInformationLength() method	Result is 00h	

Id	Description	API Expectation	APDU Expectation
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	
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	

Id	Description	API Expectation	APDU Expectation
15	**Build and send a DISPLAY TEXT command**		DISPLAY TEXT Proactive command
	Terminal Response without Result Comprehension TLV	ToolkitException.UNAVAILABLE_ELEMENT is thrown by send()	
	Get ProactiveResponseHandler		
	Call the getAdditionalInformationLength() method	ToolkitException.UNAVAILABLE_ELEMENT is thrown by getAdditionalInformationLength ()	

5.2.5.4 Method getGeneralResult

Test Area Reference: Api_2_Prh_Gtgr.

5.2.5.4.1 Conformance requirement

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

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

5.2.5.4.1.1 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.

5.2.5.4.1.2 Parameter errors

No requirements.

5.2.5.4.1.3 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 Comprehension TLV.

5.2.5.4.2 Test area files

Test Source: Test_Api_2_Prh_Gtgr.java.

Test Applet: Api_2_Prh_Gtgr_1.java.

Cap File: api_2_prh_gtgr.cap.

5.2.5.4.3 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

5.2.5.4.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Build and send a DISPLAY TEXT command** qualifier = 00h dcs = 04h buffer = 'Text'		DISPLAY TEXT 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	
10	**Call the getValueLength() method**	Result is 80h	

Id	Description	API Expectation	APDU Expectation
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 Comprehension 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 Comprehension TLV		
	ProactiveResponseHandler.getTheHandler() ; **call the getGeneralResult() method** `Result TLV = 03 00`	OUT_OF_TLV_BOUNDARIES ToolkitException is thrown	

5.2.5.5 Method getItemIdentifier

Test Area Reference: Api_2_Prh_Gtii.

5.2.5.5.1 Conformance requirement

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

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

5.2.5.5.1.1 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.

5.2.5.5.1.2 Parameter errors

No requirements.

5.2.5.5.1.3 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 Comprehension TLV.

5.2.5.5.2 Test area files

Test Source: Test_Api_2_Prh_Gtii.java.

Test Applet: Api_2_Prh_Gtii_1.java.

Cap File: api_2_prh_gtii.cap.

5.2.5.5.3 Test coverage

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

5.2.5.5.4 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 Comprehension TLV `Item Identifier TLV = 10 00`		
	Call to getItemIdentifier()	OUT_OF_TLV_BOUNDARIES ToolkitException is thrown	

5.2.5.6 Method getTextStringCodingScheme

Test Area Reference: Api_2_Prh_Gtcs.

5.2.5.6.1 Conformance requirement

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

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

5.2.5.6.1.1 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.

5.2.5.6.1.2 Parameter errors

No requirements.

5.2.5.6.1.3 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.

5.2.5.6.2 Test area files

Test Source: Test_Api_2_Prh_Gtcs.java.

Test Applet: Api_2_Prh_Gtcs_1.java.

Cap File: api_2_prh_gtcs.cap.

5.2.5.6.3 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

5.2.5.6.4 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	
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	

Id	Description	API Expectation	APDU Expectation
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	

5.2.5.7 Method GetTextStringLength

Test Area Reference: Api_2_Prh_Gttl.

5.2.5.7.1 Conformance requirement

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

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

5.2.5.7.1.1 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.

5.2.5.7.1.2 Parameter errors

No requirements.

5.2.5.7.1.3 Context errors

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

5.2.5.7.2 Test area files

Test Source: Test_Api_2_Prh_Gttl.java.

Test Applet: Api_2_Prh_Gttl_1.java.

Cap File: api_2_prh_gttl.cap.

5.2.5.7.3 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

5.2.5.7.4 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	
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	

Id	Description	API Expectation	APDU Expectation
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	

5.2.5.8 Method getLength

Test Area Reference Api_2_Prh_Glen.

5.2.5.8.1 Conformance requirement

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

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

5.2.5.8.1.1 Normal execution

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

5.2.5.8.1.2 Parameter errors

No requirements.

5.2.5.8.1.3 Context errors

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

5.2.5.8.2 Test area files

Test Source: Test_Api_2_Prh_Glen.java.

Test Applet: Api_2_Prh_Glen_1.java.

Cap File: api_2_prh_glen.cap.

5.2.5.8.3 Test coverage

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

5.2.5.8.4 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() `Call getLength() method`	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() `Call getLength() method`	Result of getLength() is FFh	

5.2.5.9 Method copy

Test Area Reference Api_2_Prh_Copy.

5.2.5.9.1 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
```

5.2.5.9.1.1 Normal execution

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

5.2.5.9.1.2 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 Comprehension TLV List, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.OUT_OF_TLV_BOUNDARIES.

5.2.5.9.1.3 Context errors

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

5.2.5.9.2 Test area files

Test Source: Test_Api_2_Prh_Copy.java.

Test Applet: Api_2_Prh_Copy_1.java.

Cap File: api_2_prh_copy .cap.

5.2.5.9.3 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

5.2.5.9.4 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**
	`copy()`
	`dstBuffer.length = 5`
	`dstOffset = 6`
	`dstLength = 0`
		ArrayIndexOutOfBoundsException is thrown	

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

4	**dstLength > dstBuffer.length**
	`copy()`
	`dstBuffer.length = 5`
	`dstOffset = 0`
	`dstLength = 6`
		ArrayIndexOutOfBoundsException is thrown	

5	**dstOffset + dstLength > dstBuffer.length**
	`copy()`
	`dstBuffer.length = 5`
	`dstOffset = 3`
	`dstLength = 3`
		ArrayIndexOutOfBoundsException is thrown	

6	**dstLength < 0**
	`copy()`
	`dstBuffer.length = 5`
	`dstOffset = 0`
	`dstLength = -1`
		ArrayIndexOutOfBoundsException is thrown	

7	**dstLength > length of the Comprehension TLV list**
	`copy()`
	`dstBuffer.length = 13`
	`dstOffset = 0`
	`dstLength = 13`
		ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

8	**Successful call, dstBuffer is the whole buffer**
	`copy()`
	`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**
	`copy()`
	`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	**Initialize dstBuffer** dstBuffer = 00h 01h 02h … 13h		
	Successful call, dstBuffer is part of a buffer copy() 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	

5.2.5.10 Method findTLV

Test Area Reference Api_2_Prh_Find.

5.2.5.10.1 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
```

5.2.5.10.1.1 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.

5.2.5.10.1.2 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.

5.2.5.10.1.3 Context errors

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

5.2.5.10.2 Test area files

Test Source: Test_Api_2_Prh_Find.java.

Test Applet: Api_2_Prh_Find_1.java.

Cap File: api_2_prh_find.cap.

5.2.5.10.3 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

5.2.5.10.4 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** findTLV() tag = 01h occurrence = 1	Result is TLV_FOUND_CR_SET	
3	**Call the getValueLength() method**	Result is 03h	
4	**Search 2nd TLV** findTLV() 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 findTLV() 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** findTLV() tag = 01h occurrence = 2	Result is TLV_NOT_FOUND	
9	**Call the getValueLength() method**	ToolkitException.UNAVAILABLE_ELEMENT shall be thrown.	
10	**Search 3rd TLV** findTLV() tag = 03h occurrence = 1	Result is TLV_FOUND_CR_NOT_SET	
11	**Call the getValueLength() method**	Result is 01h	
12	**Search 3rd TLV** findTLV() tag = 03h occurrence = 2	Result is TLV_FOUND_CR_NOT_SET	
13	**Call the getValueLength() method**	Result is 02h	

Id	Description	API Expectation	APDU Expectation

14	**Search tag 83h**
	`findTLV()`
	`Tag = 83h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_NOT_SET	

15	**Search tag 82h**
	`findTLV()`
	`Tag = 82h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_SET	

5.2.5.11 Method getValueLength

Test Area Reference Api_2_Prh_Gvle.

5.2.5.11.1 Conformance requirement

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

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

5.2.5.11.1.1 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.

5.2.5.11.1.2 Parameter errors

No requirements.

5.2.5.11.1.3 Context errors

- CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.HANDLER_NOT_AVAILABLE.
- CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.UNAVAILABLE_ELEMENT.

5.2.5.11.2 Test area files

Test Source: Test_Api_2_Prh_Gvle.java.

Test Applet: Api_2_Prh_Gvle_1.java.

Cap File: api_2_prh_gvle.cap.

5.2.5.11.3 Test coverage

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

5.2.5.11.4 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() Call getValueLength() method	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	**Search TLV 0Dh**		
	Call getValueLength() method	Result is 00h	
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)		
	Call getValueLength() method	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)		
	Call getValueLength() method	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)		
	Call getValueLength() method	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)		
	Call getValueLength() method	Result is F0h	

5.2.5.12 Method getValueByte

Test Area Reference Api_2_Prh_Gvby.

5.2.5.12.1 Conformance requirement

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

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

5.2.5.12.1.1 Normal execution

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

5.2.5.12.1.2 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.

5.2.5.12.1.3 Context errors

- CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.HANDLER_NOT_AVAILABLE.
- CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.UNAVAILABLE_ELEMENT.

5.2.5.12.2 Test area files

Test Source: Test_Api_2_Prh_Gvby.java.

Test Applet: Api_2_Prh_Gvby_1.java.

Cap File: api_2_prh_gvby.cap.

5.2.5.12.3 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

5.2.5.12.4 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()		
	`Call getValueByte(0) method`	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	**Search TLV 01h (Command Details TLV)**		
	`Call getValueByte(3) method`	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
3	**Search TLV 01h (Command Details TLV)**		
	`Call getValueByte(2) method`	Result is 00h (qualifier)	
4	**Search TLV 02h (Device Identities TLV)**		
	`Call getValueByte(0) method`	Result is 82h (Source)	
5	**Search TLV 0Dh (Text String TLV)**		
	`Call getValueByte(7E) method`	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)		
	`Call getValueByte(7E) method`	Result is 7Eh	
7	`Call getValueByte(7F) method`	Result is 7Fh	
8	`Call getValueByte(EF) method`	Result is EFh	

5.2.5.13 Method copyValue

Test Area Reference Api_2_Prh_Cpyvs_Bss.

5.2.5.13.1 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
```

5.2.5.13.1.1 Normal execution

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

5.2.5.13.1.2 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.

5.2.5.13.1.3 Context errors

- CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.HANDLER_NOT_AVAILABLE.
- CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.UNAVAILABLE_ELEMENT.

5.2.5.13.2 Test area files

Test Source: Test_Api_2_Prh_Cpyv.java.

Test Applet: Api_2_Prh_Cpyv_1.java.

Cap File: api_2_prh_cpyv.cap.

5.2.5.13.3 Test coverage

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

5.2.5.13.4 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		
	call copyValue() method with a null dstBuffer	NullPointerException is thrown	
2	**dstOffset > dstBuffer.length** copyValue() dstBuffer.length = 5 dstOffset = 6 dstLength = 0	ArrayIndexOutOfBoundsException is thrown	
3	**dstOffset < 0** copyValue() dstBuffer.length = 5 dstOffset = -1 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**dstLength >dstBuffer.length** copyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**dstOffset + dstLength >dstBuffer.length** copyValue() dstBuffer.length = 5 dstOffset = 3 dstLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength < 0** copyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**valueOffset > Text String Length** copyValue() valueOffset = 7 dstBuffer.length = 15 dstOffset = 0 dstLength = 0	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**valueOffset < 0** copyValue() valueOffset = -1 dstBuffer.length = 15 dstOffset = 0 dstLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
9	**dstLength > Text String length** copyValue() valueOffset = 0 dstBuffer.length = 15 dstOffset = 0 dstLength = 7	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
10	**ValueOffset + dstLength > Text String length** copyValue() 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		
	call copyValue() method	ToolkitException.UNAVAILABLE_ELEMENT is thrown	

Id	Description	API Expectation	APDU Expectation
12	**Select Text String TLV**		
	Successful call copyValue() 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	**Initialize dstBuffer** dstBuffer = 55 55 … 55		
	Successful call copyValue() 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	

5.2.5.14 Method compareValue

Test Area Reference Api_2_Prh_Cprv.

5.2.5.14.1 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
```

5.2.5.14.1.1 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 Comprehension TLV List is less than that in compareBuffer.
- CRRN3: returns 1 if the first miscomparing byte in Comprehension TLV List is greater than that in compareBuffer.

5.2.5.14.1.2 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, 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.

5.2.5.14.1.3 Context errors

- CRRC1: if the handler is busy an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.HANDLER_NOT_AVAILABLE.
- CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.UNAVAILABLE_ELEMENT.

5.2.5.14.2 Test area files

Test Source: Test_Api_2_Prh_Cprv.java.

Test Applet: Api_2_Prh_Cprv_1.java.

Cap File: api_2_prh_cprv.cap.

5.2.5.14.3 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

5.2.5.14.4 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**
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 6`
	`compareLength = 0`
		ArrayIndexOutOfBoundsException is thrown	

3	**compareOffset < 0**
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = -1`
	`compareLength = 1`
		ArrayIndexOutOfBoundsException is thrown	

4	**compareLength >compareBuffer.length**
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 0`
	`compareLength = 6`
		ArrayIndexOutOfBoundsException is thrown	

5	**compareOffset + compareLength >compareBuffer.length**
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 3`
	`compareLength = 3`
		ArrayIndexOutOfBoundsException is thrown	

6	**compareLength < 0**
	`compareValue()`
	`compareBuffer.length = 5`
	`compareOffset = 0`
	`compareLength = -1`
		ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
7	**valueOffset > Text String Length** compareValue() valueOffset = 7 compareBuffer.length = 15 compareOffset = 0 compareLength = 0	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**valueOffset < 0** compareValue() valueOffset = -1 compareBuffer.length = 15 compareOffset = 0 compareLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
9	**compareLength > Text String length** compareValue() valueOffset = 0 compareBuffer.length = 15 compareOffset = 0 compareLength = 7	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
10	**valueOffset + compareLength > Text String length** compareValue() 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()		
	call compareValue()method	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
12	**Select Text String TLV**		
	Initialize compareBuffer CompareBuffer = 04 00 01 … 0F		
	Compare buffers compareValue() ValueOffset = 0 CompareOffset = 0 CompareLength = 17	Result is 00h	
13	**Initialize compareBuffer** CompareBuffer = 04 00 01 … 10		
	Compare buffers with same parameters	Result is -1	
14	**Initialize compareBuffer** CompareBuffer = 03 00 01 … 0F		
	Compare buffers with same parameters	Result is +1	
15	**Initialize compareBuffer** CompareBuffer = 55 55 55 01 02 03 04 05 06 07 08 09 0A 0B 0C 55 55 55 55 55		
	Compare buffers compareValue() ValueOffset = 2 CompareOffset = 3 CompareLength = 12	Result is 00h	
16	**Initialize 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	

Id	Description	API Expectation	APDU Expectation
17	**Initialize 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	

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

Test Area Reference Api_2_Prh_Facyb_Bs.

5.2.5.15.1 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
```

5.2.5.15.1.1 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.

5.2.5.15.1.2 Parameter errors

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

5.2.5.15.1.3 Context errors

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

5.2.5.15.2 Test area files

Test Source: Test_Api_2_Prh_Facyb_Bs.java.

Test Applet: Api_2_Prh_Facyb_Bs_1.java.

Cap File: api_2_prh_facyb_bs.cap.

5.2.5.15.3 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

5.2.5.15.4 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()		
	call findAndCopyValue() method with a null dstBuffer	NullPointerException is thrown	

2	**dstOffset > dstBuffer.length**
	`findAndCopyValue()`
	`tag = 0Dh`
	`dstBuffer.length = 20`
	`dstOffset = 21`
		ArrayIndexOutOfBoundsException is thrown	

3	**dstOffset < 0**
	`findAndCopyValue()`
	`dstBuffer.length = 20`
	`dstOffset = -1`
		ArrayIndexOutOfBoundsException is thrown	

4	**length > dstBuffer.length**
	`findAndCopyValue()`
	`dstBuffer.length = 15`
	`dstOffset = 0`
		ArrayIndexOutOfBoundsException is thrown	

5	**dstOffset + length >dstBuffer.length**
	`findAndCopyValue()`
	`dstBuffer.length = 20`
	`dstOffset = 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)		
	findAndCopyValue() `tag = 04h`	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the getValueLength() method	ToolkitException.UNAVAILABLE_ELEMENT is thrown.	

7	**Successful call**
	`findAndCopyValue()`
	`Tag = 0Dh`
	`DstBuffer.length = 17`
	`DstOffset = 0`
		Result of findAndcopyValue() is 17	

8	**Compare buffer** `Buffer = 04 00 01 … 0F`	Result is 00h	
9	**Initialize dstBuffer** `dstBuffer = 55 55 … 55`		

	Successful call
	`findAndCopyValue()`
	`DstBuffer.length = 20`
	`DstOffset = 2`
		Result of findAndcopyValue() is 19	

Id	Description	API Expectation	APDU Expectation
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 findAndCopyValue() 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) findAndCopyValue() Tag = 8Dh DstBuffer.length = 17 DstOffset = 0	Result of findAndcopyValue() is 17	
14	**Compare buffer** Buffer = 04 00 01 … 0F	Result is 00h	

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

Test Area Reference Api_2_Prh_Facybbs_Bss.

5.2.5.16.1 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
```

5.2.5.16.1.1 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.

5.2.5.16.1.2 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.
- 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.

5.2.5.16.1.3 Context errors

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

5.2.5.16.2 Test area files

Test Source: Test_Api_2_Prh_Facybbs_Bss.java.

Test Applet: Api_2_Prh_Facybbs_Bss_1.java.

Cap File: api_2_prh_facybbs_bss.cap.

5.2.5.16.3 Test coverage

CRR number	Test case number
N1	13, 15, 17, 19
N2	11
N3	12, 14, 16, 18
N4	20, 21
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
P4	22
C1	Does not apply for Proactive Response Handler

5.2.5.16.4 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** findAndCopyValue() tag = 0Dh, occurrence = 1 valueOffset = 0 dstBuffer.length = 5 dstOffset = 6 dstLength = 0	ArrayIndexOutOfBoundsException is thrown	
3	**dstOffset < 0** findAndCopyValue() dstBuffer.length = 5 dstOffset = -1 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
4	**dstLength > dstBuffer.length** findAndCopyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**dstOffset + dstLength > dstBuffer.length** findAndCopyValue() dstBuffer.length = 5 dstOffset = 3 dstLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength < 0** findAndCopyValue() 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 findAndCopyValue() tag = 0Dh, occurrence = 1 valueOffset = 7 dstBuffer.length = 15 dstOffset = 0 dstLength = 0	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**valueOffset < 0** findAndCopyValue() valueOffset = -1 dstBuffer.length = 15 dstOffset = 0 dstLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
9	**dstLength > Text String length** findAndCopyValue() valueOffset = 0 dstBuffer.length = 15 dstOffset = 0 dstLength = 7	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
10	**valueOffset + dstLength > Text String length** findAndCopyValue() 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.	
12	**Successful call** findAndCopyValue() 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	**Initialize dstBuffer** dstBuffer = 55 55 … 55		
	Successful call findAndCopyValue() 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 findAndCopyValue() 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** findAndCopyValue() 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) findAndCopyValue() 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	**Invalid parameter** findAndCopyValue() occurrence = 0	ToolkitException.BAD_INPUT_PARAMETER is thrown	

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

Test Area Reference Api_2_Prh_Facrb_Bs.

5.2.5.17.1 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
```

5.2.5.17.1.1 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 Comprehension TLV is less than that in compareBuffer returns -1.
- CRRN5: if the first miscomparing byte in Comprehension TLV is greater than that in compareBuffer returns 1.
- CRRN6: The search method is comprehension required flag independent.

5.2.5.17.1.2 Parameter errors

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

5.2.5.17.1.3 Context errors

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

5.2.5.17.2 Test area files

Test Source: Test_Api_2_Prh_Facrb_Bs.java.

Test Applet: Api_2_Prh_Facrb_Bs_1.java.

Cap File: api_2_prh_facrb_bs.cap.

5.2.5.17.3 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

5.2.5.17.4 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**
	`findAndCompareValue()`
	`tag = 0Dh`
	`compareBuffer.length = 20`
	`compareOffset = 21`
		ArrayIndexOutOfBoundsException is thrown	

3	**compareOffset < 0**
	`findAndCompareValue()`
	`compareBuffer.length = 20`
	`compareOffset = -1`
		ArrayIndexOutOfBoundsException is thrown	

4	**length > compareBuffer.length**
	`findAndCompareValue()`
	`compareBuffer.length = 15`
	`compareOffset = 0`
		ArrayIndexOutOfBoundsException is thrown	

5	**CompareOffset + length > compareBuffer.length**
	`findAndCompareValue()`
	`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	**Initialize compareBuffer**
	`CompareBuffer =`
	`04 00 01 … 0F`		

	Compare buffers
	`findAndCompareValue()`
	`Tag = 0Dh`
	`CompareOffset = 0`
		Result is 00h	

8	**Verify current TLV** `Call getValueLength() method`	Result is 17	

9	**Initialize compareBuffer**
	`CompareBuffer =`
	`04 00 01 … 10`		

	Compare buffers with same parameters	Result is -1	

Id	Description	API Expectation	APDU Expectation
10	**Initialize compareBuffer** CompareBuffer = 03 00 01 … 0F		
	Compare buffers with same parameters	Result is +1	
11	**Initialize compareBuffer** CompareBuffer = 55 55 04 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 55		
	Compare buffers findAndCompareValue() CompareOffset = 2	Result is 00h	
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()		
	Initialize compareBuffer CompareBuffer = 55 55 04 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 55		
	Compare buffers findAndCompareValue() CompareOffset = 2	Result is 00h	
13	**Initialize compareBuffer** CompareBuffer = 55 55 04 01 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 55		
	Compare buffers findAndCompareValue() CompareOffset = 2	Result is -1	
14	**Initialize compareBuffer** CompareBuffer = 55 55 04 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0D 10 55		
	Compare buffers findAndCompareValue() 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()		
	Initialize compareBuffer CompareBuffer = 04 00 01 … 0F		
	Compare buffers (with tag 8Dh) findAndCompareValue() Tag = 8Dh CompareOffset = 0	Result is 00h	

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

Test Area Reference Api_2_Prh_Facrbbs_Bss.

5.2.5.18.1 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
```

5.2.5.18.1.1 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 Comprehension TLV is less than that in compareBuffer -1 is returned.
- CRRN5: if the first miscomparing byte in Comprehension TLV is greater than that in compareBuffer 1 is returned.
- CRRN6: The search method is comprehension required flag independent.

5.2.5.18.1.2 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.

5.2.5.18.1.3 Context errors

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

5.2.5.18.2 Test area files

Test Source: Test_Api_2_Prh_Facrbbs_Bss.java.

Test Applet: Api_2_Prh_Facrbbs_Bss_1.java.

Cap File: api_2_prh_facrbbs_bss.cap.

5.2.5.18.3 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

5.2.5.18.4 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** findAndCompareValue() tag = 0Dh, occurrence = 1 valueOffset = 0 compareBuffer.length = 5 compareOffset = 6 compareLength = 0	ArrayIndexOutOfBoundsException is thrown	
3	**compareOffset < 0** findAndCompareValue() compareBuffer.length = 5 compareOffset = -1 compareLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**compareLength >compareBuffer.length** findAndCompareValue() compareBuffer.length = 5 compareOffset = 0 compareLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**CompareOffset + compareLength >compareBuffer.length** findAndCompareValue() CompareBuffer.length = 5 CompareOffset = 3 CompareLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**compareLength < 0** findAndCompareValue() 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 findAndCompareValue() tag = 0Dh, occurrence = 1 valueOffset = 7 compareBuffer.length = 15 compareOffset = 0 compareLength = 0	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

Id	Description	API Expectation	APDU Expectation
8	**valueOffset < 0** findAndCompareValue() valueOffset = -1 compareBuffer.length = 15 compareOffset = 0 compareLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
9	**compareLength > Text String length** findAndCompareValue() valueOffset = 0 compareBuffer.length = 15 compareOffset = 0 compareLength = 7	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
10	**valueOffset + compareLength > Text String length** findAndCompareValue() valueOffset = 2 compareBuffer.length = 15 compareOffset = 0 compareLength = 5	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
11	**Invalid parameter** findAndCompareValue() Occurrence = 0	ToolkitException.BAD_INPUT_PARAMETER is thrown	
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	**Initialize compareBuffer** CompareBuffer = 04 00 01 … 0F		
	findAndCompareValue() tag = 0Dh, occurrence = 1 valueOffset = 0 compareOffset = 0 compareLength = 17	Result is 00h	
14	**Verify current TLV** Call getValueLength() method	Result is 17	
15	**Initialize compareBuffer** compareBuffer = 04 00 01 … 10		
	Compare buffers with same parameters	Result is -1	
16	**Initialize compareBuffer** compareBuffer = 03 00 01 … 0F		
	Compare buffers with same parameters	Result is +1	
17	**Initialize compareBuffer** compareBuffer = 55 55 55 01 02 03 04 05 06 07 08 09 0A 0B 0C 55 55 55 55 55		
	Compare buffers findAndCompareValue() valueOffset = 2 compareOffset = 3 compareLength = 12	Result is 00h	
18	**Initialize 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	

Id	Description	API Expectation	APDU Expectation
19	**Initialize compareBuffer** compareBuffer = 55 55 55 01 02 03 04 05 06 07 08 09 0A 0A 0D 55 55 55 55 55		
	Compare buffers with same parameters	Result is +1	
20	**Send a GET INPUT command**		GET INPUT 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()		
	Initialize compareBuffer compareBuffer = 04 00 01 … 0F		
	findAndCompareValue() tag = 0Dh, occurrence = 1 valueOffset = 0 compareOffset = 0 compareLength = 17	Result is 00h	
21	**Initialize compareBuffer** compareBuffer = 00 11 22 33 44 55		
	findAndCompareValue() tag = 0Dh, occurrence = 2 valueOffset = 0 compareOffset = 0 compareLength = 6	Result is 00h	
22	**Initialize 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()		
	Initialize compareBuffer CompareBuffer = 04 00 01 … 0F		
	Compare buffers (with tag 8Dh) findAndCompareValue() tag = 8Dh, occurrence = 1 valueOffset = 0 compareOffset = 0 compareLength = 17	Result is 00h	

5.2.5.19 Method getCapacity

Test Area Reference: Api_2_Prh_Gcap.

5.2.5.19.1 Conformance requirement

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

```
public byte getCapacity()
```

5.2.5.19.1.1 Normal execution

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

5.2.5.20.1.2 Parameter errors

No requirements.

5.2.5.20.1.3 Context errors

No requirements.

5.2.5.19.2 Test area files

Test Source: Test_Api_2_Prh_Gcap.java.

Test Applet: Api_2_Prh_Gcap_1.java.

Cap File: api_2_prh_gcap.cap.

5.2.5.19.3 Test coverage

CRR number	Test case number
N1	1

5.2.5.19.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**ProactiveResponseHandler available** 1- Send envelope Menu Selection 2- The applet sends a proactive command 3- Fetch the proactive command and send Terminal Response 4- The applet calls getCapacity() method 5- The applet calls 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

5.2.5.20 Method getChannelIdentifier

Test Area Reference: Api_2_Prh_Gcid.

5.2.5.20.1 Conformance requirement

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

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

5.2.5.201.1.1 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.

5.2.5.20.1.2 Parameter errors

No requirements.

5.2.5.20.1.3 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 Comprehension TLV Channel Status length is equal to 0.

5.2.5.20.2 Test area files

Test Source: Test_Api_2_Prh_Gcid.java.

Test Applet: Api_2_Prh_Gcid_1.java.

Cap File: api_2_prh_gcid.cap.

5.2.5.21.3 Test coverage

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

5.2.5.20.4 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() method to open a channel and ProactiveHandler.send() method.. 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() method to open a channel and ProactiveHandler.send() method 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.

Id	Description	API Expectation	APDU Expectation
5	**Channel status TLV is currently selected TLV** 1- Call ProactiveHandler.init() method to open a channel and ProactiveHandler.send() method. Call ViewHandler.FindTLV() method with Device Identity Tag. 2- Call ProactiveResponseHandler.getChannelIdentifier() method. 3- Compare ProactiveResponseHandler.getChannelIdentifier() and ViewHandler.getValueByte(0) method results.	2- Returns 0x03 3- Check getChannelIdentifier() =getValueByte(0)	1- OPEN CHANNEL Proactive Command is fetched. TERMINAL RESPONSE is issued with channel status value = 0x0305.

5.2.5.21 Method copyChannelData

Test Area Reference: Api_2_Prh_Cchd.

5.2.5.21.1 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
```

5.2.5.21.1.1 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.

5.2.5.21.1.2 Parameter errors

- 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.

5.2.5.21.1.3 Context errors

- CRRC1: The method shall throw a UNAVAILABLE_ELEMENT ToolkitException if the Result TLV is not present.

5.2.5.21.2 Test area files

Test Source: Test_Api_2_Prh_Cchd.java.

Test Applet: Api_2_Prh_Cchd_1.java.

Cap File: api_2_prh_cchd.cap.

5.2.5.21.3 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

5.2.5.21.4 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 = 6 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 = 6 DstOffset = 0 DstLength = -1 2- check dstBuffer is empty.	1- an ArrayIndexOutOfBoundsException exception is thrown. 2- no copy is performed.	

Id	Description	API Expectation	APDU Expectation
4	**CopyChannelData() with dstOffset+dstLength greater than dstBuffer.length** 1- call ProactiveResponseHandler.copyChannelData() with dstOffset+dstLength greater than dstBuffer.length. DstBuffer.length = 6 DstOffset = 5 DstLength = 2 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 = 10 DstOffset = 0 DstLength = 10	an 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 = 10 DstOffset = 0 DstLength = 10	2- an UNAVAILABLE_ELEMENT ToolkitException is thrown.	1- RECEIVE DATA proactive command is fetched TERMINAL RESPONSE without ChannelData TLV element.
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 = 6 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 = 6 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.	

Id	Description	API Expectation	APDU Expectation
12	**Successful copyChannelData()** 1- Initialize dstBuffer to [00, 01…] 2- Call ProactiveResponseHandler.copyChannelData() DstBuffer.length = 6 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 = 6 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 dstBuffer.	Check that dstBuffer contains the first Channel Data TLV from the TERMINAL RESPONSE.	

5.2.5.22 Method getValueShort

Test Area Reference: Api_2_Prh_Gvsh.

5.2.5.22.1 Conformance requirement

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

```
public short getValueShort(short valueOffset)
                    throws ToolkitException
```

5.2.5.22.1.1 Normal execution

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

5.2.5.22.1.2 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.

5.2.5.22.1.3 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.

5.2.5.22.2 Test area files

Specific triggering: None.

Test Source: Test_Api_2_Prh_Gvsh.java.

Test Applet: Api_2_Prh_Gvsh_1.java.

Cap File: api_2_prh_gvsh.cap.

5.2.5.22.3 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

5.2.5.22.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	Send a GET INPUT command **Terminal Response, Text String length = 7Eh** Text String TLV = 0D 7F 04 01 02 … 7E **ProactiveResponseHandler.getTheHandler()** Call getValueShort(0) method	ToolkitException.UNAVAILABLE_ELEMENT is thrown	GET INPUT Proactive command
2	Search TLV 01h (Command Details TLV) Call getValueShort(3) method	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
3	Search TLV 01h (Command Details TLV) Call getValueShort(1) method	Result is 23h 00h (Type, qualifier)	
4	Search TLV 02h (Device Identities TLV) Call getValueShort(0) method	Result is 82h 81h (Source, destination)	
5	Search TLV 0Dh (Text String TLV) Call getValueShort(7D) method	Result is 7Dh 7Eh	
6	Send a GET INPUT 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)** Call getValueShort(7D) method	Result is 7Dh 7Eh	GET INPUT Proactive command
7	Call getValueShort(7F) method	Result is 7Fh 80h	
8	Call getValueShort(EE) method	Result is EEh EFh	

5.2.5.23 Method getChannelStatus

Test Area Reference: Api_2_Prh_Gcst.

5.2.5.23.1 Conformance requirement

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

```
public short getChannelStatus(byte channelIdentifier)
                      throws ToolkitException
```

5.2.5.23.1.1 Normal execution

- CRRN1: The method shall return the value of the first Channel Status TLV element.
- CRRN2: The Channel Status value returned shall be from the element whose channel identifier is equal to the ChannelIdentifier parameter.
- CRRN3: If the element is available it becomes the currently selected TLV.

5.2.5.23.1.2 Parameter errors

No requirements.

5.2.5.23.1.3 Context errors

- CRRC1: The method shall throw ToolkitException.UNAVAILABLE_ELEMENT if no Channel Status TLV element with the right identifier could be found.
- CRRC2: The method shall throw ToolkitException.OUT_OF_TLV_BOUNDARIES if a Channel Status TLV element with the right identifier could be found but its value is less than 2 bytes long.

5.2.5.23.2 Test area files

Test Source: Test_ Api_2_Prh_Gcst.java.

Test Applet: Api_2_Prh_Gcst_1.java.

Cap File: api_2_prh_gcst.cap.

5.2.5.23.3 Test coverage

CRR number	Test case number
N1	6
N2	5,7
N3	8
C1	1,2,3
C2	4

5.2.5.23.4 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. getChannelStatus(0x01) 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 the identifier is not present** 1- Build and send a OPEN CHANNEL proactive command 2- Call ProactiveResponseHandler. getChannelStatus(0x02) method. 3- Call ProactiveHandler.initCloseChannel(0x01) and ProactiveHandler.send() methods.	2- UNAVAILABLE_ELEMENT ToolkitException is thrown	1- OPEN CHANNEL Proactive command is fetched. TERMINAL RESPONSE is issued with channel status value = 0x8100. 3- Succesfull terminal response to initCloseChannel proactive command.

Id	Description	API Expectation	APDU Expectation
3	**Channel status TLV with a length equal to 0** 1- Build and send a OPEN CHANNEL proactive command 2- Call ProactiveResponseHandler. getChannelStatus(0x01) method.	2- UNAVAILABLE_ELEMENT ToolkitException is thrown	1- OPEN CHANNEL Proactive command is fetched. TERMINAL RESPONSE with Channel status TLV length equal to 0.
4	**Channel status TLV with a length equal to 1** 1- Build and send a OPEN CHANNEL proactive command 2- Call ProactiveResponseHandler. getChannelStatus(0x01) 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 1.
5	**Get channel status value** 1- Call ProactiveHandler.init() method to open a channel and ProactiveHandler.send() method. 2- Call ProactiveResponseHandler. getChannelStatus(0x01) method. 3- Build and send a get channel status proactive command.	2- Returns 0x8100	1- OPEN CHANNEL Proactive Command is fetched. TERMINAL RESPONSE is issued with channel status value = 0x8100.
6	**Get channel status value with 2 TLV** 1- Build and send a get channel status proactive command 2- Call ProactiveResponseHandler. getChannelStatus(0x01) method.	2- Returns 0x8100	1- Get channel status proactive command is fetched. TERMINAL RESPONSE is issued with 2 channel status value = 0x8100 and 0x8101.
7	**Get channel status value with 2 TLV** 1- Build and send a Get Channel status proactive command. 2- Call ProactiveResponseHandler. getChannelStatus(0x01) method. 4- Call ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods.	2- Returns 0x8100	1- Get channel status proactive command is fetched. TERMINAL RESPONSE is issued with 2 channel status value = 0x8200 and 0x8100. 3- Succesfull terminal response to initCloseChannel proactive command.
8	**Channel status TLV is currently selected TLV** 1- Call ProactiveHandler.init() method to open a channel and ProactiveHandler.send() method. Call ViewHandler.FindTLV() method with Device Identity Tag. 2- Call ProactiveResponseHandler. getChannelStatus(0x03) method. 3- Compare ProactiveResponseHandler. getChannelStatus() and ViewHandler.getValueShort(0) method results. 4- Call ProactiveHandler.initCloseChannel(0x03) and ProactiveHandler.send() methods.	2- Returns 0x0304 3- Check getChannelStatus() =getValueShort(0)	1- OPEN CHANNEL Proactive Command is fetched. TERMINAL RESPONSE is issued with channel status value = 0x0304. 3- Succesfull terminal response to initCloseChannel proactive command.

5.2.6 Interface ToolkitConstants

5.2.6.1 Constants

Test Area Reference: Api_2_Tkc_Cons.

5.2.6.1.1 Conformance requirement

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

5.2.6.1.1.1 Normal execution

- CRRN1: The Toolkit Constants shall all have the same name and value as defined in ETSI TS 102 241 [9].

5.2.6.1.1.2 Parameter errors

No requirements.

5.2.6.1.1.3 Context errors

No requirements.

5.2.6.1.2 Test area files

None.

5.2.6.1.3 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.

5.2.7 Interface ToolkitInterface

5.2.7.1 Method processToolkit

Test Area Reference: Api_2_Tki_Prtk.

5.2.7.1.1 Conformance requirement:

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

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

5.2.7.1.1.1 Normal execution

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

5.2.7.1.1.2 Parameter errors

No requirements.

5.2.7.1.1.3 Context errors

No requirements.

5.2.7.1.2 Test area files

The method is tested in the CAT Runtime Environment.

5.2.7.1.3 Test coverage

CRR number	Test case number
N1	Tested in the whole test suite
N2	Tested in the whole test suite

5.2.8 Interface ToolkitRegistry

5.2.8.1 Method allocateTimer

Test Area Reference: Api_2_Tkr_Atim.

5.2.8.1.1 Conformance requirement:

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

```
public byte allocateTimer()
                  throws ToolkitException
```

5.2.8.1.1.1 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: By calling this method the applet is registered to the EVENT_TIMER_EXPIRATION of the allocated timer.
- CRRN4: The timer is allocated by the applet until it explicitly releases it.
- CRRN5: When a timer is allocated, the applet can issue the Timer Management proactive command to start, stop or get the value of its allocated timer.

5.2.8.1.1.2 Parameter errors

No requirements.

5.2.8.1.1.3 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.

5.2.8.1.2 Test area files

Test Source: Test_Api_2_Tkr_Atim.java.

Test Applet: Api_2_Tkr_Atim_1.java.

Api_2_Tkr_Atim_2.java.

Api_2_Tkr_Atim_3.java.

Cap File: api_2_tkr_atim.cap.

Installation parameters:

- The maximum number of timers 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.

5.2.8.1.3 Test coverage

CRR number	Test case number
N1	1, 4
N2	1, 4
N3	3
N4	3, 4
N5	Cat Runtime Environment, Cre_Pcs_Pcco
C1	2
C2	5

5.2.8.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Allocates up to 8 timers (Applet1)** `Call 8 times allocateTimer()and isEventSet(TIMER EXPIRATION).`	No exception shall be thrown. Timer ID returned shall be between 01 and 08 inclusive. It shall be different after each call. Shall return true.	
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).`
	`Call releaseTimer(id) each time a timer expires.`
	`Call isEventSet(EVENT_TIMER_EXPIRATION) method`
		Shall trigger each time an ENVELOPE(TIMER EXPIRATION) is sent to the UICC, for Timer ID = '01' to '08'.
		Returns false.	

4	**Allocate up to 4 timers (Applet2)** `Call 4 times 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.	

5.2.8.2 Method changeMenuEntry

Test Area Reference: Api_2_Tkr_Cmet.

5.2.8.2.1 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
```

5.2.8.2.1.1 Normal execution

- CRRN1: After the invocation of this method, during the current card session, the CAT Runtime Environment shall dynamically update the menu stored in the Terminal. The CAT Runtime Environment shall use the data of the EF_{SUME} file under the DF_Telecom when issuing the SET UP MENU proactive 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 changeMenuEntry() method is called with helpSupported set to true then a call to isEventSet() method on EVENT_MENU_SELECTION_HELP_REQUEST event shall return true.
- CRRN5: if changeMenuEntry() method is called with helpSupported set to true andif an ENVELOPE(MENU_SELECTION_HELP_REQUEST) command is received by the UICC for this entry, then after the completion of the SET UP MENU command, the CAT Runtime Environment shall trigger the applet.
- CRRN6: if changeMenuEntry() method is called with helpsupported set to true, the CAT Runtime Environment shall issue a SET UP MENU command with command qualifier = '80'.
- CRRN7: if changeMenuEntry() method is called with helpSupported set to false and if no entry is supporting help then a call to isEventSet() method on EVENT_MENU_SELECTION_HELP_REQUEST event shall return false.
- CRRN8: if changeMenuEntry() method is called with helpSupported set to false, if no entry is supporting help and if an ENVELOPE(MENU_SELECTION_HELP_REQUEST) command is received by the UICC, then after the completion of the SET UP MENU command, the CAT Runtime Environment shall not trigger the applet.
- CRRN9: The CAT Runtime Environment shall supply in the SET UP MENU command, the icon identifier provided in the icon identifier list within the item icon identifier list Comprehension TLV if all the applets registered to the EVENT_MENU_SELECTION provide it.
- CRRN10: The CAT Runtime Environment shall set in the SET UP MENU command, 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 is different from '00', the CAT Runtime Environment shall issue a SET UP MENU proactive command containing an Item Next Action Indicator Comprehension TLV with the comprehension flag set to 0 as defined in ETSI TS 102 223 [6].

5.2.8.2.1.2 Parameter errors

- CRRP1: A java.lang.NullPointerException is thrown if menuEntry is null.
- CRRP2: A java.lang.ArrayIndexOutOfBoundsException is thrown if offset would cause access outside array bounds.
- CRRP3: A java.lang.ArrayIndexOutOfBoundsException is thrown if length would cause access outside array bounds.

- CRRP4: A java.lang.ArrayIndexOutOfBoundsException is thrown if both offset and length would cause access outside array bounds.

5.2.8.2.1.3 Context errors

- CRRC1: A ToolkitException with MENU_ENTRY_NOT_FOUND reason is thrown if the Menu Identifier isn't associated to the calling applet instance.
- CRRC2: A ToolkitException with ALLOWED_LENGTH_EXCEEDED reason is thrown if the menu entry string is bigger than the allocated space.

5.2.8.2.2 Test area files

Additional requirements for the UICC personalization:

- content of EF sume shall be:

Title Alpha Identifier: "TOOLKIT TEST".

Test Source: Test_Api_2_Tkr_Cmet.java.

Test Applet: Api_2_Tkr_Cmet_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'.

Cap File: api_2_tkr_cmet.cap.

5.2.8.2.3 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	Tested in CAT Runtime Environment: Cre_Apt_Emsh (Test case 1)
N9	8, 9
N10	8
N11	4
P1	10
P2	11, 12, 13
P3	14, 15
P4	16
C1	17, 18
C2	19

5.2.8.2.4 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- Call changeMenuEntry() with parameters: Id = '02' MenuEntry = "UseAllBuffer" Offset = 0 Length = menuEntry.length NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0. 2- Call isEventSet(EVENT_MENU_SELECTION). 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST).	1- No exception shall be thrown. 2- shall return true. 3- shall return false.	The UICC shall issue a SETUP MENU proactive command which contains the new text for entry ID '02'.
2	**Changing the title with part of menuEntry buffer** 1- Call changeMenuEntry() with parameters: Id = '01' MenuEntry = "UsePartOfBuffer" Offset = 3 Length = 12 NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0. 2- Call isEventSet(EVENT_MENU_SELECTION). 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	1- No exception shall be thrown. 2- Shall return true. 3- Shall return false.	The UICC shall issue a SETUP MENU proactive command which contains the new text for entry ID '01'.
3	**Length = 0** 1- Call 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- Call isEventSet(EVENT_MENU_SELECTION). 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST).	1- No exception shall be thrown. 2- Shall return true. 3- shall return false.	The UICC shall issue a SETUP MENU proactive command which contains for entry '01'and entry '02', no text part.

Id	Description	API Expectation	APDU Expectation
4	**Setting a next action indicator != 0** 1- Call changeMenuEntry() with parameters: Id = '02' MenuEntry = "NextActionIndic" Offset = 0 Length = menuEntry.length NextAction = '10' (SETUP CALL) HelpSupported = false IconQualifier = 0 IconIdentifier = 0 2- Call isEventSet(EVENT_MENU_SELECTION). 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST). 4- Call 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 UICC shall issue a SETUP MENU proactive command which contains an Items Next Action Indicator list and which contains a command qualifier '80'.
5	**Checking applet is triggered by a MENU_SELECTION_HELP_REQUEST** Send ENVELOPE(MENU_SELECTION_HELP_REQUEST) with Item Identifier = '02'	Applet is triggered by a MENU_SELECTION_HELP_REQUEST and the Item Identifier is 02	
6	**help supported=true** 1- Call changeMenuEntry() with parameters: Id = '01' MenuEntry = "HelpSupported" Offset = 0 Length = menuEntry.length NextAction = 0 HelpSupported = true IconQualifier = 0 IconIdentifier = 0 2- Call isEventSet(EVENT_MENU_SELECTION). 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST).	1- No exception shall be thrown. 2- Shall return true. 3- Shall return true.	The UICC 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 triggered by a MENU_SELECTION_HELP_REQUEST and the Item Identifier is 01	

Id	Description	API Expectation	APDU Expectation
8	**Setting icons, help supported = false** 1- call 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- Call isEventSet(EVENT_MENU_SELECTION). 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST).	1- No exception shall be thrown. 2- Shall return true. 3- Shall return false.	The UICC shall issue a SETUP MENU proactive command which contains an Icon Identifier List.
9	**MenuEntry is disabled** 1- Call disableMenuEntry('01'). 2- Call changeMenuEntry() with parameters: Id = '01' MenuEntry = "EnableEntry" Offset = 0 Length = menuEntry.length NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0 3- Call isEventSet(EVENT_MENU_SELECTION). 4- Call 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 UICC shall issue a SETUP MENU proactive command which contains the entry. Without Icon identifier List Comprehension TLV
10	**MenuEntry is null** Call ChangeMenuEntry() method with parameters: MenuEntry = NULL	Shall throw java.lang.NullPointerException.	
11	**Offset causes access outside array bounds** changeMenuEntry() 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** changeMenuEntry() Id = '01' MenuEntry = "Violation" Offset = 255 Length = 1 NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0	Shall throw java.lang.ArrayIndexOutOfBoundsException.	

Id	Description	API Expectation	APDU Expectation
13	**Offset < 0 causes access outside array bounds** changeMenuEntry() 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** changeMenuEntry() 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** changeMenuEntry() 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** changeMenuEntry() Id = '01' MenuEntry = "Violation" Offset ∈ [1, MenuEntry.length] Length = MenuEntry.length NextAction = 1 HelpSupported = false IconQualifier = 0 IconIdentifier = 0	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
17	**Invalid ID used** changeMenuEntry() 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** changeMenuEntry() 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.	

Id	Description	API Expectation	APDU Expectation
19	**The text is bigger than the allocated space** changeMenuEntry() 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. Call isEventSet(EVENT_MENU_SELECTION) 3. Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	1. No exception shall be thrown. 2. Shall return true. 3. Shall return false.	The UICC shall issue a SETUP MENU proactive command which contains the new text for entry ID '02'.

5.2.8.3 Method clearEvent

Test Area Reference: Api_2_Tkr_Cevt.

5.2.8.3.1 Conformance requirement:

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

```
public void clearEvent(short event)
                throws ToolkitException,
                       javacard.framework.TransactionException
```

5.2.8.3.1.1 Normal execution

- CRRN1: A call to isEventSet() method for a cleared event shall return false after a call to clearEvent.
- CRRN2: The CAT Runtime Environment shall not trigger the applet on the occurrence of the cleared event anymore.
- CRRN3: After the call to clearEvent() method with EVENT_CALL_CONTROL_BY_NAA event, no applet is registered to this event, and the CAT Runtime Environment shall allow an applet to register to this event.
- CRRN4: If an applet is still registered to EVENT_CALL_CONTROL_BY_NAA event, the CAT Runtime Environment shall not allow an applet to register to it.

5.2.8.3.1.2 Parameter errors

- CRRP1: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event is EVENT_MENU_SELECTION.
- CRRP2: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event is EVENT_MENU_SELECTION_HELP_REQUEST.
- CRRP3: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event is EVENT_TIMER_EXPIRATION.

- CRRP4: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event is EVENT_STATUS_COMMAND.
- CRRP5: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event is EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION.

5.2.8.3.1.3 Context errors

- CRRC1: shall throw javacard.framework.TransactionException - if the operation would cause the commit capacity to be exceeded.

5.2.8.3.2 Test area files

Test Source: Test_Api_2_Tkr_Cevt.java.

Test Applet: Api_2_Tkr_Cevt_1.java.

- As default but applet registers to an event list which contains all defined events in ETSI TS 102 241 [9] excepted those that are not allowed or supported by setEvent().

Cap File: api_2_tkr_cevt.cap.

5.2.8.3.3 Test coverage

CRR number	Test case number
N1	1, 2
N2	4
N3	Tested in CAT Runtime Environment, Cre_Apt_Eccn
N4	Tested in CAT Runtime Environment, Cre_Apt_Eccn
P1	3
P2	3
P3	3
P4	3
P5	3
C1	not testable

5.2.8.3.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Clear ALLOWED unregistered events** For events ranging from -1, 1, 7 to 9, 11 to 23, 25 to 29, 123, 124, 126 and 127* excepted those that aren't allowed (7, 8, 11, 19, 27), the applet calls: 1- Call clearEvent() method 2- Call 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, 7 to 9, 11 to 23, 25 to 29, 123, 124, 126 and 127)* excepted (7, 8, 11, 19, 27, 124), the applet calls setEvent() method. Call registerFileEvent() method. 2- For each ALLOWED and SUPPORTED event (-1, 1, 7 to 9, 11 to 23, 25 to 29, 123, 124, 126 and 127)* excepted (7, 8, 11, 19, 27), the applet calls: 2.1- Call clearEvent() method 2.2- Call isEventSet() method	1- No exception shall be thrown. 2.1- No exception shall be thrown. 2.2- Shall return false.	

Id	Description	API Expectation	APDU Expectation
3	**Clear NOT ALLOWED events** For each event among: EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION, EVENT_STATUS_COMMAND, EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION, 1- The applet calls clearEvent(event) method.	1- Each time, clearEvent shall throw a ToolkitException with reason EVENT_NOT_ALLOWED.	
4	**Check applet is not triggered by an ENVELOPE(EVENT_EVENT_DOWNLOAD_USER_ACTIVITY) command** 1 - reset and initialize the card 2 - An ENVELOPE(EVENT_EVENT_DOWNLOAD_USER_ACTIVITY) is sent.	Applet is not triggered by an ENVELOPE(EVENT_EVENT_DOWNLOAD_USER_ACTIVITY) command	

NOTE: Although the clearEvent() method is defined for large range, only the allowed events are tested here, because a range is reserved for propriatary use in TS 102 241 [9], clause 4, and a range is omitted for compatibility with future releases of TS 102 241 [9].

5.2.8.4 Method disableMenuEntry

Test Area Reference: Api_2_Tkr_Dmet.

5.2.8.4.1 Conformance requirement:

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

```
public void disableMenuEntry(byte id)
                     throws ToolkitException
```

5.2.8.4.1.1 Normal execution

- CRRN1: This method does not modify the registration state to the EVENT_MENU_SELECTION.
- CRRN2: This method does not modify the registration state to the EVENT_MENU_SELECTION_HELP_REQUEST.
- CRRN3: After invocation of this method, during the current card session, the CAT Runtime Environment 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 CAT Runtime Environment shall issue a SETUP MENU proactive command containing Item Data Object for Item 1 TLV with a length of zero and no value part.

5.2.8.4.1.2 Parameter errors

No requirements.

5.2.8.4.1.3 Context errors

- CRRC1: shall throw a ToolkitException with reason ENTRY_NOT_FOUND if the menu entry does not exist for this applet.

5.2.8.4.2 Test area files

Test Source: Test_Api_2_Tkr_Dmet.java.

Test Applet: Api_2_Tkr_Dmet_1.java.

Cap File: api_2_tkr_dmet.cap.

- 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'.
- Additional requirements for the UICC personalization:
  - content of EF sume shall be:
    - Title Alpha Identifier: "TOOLKIT TEST".

5.2.8.4.3 Test coverage

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

5.2.8.4.4 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 initialize the card 2- Call isEventSet(EVENT_MENU_SELECTION) 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	1- Shall return true 2- Shall return false	1- The UICC 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- Call disableMenuEntry('01') 2- Call isEventSet(EVENT_MENU_SELECTION) 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	1- No exception shall be thrown. 2- Shall return true. 3- Shall return false.	3- The UICC 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- Call isEventSet(EVENT_MENU_SELECTION) 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	2- Shall return true 3- Shall return true	3- The UICC shall issue a SET UP MENU proactive command with entry '02', indicating help supported.

Id	Description	API Expectation	APDU Expectation
4	**Check the menu after disabling a previously enabled entry registered to EVENT_MENU_SELECTION_HELP_REQUEST** 1- Call disableMenuEntry('02') 2- Call isEventSet(EVENT_MENU_SELECTION) 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	1- No exception shall be thrown. 2- Shall return true. 3- Shall return true.	3- The UICC 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.	

5.2.8.5 Method enableMenuEntry

Test Area Reference: Api_2_Tkr_Emet.

5.2.8.5.1 Conformance requirement:

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

```
public void enableMenuEntry(byte id)
                    throws ToolkitException
```

5.2.8.5.1.1 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 CAT Runtime Environment shall dynamically issue a SETUP MENU proactive command which does contain an ITEM COMPREHENSION TLV object for this entry.

5.2.8.5.1.2 Parameter errors

No requirements.

5.2.8.5.1.3 Context errors

- CRRC1: shall throw a ToolkitException with reason MENU_ENTRY_NOT_FOUND if the menu entry doesn't exist for this applet.

5.2.8.5.2 Test area files

Additional requirements for the UICC personalization:

- content of EF sume shall be:
  - Title Alpha Identifier: "TOOLKIT TEST".
  - Test Source: Test_Api_2_Tkr_Emet.java.
  - Test Applet: Api_2_Tkr_Emet_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'.
  - Cap File: api_2_tkr_emet.cap.

5.2.8.5.3 Test coverage

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

5.2.8.5.4 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- Call isEventSet(EVENT_MENU_SELECTION) 2- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST) 3- Call disableMenuEntry('01')	1- Shall return true 2- Shall return false 3- No exception shall be thrown.	3- The UICC 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- Call enableMenuEntry('01') 2- Call isEventSet(EVENT_MENU_SELECTION) 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	1- No exception shall be thrown. 2- Shall return true. 3- Shall return false.	3- The UICC 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- Call isEventSet(EVENT_MENU_SELECTION) 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST) 4- Call disableMenuEntry('02')	2- Shall return true 3- Shall return true 4- No exception shall be thrown	4- The UICC shall issue a SET UP MENU proactive command with entry '01'.
4	**Check menu state after enabling a previously enabled entry registered to EVENT_MENU_SELECTION_HELP_REQUEST** 1- Call enableMenuEntry('02'). 2- Call isEventSet(EVENT_MENU_SELECTION) 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	1- No exception shall be thrown. 2- Shall return true. 3- Shall return true.	3- The UICC shall issue a SET UP MENU proactive command with entries '01' and '02' indicating help supported.

Id	Description	API Expectation	APDU Expectation
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.	

5.2.8.6 Method getPollInterval

Test Area Reference: Api_2_Tkr_Gpol.

5.2.8.6.1 Conformance requirement:

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

```
public short getPollInterval()
```

5.2.8.6.1.1 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.

5.2.8.6.1.2 Parameter errors

No requirements.

5.2.8.6.1.3 Context errors

No requirements.

5.2.8.6.2 Test area files

Test Source: Test_Api_2_Tkr_Gpol.java.

Test Applet: Api_2_Tkr_Gpol_1.java.

Cap File: api_2_tkr_gpol.cap.

5.2.8.6.3 Test coverage

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

5.2.8.6.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Applet isn't registered to EVENT_STATUS_COMMAND** Call getPollInterval() method.	Shall return 0.	
2	**Requesting max duration** 1- Call requestPollInterval(15300) 2- Reset and initialize the card 3- Call getPollInterval() method	1- No exception shall be thrown. 3- Shall return a value between 1 and 15300.	

Id	Description	API Expectation	APDU Expectation
3	**Requesting System Duration** 1- Call requestPollInterval(POLL_SYSTEM_DURATION) 2- Reset and initialize the card 3- Call getPollInterval() method.	1- No exception shall be thrown. 3- Shall return a value between 1 and 15300.	
4	**Requesting no Duration** 1- Call requestPollInterval(POLL_NO_DURATION) 2- Reset and initialize the card 3- Call getPollInterval() method.	1- No exception shall be thrown. 3- Shall return 0.	

5.2.8.7 Method initMenuEntry

Test Area Reference: Api_2_Tkr_Imet.

5.2.8.7.1 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
```

5.2.8.7.1.1 Normal execution

- CRRN1: The CAT Runtime Environment 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 CAT Runtime Environment shall use the data of the EFsume 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 UICC for this entry, then the CAT Runtime Environment shall trigger the applet.
- CRRN5: if help supported was true, the CAT Runtime Environment 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 CAT Runtime Environment shall supply in the SET UP MENU command with the icon identifier provided in the icon identifier list within the item icon identifier list Comprehension TLV if all the applets registered to the EVENT_MENU_SELECTION provide it.
- CRRN8: The CAT Runtime Environment 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 CAT Runtime Environment shall issue a SETUP MENU proactive command containing an Items Next Action Indicator Comprehension 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 UICC for this identifier, then the CAT Runtime Environment shall trigger the applet.

5.2.8.7.1.2 Parameter errors

- 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.

5.2.8.7.1.3 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 initialized (eg no more item data in applet loading parameter).

5.2.8.7.2 Test area files

Additional requirements for the UICC personalization:

- content of EFsume shall be:
  - Title Alpha Identifier: "TOOLKIT TEST"
  - Test case trigger:
    - 1- Applet instanciation.
    - 2- Menu selection.
    - 3- Menu selection Help Supported.
  - Test Source: Test_Api_2_Tkr_Imet.java
  - Test Applet: Api_2_Tkr_Imet_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'.
  - Cap File: api_2_tkr_imet.cap.

5.2.8.7.3 Test coverage

CRR number	Test case number
N1	16
N2	9
N3	11
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

5.2.8.7.4 Test procedure

Id	Description	API Expectation	APDU Expectation

1	**NULL as parameter to menuEntry**
	`initMenuEntry()`
	`MenuEntry = NULL`
		Shall throw a java.lang.NullPointerException.	

2	**Offset > menuEntry.length**
	`initMenuEntry()`
	`MenuEntry = "ToolkitTest"`
	`Offset = 12`
	`Length = 0`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

3	**Offset < 0**
	`initMenuEntry()`
	`MenuEntry = "ToolkitTest"`
	`Offset = -1`
	`Length = 11`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

4	**Offset = 255**
	`initMenuEntry()`
	`MenuEntry = "ToolkitTest"`
	`Offset = 255`
	`Length = 11`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

5	**Length = menuEntry.length+1**
	`initMenuEntry()`
	`MenuEntry = "ToolkitTest"`
	`Offset = 0`
	`Length = 12`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

6	**Length < 0**
	`initMenuEntry()`
	`MenuEntry = "ToolkitTest"`
	`Offset = 0`
	`Length = -1`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

7	**Offset + length > menuEntry.length**
	`initMenuEntry()`
	`MenuEntry = "ToolkitTest"`
	`Offset = 11`
	`Length = 1`
		Shall throw java.lang.ArrayIndexOutOfBoundsException.	

Id	Description	API Expectation	APDU Expectation
8	**MenuEntry.length > size allocated at loading for each menu entry** initMenuEntry() MenuEntry = "ToolkitTest impossible" Offset = 0 Length = 16	ALLOWED_LENGTH_EXCEEDED ToolkitException is thrown.	
9	**Successful call, menuEntry is the whole buffer** 1- Call initMenuEntry() method MenuEntry = "TOOLKIT TEST 1" Offset = 0 Length = 14 NextAction = '00' HelpSupported = false IconQualifier = '00' IconIdentifier = 0 2- Call 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- Call initMenuEntry() method MenuEntry = "1234567TOOLKIT TEST 2" Offset = 7 Length = 14 NextAction = '00' HelpSupported = false IconQualifier = '00' IconIdentifier = 0 2- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	1- No exception shall be thrown,Shall return ID '02'. 2- Shall return false.	
11	**Successful call, menuEntry with help supported** 1- Call initMenuEntry() method MenuEntry = "TOOLKIT TEST 3" Offset = 0 Length = 14 NextAction = '00' HelpSupported = true IconQualifier = '00' IconIdentifier = 0 2- Call 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** initMenuEntry() 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'	

Id	Description	API Expectation	APDU Expectation
13	**Successful call, menuEntry with a next action indication** initMenuEntry() 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** Call initMenuEntry() method MenuEntry = "ToolkitTest" Offset = 0 Length = 0 NextAction = '00' HelpSupported = false IconQualifier = '00' IconIdentifier = 0	No exception shall be thrown. hall return ID '06'.	
15	**Initialize more entry than allocated at loading** initMenuEntry() MenuEntry = "ToolkitTest" Offset = 0 Length = 11	REGISTRY_ERROR ToolkitException is thrown.	
16	**Dynamic update of the menu stored by the ME** Fetch the setup menu proactive command		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 triggered 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 triggered 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 triggered 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 triggered by an ENVELOPE(MENU_SELECTION) command & Menu Entry ID = '04'	

Id	Description	API Expectation	APDU Expectation
21	**Check Applet is triggered by envelope (MENU_SELECTION) command** `Menu Entry ID = '05'`	Applet is triggered 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 triggered 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 triggered by an ENVELOPE(MENU_SELECTION) command & Menu Entry ID = '06'	

5.2.8.8 Method isEventSet

Test Area Reference: Api_2_Tkr_Ievs.

5.2.8.8.1 Conformance requirement:

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

```
public boolean isEventSet(short event)
```

5.2.8.8.1.1 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.

5.2.8.8.1.2 Parameter errors

No requirements.

5.2.8.8.1.3 Context errors

No requirements.

5.2.8.8.2 Test area files

Test Source: Test_Api_2_Tkr_Ievs.java

Test Applet: Api_2_Tkr_Ievs_1.java

Api_2_Tkr_Ievs_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.

Cap File: api_2_tkr_ievs.cap.

5.2.8.8.3 Test coverage

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

5.2.8.8.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Install Applet1 only registered to EVENT_UNRECOGNIZED_ENVELOPE and EVENT_MENU_SELECTION** **Test that events aren't set** Applet calls isEventSet() method for each event ranging from (-1, 1, 7 to 9, 11 to 23, 25 to 29, 123, 124, 126 and 127)* excepted EVENT_UNRECOGNIZED_ENVELOPE(-1) and EVENT_MENU_SELECTION(7).	Shall return false each time.	
2	**For EVENT_UNRECOGNIZED_ENVELOPE** Call isEventSet(EVENT_UNRECOGNIZED_ENVELOPE)	Shall return true.	
3	**For EVENT_MENU_SELECTION** Call isEventSet(EVENT_MENU_SELECTION)	Shall return true	
4	**After clearing EVENT_UNRECOGNIZED_ENVELOPE** 1- Call clearEvent(EVENT_UNRECOGNIZED_ENVELOPE) 2- Call isEventSet(EVENT_UNRECOGNIZED_ENVELOPE)	1- No exception shall be thrown. 2- Shall return false.	
5	**Setting events** For all allowed events defined in ETSI TS 102 241[9] for setEvent()method: EVENT_PROFILE_DOWNLOAD, EVENT_CALL_CONTROL_BY_NAA, 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_EVENT_DOWNLOAD_ACCESS_TECHNOL0GY_CHANGE, EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED, EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE, EVENT_EVENT_DOWNLOAD_BROWSING_STATUS, EVENT_PROACTIVE_HANDLER_AVAILABLE, EVENT_APPLICATION_DESELECT, EVENT_FIRST_COMMAND_AFTER_ATR, EVENT_UNRECOGNIZED_ENVELOPE applet calls: 1- Call setEvent() method 2- Call isEventSet() method	1- No exception shall be thrown. 2- Shall return true each time.	

Id	Description	API Expectation	APDU Expectation
6	**For EVENT_MENU_SELECTION_HELP_REQUEST** 1- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST) 2- Call changeMenuEntry() with help supported 3- Call isEventSet(EVENT_MENU_SELECTION_HELP_REQUEST)	1- Shall return false. 2- Shall return true.	
7	**For EVENT_TIMER_EXPIRATION** 1- Call isEventSet(EVENT_TIMER_EXPIRATION) 2- Call allocateTimer() 3- Call isEventSet(EVENT_TIMER_EXPIRATION)	1- Shall return false. 3- Shall return true.	
8	**For EVENT_STATUS_COMMAND** Call isEventSet(EVENT_STATUS_COMMAND) Call requestPollInterval(POLL_SYSTEM_DURATION) Call isEventSet(EVENT_STATUS_COMMAND)	1- Shall return false. 3- Shall return true.	
9	**For EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION** 1- Call isEventSet (EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION) 2- Call allocateServiceIdentifier() 3- Call isEventSet (EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION)	1- Shall return false. 3- Shall return true.	
10	**Install Applet2 only registered to EVENT_MENU_SELECTION** Call isEventSet(EVENT_UNRECOGNIZED_ENVELOPE)	Shall return false.	

NOTE: Although the method isEventSet() is defined for a large range only the allowed events are tested, because a range is reserved for propriatary use in ETSI TS 102 241 [9], clause 4, and a range is omitted for compatibility with future releases of ETSI TS 102 241 [9].

5.2.8.9 Method releaseTimer

Test Area Reference: Api_2_Tkr_Rtim.

5.2.8.9.1 Conformance requirement:

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

```
public void releaseTimer(byte timerIdentifier)
                  throws ToolkitException
```

5.2.8.9.1.1 Normal execution

- CRRN1: Release a Timer that has been allocated to the calling applet.
- CRRN2: After invocation of the method the indicated timer shall be released and available for reallocation.
- CRRN3: The applet is deregistered of the EVENT_TIMER_EXPIRATION for the indicated Timer Identifier.

5.2.8.9.1.2 Parameter errors

- CRRP1: shall throw a ToolkitException with INVALID_TIMER_ID reason if the timer identifier isn't between 1 and 8.

5.2.8.9.1.3 Context errors

- CRRC1: shall throw a ToolkitException with INVALID_TIMER_ID reason if the timer is not allocated to this applet.

5.2.8.9.2 Test area files

Test Source: Test_Api_2_Tkr_Rtim.java.

Test Applet: Api_2_Tkr_Rtim_1.java.

Cap File: api_2_tkr_rtim.cap.

- Installation parameter:
  - As Default, except max timer which is set to 8.

5.2.8.9.3 Test coverage

CRR number	Test case number
N1	2, 3, 4
N2	5, 6
N3	7
N4	7
P1	1, 3
C1	CAT Runtime Environment, Cre_Pcs_Pcco

5.2.8.9.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Release 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	**Release allocated timers**
	`1- Call 8 times allocateTimer()`
	`2- Call 7 times releaseTimer(id)`
	`3- Call isEventSet(EVENT_TIMER_EXPIRATION)`
		1- No exception shall be thrown.
		2- Each time, no exception shall be thrown.
		3- Shall return true	

3	**Release invalid timer ID**
	`1- Call releaseTimer('FF') method`
	`2- Call isEventSet(EVENT_TIMER_EXPIRATION)`
		1- Shall throw a ToolkitException with INVALID_TIMER_ID reason code.
		2- Shall return true.	

4	**Release last timer**
	`1- Call releaseTimer(last timer allocated)`
	`2- Call isEventSet(EVENT_TIMER_EXPIRATION)`
		1- No exception shall be thrown.
		2- Shall return false.	

5	**Check we can allocate timers after they have been released** `Call 8 times allocateTimer() method`	No exception shall be thrown.	
6	**Release all timers** `For 1 to 8, Call releaseTimer(id)`	No exception shall be thrown.	

Id	Description	API Expectation	APDU Expectation
7	**Check applet is not triggered by envelope(EVENT_TIMER_EXPIRATION) command** Send envelope(EVENT_TIMER_EXPIRATION)	Applet is not triggered by any envelope(EVENT_TIMER_EXPIRATION) command	

5.2.8.10 Method requestPollInterval

Test Area Reference: Api_2_Tkr_Rpol.

5.2.8.10.1 Conformance requirement

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

```
public void requestPollInterval(short duration)
                        throws ToolkitException
```

5.2.8.10.1.1 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.

5.2.8.10.1.2 Parameter errors

- CRRP1: the method should throw a ToolkitException with REGISTRY_ERROR reason if duration is > 15300 or is < -1 (POLL_SYSTEM_DURATION).

5.2.8.10.1.3 Context errors

No requirements.

5.2.8.10.2 Test area files

Test Source: Test_Api_2_Tkr_Rpol.java.

Test Applet: Api_2_Tkr_Rpol_1.java.

Cap File: api_2_tkr_rpol.cap.

5.2.8.10.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4
N2	6, 7
P1	5

5.2.8.10.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Request a value between 1 and 15300 s** 1- Call isEventSet(EVENT_STATUS_COMMMAND) 2- Call requestPollInterval(duration) for boundaries values: 1, 255, 256, 15300. 3- Call isEventSet(EVENT_STATUS_COMMAND).	1- Shall return false. 2- No exception shall be thrown. 3- Shall return true.	

Id	Description	API Expectation	APDU Expectation
2	**Check Applet is triggered by a STATUS command** 1- reset and card initialization 2- Send STATUS command	2- Applet is triggered by a STATUS command	
3	**Request POLL SYSTEM DURATION** 1- Call isEventSet(EVENT_STATUS_COMMMAND). 2- Call RequestPollInterval(POLL_SYSTEM_DURATION). 3- Call IsEventSet(EVENT_STATUS_COMMAND).	1- Shall return true. 2- No exception shall be thrown. 3- Shall return true.	
4	**Check Applet is triggered by a STATUS command** 1- reset and card initialization 2- Send STATUS command	2- Applet is triggered by a STATUS command	
5	**Request invalid duration** Call requestPollInterval(duration) for following values: 15301, 32767, -2, -32768	Each time, a ToolkitException with REGISTRY_ERROR reason code, shall be thrown.	
6	**Request POLL NO DURATION** 1- Call isEventSet(EVENT_STATUS_COMMMAND) 2- Call requestPollInterval(POLL_NO_DURATION) 3- Call isEventSet(EVENT_STATUS_COMMAND)	1- Shall return true. 2- No exception shall be thrown. 3- Shall return false.	
7	**Check Applet is not triggered by an STATUS command.** 1- reset and card initialization 2- Send STATUS command	2- Applet is not triggered by a STATUS command	

5.2.8.11 Method setEvent

Test Area Reference: Api_2_Tkr_Sevt

5.2.8.11.1 Conformance requirement

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

```
public void setEvent(short id)
             throws ToolkitException,
                    javacard.framework.TransactionException
```

5.2.8.11.1.1 Normal execution

- CRRN1: A following call to isEventSet() method with the same event id shall answer true for the applet.
- CRRN2: The CAT Runtime Environment shall trigger the applet if an occurrence of the set event happens.
- CRRN3: the method shall accept all the events defined in TS 102 241 [9] except: EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION, EVENT_STATUS_COMMAND, EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION and EVENT_EXTERNAL_FILE_UPDATE.
- 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.

5.2.8.11.1.2 Parameter errors

- 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.
- CRRP6: shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event was EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION.
- CRRP7: shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event was EVENT_EXTERNAL_FILE_UPDATE.

5.2.8.11.1.3 Context errors

- CRRC1: shall throw a ToolkitException with EVENT_ALREADY_REGISTERED if event is EVENT_CALL_CONTROL_BY_NAA but another applet is already registered to it.
- CRRC2: shall throw a ToolkitException with EVENT_ALREADY_REGISTERED if event is EVENT_CALL_CONTROL_BY_NAA but another applet that it is not in selectable state is already registered to it.
- CRRC3: shall throw javacard.framework.TransactionException - if the operation would cause the commit capacity to be exceeded.
- CRRC4: shall throw a ToolkitException with TAR_NOT_DEFINED if the event requests a tag and the applet has no TAR defined.

5.2.8.11.2 Test area files

Test Source: Test_Api_2_Tkr_Sevt.java.

Test Applet: Api_2_Tkr_Sevt_1.java.

Api_2_Tkr_Sevt_2.java.

Api_2_Tkr_Sevt_3.java.

Api_2_Tkr_Sevt_4.java.

The load script installs the 4 instances.

Cap File: api_2_tkr_sevt.cap.

5.2.8.11.3 Test coverage

CRR number	Test case number
N1	2
N2	1, 10, 11
N3	2, 4, 5, 6, 7, 8, 9
N4	14
N5	not testable
P1	3
P2	4
P3	5

CRR number	Test case number
P4	6
P5	7
P6	8
P7	9
C1	12
C2	13
C3	not testable
C4	not testable

5.2.8.11.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Applet1 is triggered by envelope (EVENT_EVENT_DOWNLOAD_USER_ACTIVITY) command.** Send ENVELOPE(EVENT_EVENT_DOWNLOAD_USER_ACTIVITY)	Applet1 shall be triggered	
2	**Set ALLOWED and SUPPORTED events** 1- For all allowed events (-1, 1, 7 to 9, 11 to 23, 25 to 29, 123, 124, 126 and 127 excepted 7, 8, 11, 19, 27, 124) defined in ETSI TS 102 241 []*: EVENT_PROFILE_DOWNLOAD, EVENT_CALL_CONTROL_BY_NAA, 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_EVENT_DOWNLOAD_ACCESS_TECHNOL0GY_CHANGE, EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED, EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE, EVENT_EVENT_DOWNLOAD_BROWSING_STATUS, EVENT_PROACTIVE_HANDLER_AVAILABLE, EVENT_APPLICATION_DESELECT, EVENT_FIRST_COMMAND_AFTER_ATR, EVENT_UNRECOGNIZED_ENVELOPE 1.1- Call clearEvent(event) 1.2- Call isEventSet(event) 1.3- Call setEvent(event) 1.4- Call isEventSet(event) 1.5- Call 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	**Set Event 0** Call setEvent(0)	Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason code.	

Id	Description	API Expectation	APDU Expectation
4	**Set EVENT_MENU_SELECTION** `Call setEvent(EVENT_MENU_SELECTION)`	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
5	**Set EVENT_MENU_SELECTION_HELP_REQUEST** `Call setEvent(EVENT_MENU_SELECTION_HELP_REQUEST)`	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
6	**Set EVENT_TIMER_EXPIRATION** `Call setEvent(EVENT_TIMER_EXPIRATION)`	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
7	**Set EVENT_STATUS_COMMAND** `Call setEvent(EVENT_STATUS_COMMAND)`	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
8	**Set EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION** `Call setEvent(EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION)`	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
9	**Set EVENT_EXTERNAL_FILE_UPDATE** `Call setEvent(EVENT_EXTERNAL_FILE_UPDATE)`	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
10	**Set EVENT_CALL_CONTROL_BY_NAA** `Call setEvent(EVENT_CALL_CONTROL_BY_NAA)`	No Exception shall be thrown	
11	**Check applet is triggered by envelope (CALL_CONTROL_BY_NAA) command** `Trigger Applet1`	Applet1 is triggered by an ENVELOPE(CALL_CONTROL_BY_NAA)	
12	**Applet2 registers to EVENT_CALL_CONTROL_BY_NAA but it is already assigned to another applet** `Applet2 call setEvent(EVENT_CALL_CONTROL_BY_NAA)`	Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	

13	**Applet3 registers to EVENT_CALL_CONTROL_BY_NAA but it is already assigned to another applet in not selectable state**
	`1- Set Applet1 in the lock state`
	`2- Trigger Applet3 which calls setEvent(EVENT_CALL_CONTROL_BY_NAA)`
	`3- Set Applet1 in the make selectable state`
		2- Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	

14	**Applet4 registers multiple registration to the same event**
	`1- setEvent(EVENT_EVENT_DOWNLOAD_MT_CALL)`
	`2- setEvent(EVENT_EVENT_DOWNLOAD_MT_CALL)`
	`3- isEventSet(EVENT_EVENT_DOWNLOAD_MT_CALL)`
		1- no exception should be thrown
		2- no exception should be thrown
		3- method should return true	

NOTE: Although the method setEvent is defined for large range only the allowed events are tested, because a range is reserved for propriatary use in ESTI TS 102 241 [9], clause 4, and a range is omitted for compatibility with future releases of ETSI TS 102 241 [9].

5.2.8.12 Method setEventList

Test Area Reference: Api_2_Tkr_Sevl.

5.2.8.12.1 Conformance requirement

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

```
public void setEventList(short[] eventList,
                         short offset,
                         short length)
                  throws java.lang.NullPointerException,
                         java.lang.ArrayIndexOutOfBoundsException,
                         ToolkitException,
                         javacard.framework.TransactionException
```

5.2.8.12.1.1 Normal execution

- CRRN1: For all events set successfully by this method, sets an event list in the Toolkit Registry entry of the applet.
- CRRN2: The CAT Runtime Environment shall trigger the applet if an occurrence of one of the successfully registered events happens.
- CRRN3: All updates on the ToolkitRegistry are atomic.
- CRRN4: No exception shall be thrown if the applet registers more than once to the same event.

5.2.8.12.1.2 Parameter errors

- 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.
- CRRP10: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event was EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION.
- CRRP11: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event was EVENT_EXTERNAL_FILE_UPDATE.

5.2.8.12.1.3 Context errors

- CRRC1: shall throw a ToolkitException with EVENT_ALREADY_REGISTERED if eventList contains EVENT_CALL_CONTROL_BY_NAA but another applet is already registered to it.
- CRRC2: shall throw javacard.framework.TransactionException - if the operation would cause the commit capacity to be exceeded.
- CRRC3: shall throw a ToolkitException with TAR_NOT_DEFINED if the eventList contains an event that requests a tag and the applet has not at least one TAR value assigned.

5.2.8.12.2 Test area files

Test Source: Test_Api_2_Tkr_Sevl.java.

Test Applet: Api_2_Tkr_Sevl_1.java.

Api_2_Tkr_Sevl_2.java.

Cap File: api_2_tkr_sevl.cap.

5.2.8.12.3 Test coverage

CRR number	Test case number
N1	1, 2
N2	18, 19
N3	21
N4	22
P1	3
P2	4, 5, 6
P3	7, 8, 9
P4	10
P5	11
P6	12
P7	13
P8	14
P9	15
P10	16
P11	17
C1	20
C2	not testable
C3	not testable

5.2.8.12.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Applet1 registering all eventList buffer** Applet1 is triggered by an envelope(MENU_SELECTION) (Id = 01) EventList = all allowed events (-1, 1, 7 to 9, 11 to 23, 25 to 29, 123, 124, 126 and 127 excepted 7, 8, 11, 19, 27, 124) defined in TS 102 241 [9]: EVENT_PROFILE_DOWNLOAD, EVENT_CALL_CONTROL_BY_NAA, 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_EVENT_DOWNLOAD_ACCESS_TECHNOL0GY_CHANGE, EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED, EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE, EVENT_EVENT_DOWNLOAD_BROWSING_STATUS, EVENT_PROACTIVE_HANDLER_AVAILABLE, EVENT_APPLICATION_DESELECT, EVENT_FIRST_COMMAND_AFTER_ATR, EVENT_UNRECOGNIZED_ENVELOPE 1- For each event in EventList, clearEvent(event) 2- Call setEventList(eventList) Offset = 0 Length = eventList.length 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 102 241 [9] (see test case 1). 1- setEventList(eventList, offset, length) Offset > 0 Length = eventList.lentgh - offset 2- For all events in eventList: Call isEventSet(event) 3- For each event in EventList: clearEvent(event)	1- No exception shall be thrown. 2- Each time shall return true for events ranging from offset to offset+length else shall return false. 3- No exception shall be thrown.	
3	**Null buffer** setEventList() EventList = null	Shall throw a java.lang.NullPointerException Exception	
4	**Out of bounds offset** setEventList() 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** setEventList() Offset = 255 Length = 1	Shall throw a java.lang.ArrayIndexOutOfBounds Exception	
6	**Offset < 0** setEventList() Offset = -1 Length = 1	Shall throw a java.lang.ArrayIndexOutOfBounds Exception	
7	**Out of bounds length** setEventList() Offset = 0 Length = eventList.length + 1	Shall throw a java.lang.ArrayIndexOutOfBounds Exception	
8	**Out of bounds and big length** setEventList() Offset = 0 Length = 255	Shall throw a java.lang.ArrayIndexOutOfBounds Exception	
9	**Length < 0** setEventList() Offset = 0 Length = -1	Shall throw a java.lang.ArrayIndexOutOfBounds Exception	
10	**Out of bounds offset + Length** setEventList() 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	**EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION** Call setEventList(eventList) with eventList indicating EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION	Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	
17	**EVENT_EXTERNAL_FILE_UPDATE** Call setEventList(eventList) with eventList indicating EVENT_EXTERNAL_FILE_UPDATE	Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	

Id	Description	API Expectation	APDU Expectation
18	**Sett EVENT_CALL_CONTROL_BY_NAA** Call setEventList(MonoEventList, 0, 1) with MonoEventList containing EVENT_CALL_CONTROL_BY_NAA	Shall not throw an exception.	
19	**Check applet is triggered by an ENVELOPE (CALL_CONTROL_BY_NAA)** Reset and initialize the card Trigger Applet1	Applet is triggered by an ENVELOPE(CALL_CONTROL_BY_NAA)	
20	**Applet2 registers to CALL_CONTROL_BY_NAA but it is already assigned** Applet2 is triggered by an envelope(MENU_SELECTION) (Id=02) 1- Call setEventList(MonoEventList,0,1) with MonoEventList containing EVENT_CALL_CONTROL_BY_NAA	1- Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	
21	**Atomicity** 1- Call setEventList(EVENT_CALL_CONTROL_BY_NAA, EVENT_EVENT_DOWNLOAD_CALL_CONNECTED) 2- isEventSet (EVENT_EVENT_DOWNLOAD_CALL_CONNECTED)	1- Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code. 2- method shallreturn false	
22	**Multiple registration to the same event** 1- setEventList(EVENT_EVENT_DOWNLOAD_MT_CALL, EVENT_EVENT_DOWNLOAD_MT_CALL) 2- isEventSet(EVENT_EVENT_DOWNLOAD_MT_CALL)	1- no exception should be thrown 2- method shall return true	

5.2.8.13 Method allocateServiceIdentifier

Test Area Reference: Api_2_Tkr_Asid.

5.2.8.13.1 Conformance requirement

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

```
public byte allocateServiceIdentifier()
                              throws ToolkitException,
                                     javacard.framework.TransactionException
```

5.2.8.13.1.1 Normal execution

- CRRN1: The returned service identifier shall be between 00 and 07 inclusive.
- CRRN2: The returned service identifier shall be different from a previously allocated but not released one.
- CRRN3: By calling this method the applet is registered to the EVENT_EVENT_ DOWNLOAD_LOCAL_CONNECTION of the allocated service.
- CRRN4: The service identifier is allocated by the applet until it explicitly releases it.
- CRRN5: When an applet allocates a service identifier, it can issue the proactive command DECLARE SERVICE to add or delete a service to the terminal service database.

5.2.8.13.1.2 Parameter errors

No requirements.

5.2.8.13.1.3 Context errors

- CRRC1: Shall throw a ToolkitException with reason NO_SERVICE_ID_AVAILABLE if all the services are allocated.
- CRRC2: Shall throw a ToolkitException with reason NO_SERVICE_ID_AVAILABLE if the maximum number of services Identifiers have been allocated to this applet according to installation parameter.

5.2.8.13.2 Test area files

Test Source: Test_Api_2_Tkr_Asid.java.

Test Applet: Api_2_Tkr_Asid_1.java: 8 services.

Api_2_Tkr_Asid_2.java: 4 services.

Api_2_Tkr_Asid_3.java: 0 services.

Cap File: api_2_tkr_asid.cap.

5.2.8.13.3 Test coverage

CRR number	Test case number
N1	1
N2	1
N3	2
N4	2
N5	Cat Runtime Environment, Cre_Pcs_Pcco
C1	3
C2	4

5.2.8.13.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Allocates up to 8 services (Applet1)** Applet1 is triggered by an envelope(EVENT_MENU_SELECTION) with Item = 01. Call 8 times: - allocateServiceIdentifier() - send associated DECLARE SERVICE - isEventSet (EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION).	No exception shall be thrown. Service ID returned shall be between 00 and 07 inclusive. It shall be different after each call. Shall return true.	
2	**Check Applet1 is triggered by envelope (EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION) command** 1- Send 8 envelopes (EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION) with all services id (not in an increase order). 2- Call releaseServiceIdentifier(id) with all services. 3- Call isEventSet (EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION) method	1- Applet1 shall be triggered each time. 3- returns false.	

Id	Description	API Expectation	APDU Expectation
3	**Allocate services more than the maximum** Applet1 is triggered by an envelope(EVENT_MENU_SELECTION) with Item = 01. 1- Applet1 calls 5 times allocateServiceIdentifier() method Applet2 is triggered by an envelope(EVENT_MENU_SELECTION) with Item = 02 2- Applet2 calls 3 times allocateServiceIdentifier() method 3- Applet2 calls 1 more allocateServiceIdentifier() method Applet1 is triggered by an envelope(EVENT_MENU_SELECTION) with Item = 01. 4- Applet1 releases all its services.	1- No exception shall be thrown. Each time, the returned service identifier shall be between '00' and '07' inclusive. It shall be different after each call. Applet1 finalizes. 2- No exception shall be thrown. Each time, the returned service identifier shall be between '00' and '07' inclusive. It shall be different after each call it shall be different from the ones allocated to Applet1. 3- A ToolkitException with reason NO_SERVICE_ID_AVAILABLE shall be thrown Applet2 finalizes. 4- No exception is thrown. Applet1 finalizes.	
4	**Allocate services more than the maximum to this applet (Applet3 and Applet2)** Applet3 is triggered by an envelope(EVENT_MENU_SELECTION) with Item = 03 1- Applet3 calls allocateServiceIdentifier() method. Applet2 is triggered by an envelope(EVENT_MENU_SELECTION) with Item = 02 2- Applet2 calls allocateServiceIdentifier() method 3- Applet2 calls allocateServiceIdentifier() again	1- A ToolkitException with reason NO_SERVICE_ID_AVAILABLE shall be thrown Applet3 finalizes. 2- No exception shall be thrown. 3- A ToolkitException with reason NO_SERVICE_ID_AVAILABLE shall be thrown	

5.2.8.14 Method releaseServiceIdentifier

Test Area Reference: Api_2_Tkr_Rsid.

5.2.8.14.1 Conformance requirement

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

```
public void releaseServiceIdentifier(byte serviceIdentifier)
                            throws ToolkitException,
                                   javacard.framework.TransactionException
```

5.2.8.14.1.1 Normal execution

- CRRN1: Release a Service Identifier that has been allocated to the calling applet.
- CRRN2: The applet is deregistered of EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION for the indicated service identifier.

5.2.8.14.1.2 Parameter errors

- CRRP1: shall throw a ToolkitException with INVALID_SERVICE_ID reason if the service identifier is not between 0 and 7.

5.2.8.14.1.3 Context errors

- CRRC1: shall throw a ToolkitException with INVALID_SERVICE_ID reason if the service is not allocated to this applet.

5.2.8.14.2 Test area files

Test Source: Test_Api_2_Tkr_Rsid.java.

Test Applet: Api_2_Tkr_Rsid_1.java: 8 services.

Api_2_Tkr_Rsid_2.java: 1 service.

Cap File: api_2_tkr_rsid.cap.

5.2.8.14.3 Test coverage

CRR number	Test case number
N1	4, 5, 6
N2	2, 3, 4, 7
P1	1, 3
C1	8

5.2.8.14.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Release not allocated services** Applet1 is triggered by envelope(EVENT_MENU_SELECTION) with Item = 1 For each service ID ranging from '00' to 'FF', applet calls releaseServiceIdentifier(ID)	Each time, method shall throw a ToolkitException with reason code INVALID_SERVICE_ID.	
2	**Release allocated services** 1- Call 8 times allocateServiceIdentifier() method. 2- Call 7 times releaseServiceIdentifier(id) method with id from 0 to 6. 3- Call isEventSet(EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION)	1- No exception shall be thrown. 2- Each time, no exception shall be thrown. 3- Shall return true	
3	**Release invalid service ID** 1- Call releaseServiceIdentifier('FF') method 2- Call isEventSet(EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION) method	1- Shall throw a ToolkitException with INVALID_SERVICE_ID reason code. 2- Shall return true.	
4	**Release last service** 1- Call releaseServiceIdentifier() method with id = '07' 2- Call isEventSet(EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION)	1- No exception shall be thrown. 2- Shall return false.	

Id	Description	API Expectation	APDU Expectation
5	**Released services can be allocated** 1- Applet1 calls 8 times allocateServiceIdentifier() method. 2- Applet1 calls releaseServiceIdentifier() method with the service Id = 1 Applet2 is triggered by envelope(EVENT_MENU_SELECTION) with Item = 2 3- Applet2 calls allocateServiceIdentifier() method.	1- No exception shall be thrown. 2- No exception shall be thrown. Applet1 finalizes 3- No exception shall be thrown, the service Id allocated shall be 1	
6	**Release all services** Applet1 is triggered by envelope(EVENT_MENU_SELECTION) with Item = 1 Applet1 calls releaseServiceIdentifier(id) method for id 0 and 2 to 7.	No exception shall be thrown. Applet1 finalizes.	
7	**Check Applet1 is not triggered by envelope(EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION) command** Send envelope(EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION)	Applet1 is not triggered by an ENVELOPE(EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION) command. Applet2 is triggered.	
8	**Release invalid service ID** Applet1 is triggered by envelope(EVENT_MENU_SELECTION) with Item = 1 1- Applet1 calls allocateServiceIdentifier() method 7 times. 2- Applet1 calls releaseServiceIdentifier() method with id = '01'	1- No exception shall be thrown, the services Id shall be different from 01 2- Shall throw a ToolkitException with INVALID_SERVICE_ID reason code.	

5.2.8.15 Method registerFileEvent(short event, byte[] baFileList, short sOffset1, short sLength1, byte[] baADFAid, short sOffset2, byte bLength2)

Test Area Reference: Api_2_Tkr_Rgfes_Bss_Bsb.

5.2.8.15.1 Conformance requirement

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

```
public void registerFileEvent(short event,
                              byte[] baFileList,
                              short sOffset1,
                              short sLength1,
                              byte[] baADFAid,
                              short sOffset2,
                              byte bLength2)
                       throws ToolkitException,
                              java.lang.NullPointerException,
                              java.lang.ArrayIndexOutOfBoundsException,
                              javacard.framework.TransactionException,
                              javacard.framework.SystemException
```

5.2.8.15.1.1 Normal execution

- CRRN1: The only event allowed and supported by the method is EVENT_EXTERNAL_FILE_UPDATE.
- CRRN2: The CAT Runtime Environment shall trigger the applet when a elementary file included in the baFileList is updated.
- CRRN3: If the path provided indicates a dedicated file (DF), the Applet shall be triggered when an elementary file within this dedicated file is updated.
- CRRN4: The baADFAid indicates the Aid of the ADF under which the file is located.
- CRRN5: If baADFAid is null, it indicates that the file is located under the MF and not located under an ADF.
- CRRN6: A call to isEventSet() method for EVENT_EXTERNAL_FILE_UPDATE should return true if the registerFileEvent() method has been successfully called.

5.2.8.15.1.2 Parameter errors

- CRRP1: Shall throw a java.lang.NullPointerException if baFileList is null.
- CRRP2: Shall throw a java.lang.ArrayIndexOutOfBoundsException if sOffset1 or sLength1 or both would cause access outside array bounds.
- CRRP3: Shall throw a java.lang.ArrayIndexOutOfBoundsException if sOffset2 or sLength2 or both would cause access outside array bounds.
- CRRP4: Shall throw a javacard.framework.TransactionException if the operation would cause the commit capacity to be exceeded.
- CRRP5: Shall throw a javacard.framework.SystemException with ILLEGAL_VALUE reason if bLength2 is not in the range of 5 bytes to 16 bytes.
- CRRP6: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_MENU_SELECTION.
- CRRP7: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_MENU_SELECTION_HELP_REQUEST.
- CRRP8: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_TIMER_EXPIRATION.
- CRRP9: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_STATUS_COMMAND.
- CRRP10: Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason if the event is not EVENT_EXTERNAL_FILE_UPDATE.

5.2.8.15.1.3 Context errors

No requirements.

5.2.8.15.2 Test area files

Test Source: Test_Api_2_Tkr_Rgfes_Bss_Bsb.java.

Test Applet: Api_2_Tkr_Rgfes_Bss_Bsb_1.java.

Cap File: api_2_tkr_rgfes_bss_bsb.cap.

5.2.8.15.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4
N2	1, 3, 21, 22, 23, 24
N3	2, 4
N4	3, 4
N5	1, 2
N6	1
P1	5
P2	6, 7, 8, 9, 10
P3	11,12,13,14
P4	Not testable
P5	15
P6	16
P7	17
P8	18
P9	19
P10	20

5.2.8.15.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Register EF under MF**		
	1- Call isEventSet (EVENT_EXTERNAL_FILE_UPDATE) method	1- Returns false	
	2- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="02 3F0011116F03 3F0011116F09" baADFAid=null	2- No exception is thrown	
	3- Call isEventSet (EVENT_EXTERNAL_FILE_UPDATE) method	3- Returns true	
	4- Update binary on MF\DFTEST\EFTARU	4- Applet is triggered	
	5- Increase on MF\DFTEST\EFCARU	5- Applet is triggered	
	6- Update record on MF\DFTEST\EFLARU	6- Applet is not triggered	
	7- Update binary on MF\DFTEST\ DFSUB_TEST\EFTAA	7- Applet is not triggered	
	8- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="02 3F0011116F03 3F0011116F09" baADFAid=null		

Id	Description	API Expectation	APDU Expectation
2	**Register DF under MF** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F001111" baADFAid=null 2- Update binary on MF\DFTEST\EFTARU 3- Increase on MF\DFTEST\EFCARU 4- Update record on MF\DFTEST\EFLARU 5- Update binary on MF\DFTEST\ DFSUB_TEST\EFTAA 6- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F001111" baADFAid=null	1- No exception is thrown 2- Applet is triggered 3- Applet is triggered 4- Applet is triggered 5- Applet is not triggered	
3	**Register EF under ADF1** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="02 3F007FFF11116F03 3F007FFF11116F09" baADFAid="AID ADF1" 2- Update binary on ADF1\DFTEST\EFTARU 3- Increase on ADF1\DFTEST\EFCARU 4- Update record on ADF1\DFTEST\EFLARU 5- Update binary on ADF1\DFTEST\ DFSUB_TEST\EFTAA 6- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="02 3F007FFF11116F03 3F007FFF11116F09" baADFAid="AID ADF1"	1- No exception is thrown 2- Applet is triggered 3- Applet is triggered 4- Applet is not triggered 5- Applet is not triggered	
4	**Register DF under ADF1** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F007FFF1111" baADFAid="AID ADF1" 2- Update binary on ADF1\DFTEST\EFTARU 3- Increase on ADF1\DFTEST\EFCARU 4- Update record on ADF1\DFTEST\EFLARU 5- Update binary on ADF1\DFTEST\ DFSUB_TEST\EFTAA 6- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F007FFF1111" baADFAid="AID ADF1"	1- No exception is thrown 2- Applet is triggered 3- Applet is triggered 4- Applet is triggered 5- Applet is not triggered 6- No exception is thrown	

Id	Description	API Expectation	APDU Expectation
5	**NullPointerException Exception** Call registerFileEvent() method with null baFileList	Shall throw a NullPointerException	
6	**sOffset1 >= baFileList.length** Call registerFileEvent() method with baFileList.length = 8 sOffset1 = 8 sLength1 = 4	Shall throw a ArrayIndexOutOfBoundsException	
7	**sOffset1 < 0** Call registerFileEvent() method with baFileList.length = 8 sOffset1 = -1 sLength1 = 4	Shall throw a ArrayIndexOutOfBoundsException	
8	**sLength1 > baFileList.length** Call registerFileEvent() method with baFileList.length = 8 sOffset1 = 0 sLength1 = 10	Shall throw a ArrayIndexOutOfBoundsException	
9	**sOffset1 + sLength1 > baFileList.length** Call registerFileEvent() method with baFileList.length = 8 sOffset1 = 5 sLength1 = 4	Shall throw a ArrayIndexOutOfBoundsException	
10	**sLength1 < 0** Call registerFileEvent() method with baFileList.length = 8 sOffset1 = 0 sLength1 = -1	Shall throw a ArrayIndexOutOfBoundsException	
11	**sOffset2 >= baFileList.length** Call registerFileEvent() method with baADFAid.length = 15 sOffset2 = 15 bLength2 = 6	Shall throw a ArrayIndexOutOfBoundsException	
12	**sOffset2 < 0** Call registerFileEvent() method with baADFAid.length = 15 sOffset2 = -1 bLength2 = 6	Shall throw a ArrayIndexOutOfBoundsException	
13	**sLength2 > baFileList.length** Call registerFileEvent() method with baADFAid.length = 15 sOffset2 = 0 bLength2 = 16	Shall throw a ArrayIndexOutOfBoundsException	
14	**sOffset2 + sLength2 > baFileList.length** Call deregisterFileEvent() method with baADFAid.length = 15 sOffset1 = 10 bLength1 = 6	Shall throw a ArrayIndexOutOfBoundsException	
15	**ILLEGAL_VALUE Exception** 1- Call registerFileEvent() method with baADFAid.length = 18 sOffset2 = 0 bLength2 = 4 2- Call registerFileEvent() method with baADFAid.length = 18 sOffset2 = 0 bLength2 = 18	1- Shall throw a SystemException with ILLEGAL_VALUE reason code 2- Shall throw a SystemException with ILLEGAL_VALUE reason code	
16	**EVENT_MENU_SELECTION not allowed** Call registerFileEvent() method with event=EVENT_MENU_SELECTION	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	

Id	Description	API Expectation	APDU Expectation
17	**EVENT_MENU_SELECTION_HELP_REQUEST not allowed** Call registerFileEvent() method with event= EVENT_MENU_SELECTION_HELP_REQUEST	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
18	**EVENT_TIMER_EXPIRATION not allowed** Call registerFileEvent() method with event=EVENT_TIMER_EXPIRATION	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
19	**EVENT_STATUS_COMMAND not allowed** Call registerFileEvent() method with event=EVENT_STATUS_COMMAND	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
20	**EVENT_NOT_SUPPORTED Exception** Call registerFileEvent() method with event=EVENT_PROFILE_DOWNLOAD	Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason code	
21	**Register a deleted and recreated EF under MF** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F0011116F03" baADFAid=null 2- Update binary on MF\DFTEST\EFTARU 3- Delete MF\DFTEST\EFTARU 4- Create MF\DFTEST\EFTARU 5- Update binary on MF\DFTEST\EFTARU 6- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F0011116F03" baADFAid=null	1- No exception is thrown 2- Applet is triggered 5- Applet is triggered	
22	**Register a deleted and recreated DF under MF** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F0011112211" baADFAid=null 2- Update binary on MF\DFTEST\DFSUB_TEST\EFTAA 3- Delete DFSUB_TEST 4- Create DFSUB_TEST, create EFTAA 5- Update binary on MF\DFTEST\DFSUB_TEST\EFTAA 6- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F0011112211" baADFAid=null	1- No exception is thrown 2- Applet is triggered 5- Applet is triggered	

Id	Description	API Expectation	APDU Expectation
23	**Register a non existing EF under MF** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F00111122112223" baADFAid=null 2- Create MF\DFTEST\DFSUB_TEST\EFTNEW (2223) 3- Update binary on MF\DFTEST\DFSUB_TEST\EFTNEW 4- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F00111122112223" baADFAid=null 5- Delete MF\DFTEST\DFSUB_TEST\EFTNEW	1- No exception is thrown 3- Applet is triggered	
24	**Register a non existing DF under MF** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F0011112212" baADFAid=null 2- Create MF\DFTEST\DFNEW (2212) 3- Create MF\DFTEST\DFNEW\EFTNEW 4- Update binary on MF\DFTEST\DFNEW\EFTNEW 5- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F0011112212" baADFAid=null 6- Delete MF\DFTEST\DFNEW 7-Restore EFs	1- No exception is thrown 4- Applet is triggered 5- Applet is triggered	

NOTE: Complementary information about tests 21, 22, 23, 24 can be found in document SCPt040568 in ETSI web site.

5.2.8.16 Method registerFileEvent(short event, FileView aFileView)

Test Area Reference: Api_2_Tkr_RgfeSo.

5.2.8.16.1 Conformance requirement

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

```
public void registerFileEvent(short event,
                              FileView aFileView)
                       throws ToolkitException,
                              java.lang.NullPointerException,
                              javacard.framework.TransactionException
```

5.2.8.16.1.1 Normal execution

- CRRN1: The only event allowed and supported by the method is EVENT_EXTERNAL_FILE_UPDATE.
- CRRN2: The CAT Runtime Environment shall trigger the applet when the aFileView object's current file is updated.

- CRRN3: If the aFileView object's current file is a dedicated file, the Applet shall be triggered when an elementary file within this dedicated file is updated.
- CRRN4: A later change in the FileView shall not modify the registration.
- CRRN5: A call to isEventSet() method for EVENT_EXTERNAL_FILE_UPDATE should return true if the registerFileEvent() method has been successfully called.

5.2.8.16.1.2 Parameter errors

- CRRP1: Shall throw a java.lang.NullPointerException if aFileView is null.
- CRRP2: Shall throw a javacard.framework.TransactionException if the operation would cause the commit capacity to be exceeded.
- CRRP3: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_MENU_SELECTION.
- CRRP4: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_MENU_SELECTION_HELP_REQUEST.
- CRRP5: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_TIMER_EXPIRATION.
- CRRP6: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_STATUS_COMMAND.
- CRRP7: Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason if the event is not EVENT_EXTERNAL_FILE_UPDATE.

5.2.8.16.1.3 Context errors

No requirements.

5.2.8.16.2 Test area files

Test Source: Test_Api_2_Tkr_RgfeSo.java.

Test Applet: Api_2_Tkr_RgfeSo _1.java.

Cap File: api_2_tkr_rgfeso.cap.

5.2.8.16.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4
N2	1, 3, 11, 12
N3	2, 4
N4	1, 3
N5	1
P1	5
P2	Not testable
P3	6
P4	7
P5	8
P6	9
P7	10

5.2.8.16.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Register EF under MF** 1- Call isEventSet (EVENT_EXTERNAL_FILE_UPDATE) method 2- Call UICCView=getTheUICCView() 3- Applet selects MF\DFTEST\EFTARU. 4- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = UICCView 5- Update binary on MF\DFTEST\EFTARU 6- Call isEventSet (EVENT_EXTERNAL_FILE_UPDATE) method 7- Applet selects EFCARU. 8- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = UICCView 9- Update binary on MF\DFTEST\EFTARU 10- Increase on MF\DFTEST\EFCARU 11- Update record on MF\DFTEST\EFLARU 12- Update binary on MF\DFTEST\DFSUB_TEST\EFTAA 13- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = UICCView 14- Applet selects MF\DFTEST\EFTARU. 15- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = UICCView	1- Returns false 5- Applet is triggered 6- Returns true 8- No exception is thrown 9- Applet is triggered 10- Applet is triggered 11- Applet is not triggered 12- Applet is not triggered	

Id	Description	API Expectation	APDU Expectation
2	**Register DF under MF** 1- Applet selects MF\DFTEST. 2- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = UICCView 3- Update binary on MF\DFTEST\EFTARU 4- Increase on MF\DFTEST\EFCARU 5- Update record on MF\DFTEST\EFLARU 6- Update binary on MF\DFTEST\ DFSUB_TEST\EFTAA 7- Applet selects MF\DFTEST 8- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = UICCView	2- No exception is thrown 3- Applet is triggered 4- Applet is triggered 5- Applet is triggered 6- Applet is not triggered	
3	**Register EF under ADF1** 1- Call ADF1View=getTheFileView() 2- Applet selects ADF1\DFTEST\EFTARU. 3- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View 4- Update binary on ADF1\DFTEST\EFTARU 5- Applet selects ADF1\DFTEST\EFCARU. 6- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View 7- Update binary on ADF1\DFTEST\EFTARU 8- Increase on ADF1\DFTEST\EFCARU 9- Update record on ADF1\DFTEST\EFLARU 10- Update binary on ADF1\DFTEST\ DFSUB_TEST\EFTAA 11- Applet selects ADF1\DFTEST\EFCARU. 12- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View 13- Applet selects ADF1\DFTEST\EFTARU. 14- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View	3- No exception is thrown 4- Applet is triggered 6- No exception is thrown 7- Applet is triggered 8- Applet is triggered 9- Applet is not triggered 10- Applet is not triggered	

Id	Description	API Expectation	APDU Expectation
4	**Register DF under ADF1** 1- Applet selects DFTEST. 2- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View 3- Update binary on ADF1\DFTEST\EFTARU 4- Increase on ADF1\DFTEST\EFCARU 5- Update record on ADF1\DFTEST\EFLARU 6- Update binary on ADF1\DFTEST\ DFSUB_TEST\EFTAA 7- Applet selects ADF1\DFTEST 8- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View	2- No exception is thrown 3- Applet is triggered 4- Applet is triggered 5- Applet is triggered 6- Applet is not triggered	
5	**NullPointerException Exception** Call registerFileEvent() method with null aFileView	Shall throw a NullPointerException	
6	**EVENT_MENU_SELECTION not allowed** Call registerFileEvent() method with event=EVENT_MENU_SELECTION	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
7	**EVENT_MENU_SELECTION_HELP_REQUEST not allowed** Call registerFileEvent() method with event= EVENT_MENU_SELECTION_HELP_REQUEST	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
8	**EVENT_TIMER_EXPIRATION not allowed** Call registerFileEvent() method with event=EVENT_TIMER_EXPIRATION	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
9	**EVENT_STATUS_COMMAND not allowed** Call registerFileEvent() method with event=EVENT_STATUS_COMMAND	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
10	**EVENT_NOT_SUPPORTED Exception** Call registerFileEvent() method with event=EVENT_PROFILE_DOWNLOAD	Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason code	
11	**Register a deleted and recreated EF under ADF** 1- Applet selects ADF1\$DF_{TEST}$\$EF_{TARU}$ 2- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View 3- Update binary on ADF1\$DF_{TEST}$\$EF_{TARU}$ 4- Delete ADF1\$DF_{TEST}$\$EF_{TARU}$ 5- Create ADF1\$DF_{TEST}$\$EF_{TARU}$ 6- Update binary on ADF1\$DF_{TEST}$\$EF_{TARU}$ 7- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View	2- No exception is thrown 3- Applet is triggered 6- Applet is triggered	

Id	Description	API Expectation	APDU Expectation
12	**Register a deleted and recreated DF under ADF** 1- Applet selects ADF1\DF$_{TEST}$\DF$_{SUB_TEST}$ 2- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View 3- Update binary on ADF1\DF$_{TEST}$\ DF$_{SUB_TEST}$\EF$_{TAA}$ 4- Delete EF$_{TAA}$, delete DF$_{SUB_TEST}$ 5- Create DF$_{SUB_TEST}$, create EF$_{TAA}$ 6- Update binary on ADF1\DF$_{TEST}$\ DF$_{SUB_TEST}$\EF$_{TAA}$ 7- Applet selects ADF1\DF$_{TEST}$\ DF$_{SUB_TEST}$ 8- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View 9- Restore EFs	2- No exception is thrown 3- Applet is triggered 6- Applet is triggered	

NOTE: Complementary information about tests 11, 12 can be found in document SCPt040568 in ETSI web site.

5.2.8.17 Method deregisterFileEvent(short event, byte[] baFileList, short sOffset1, short sLength1, byte[] baADFAid, short sOffset2, byte bLength2)

Test Area Reference: Api_2_Tkr_Drfes_Bss_Bsb.

5.2.8.17.1 Conformance requirement

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

```
public void deregisterFileEvent(short event,
                                byte[] baFileList,
                                short sOffset1,
                                short sLength1,
                                byte[] baADFAid,
                                short sOffset2,
                                byte bLength2)
                         throws ToolkitException,
                                java.lang.NullPointerException,
                                java.lang.ArrayIndexOutOfBoundsException,
                                javacard.framework.TransactionException,
                                javacard.framework.SystemException
```

5.2.8.17.1.1 Normal execution

- CRRN1: The only event allowed and supported by the method is EVENT_EXTERNAL_FILE_UPDATE.
- CRRN2: The applet is deregistered to the file included in the baFileList.
- CRRN3: If a file in baFileList is a dedicated file the deregistration shall not affect the monitoring of an elementary file within the dedicated file that was individually registered.
- CRRN4: If a file in baFileList is an elementary file the deregistration will not affect the monitoring of the parent dedicated file that was individually registered.
- CRRN5: The baADFAid indicates the Aid of the ADF under which the file is located.
- CRRN6: If baADFAid is null, it indicates that the file is located under the MF and not located under an ADF.

- CRRN7: A call to isEventSet() method for EVENT_EXTERNAL_FILE_UPDATE should return false if the applet has been deregistered completely to all its registered EFs and DFs.

5.2.8.17.1.2 Parameter errors

- CRRP1: Shall throw a java.lang.NullPointerException if baFileList is null.
- CRRP2: Shall throw a java.lang.ArrayIndexOutOfBoundsException if sOffset1 or sLength1 or both would cause access outside array bounds.
- CRRP3: Shall throw a java.lang.ArrayIndexOutOfBoundsException if sOffset2 or sLength2 or both would cause access outside array bounds.
- CRRP4: Shall throw a javacard.framework.TransactionException if the operation would cause the commit capacity to be exceeded.
- CRRP5: Shall throw a javacard.framework.SystemException with ILLEGAL_VALUE reason if bLength2 is not in the range of 5 - 16 bytes.
- CRRP6: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_MENU_SELECTION.
- CRRP7: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_MENU_SELECTION_HELP_REQUEST.
- CRRP8: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_TIMER_EXPIRATION.
- CRRP9: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_STATUS_COMMAND.
- CRRP10: Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason if the event is not EVENT_EXTERNAL_FILE_UPDATE.

5.2.8.17.1.3 Context errors

No requirements.

5.2.8.17.2 Test area files

Test Source: Test_Api_2_Tkr_Drfes_Bss_Bsb.java.

Test Applet: Api_2_Tkr_Drfes_Bss_Bsb _1.java.

Cap File: api_2_tkr_drfes_bss_bsb.cap.

5.2.8.17.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5, 6
N2	1, 4
N3	2, 5
N4	3, 6
N5	4, 5, 6
N6	1, 2, 3
N7	1
P1	7
P2	8, 9, 10, 11, 12
P3	13, 14, 15, 16
P4	Not testable
P5	17
P6	18
P7	19
P8	20
P9	21
P10	22

5.2.8.17.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Deregister EF under MF** 1- Call isEventSet (EVENT_EXTERNAL_FILE_UPDATE) method 2- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="03 3F0011116F03 3F0011116F09 3F0011116F0C" baADFAid=null 3- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$ 4- Increase on MF\DF$_{TEST}$\EF$_{CARU}$ 5- Update record on MF\DF$_{TEST}$\EF$_{LARU}$ 6- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="02 3F0011116F03 3F0011116F09" baADFAid=null 7- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$ 8- Increase on MF\DF$_{TEST}$\EF$_{CARU}$ 9- Update record on MF\DF$_{TEST}$\EF$_{LARU}$ 10- Call isEventSet (EVENT_EXTERNAL_FILE_UPDATE) method 11- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F0011116F0C" baADFAid=null 12- Update record on MF\DF$_{TEST}$\EF$_{LARU}$ 13- Call isEventSet (EVENT_EXTERNAL_FILE_UPDATE) method	1- Returns false 3- Applet is triggered 4- Applet is triggered 5- Applet is triggered 6- No exception is thrown 7- Applet is not triggered 8- Applet is not triggered 9- Applet is triggered 10- Returns true 11- No exception is thrown 12- Applet is not triggered 13- Returns false	

Id	Description	API Expectation	APDU Expectation
2	**Deregister DF does not affect child EF** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="02 3F0011116F03 3F001111" baADFAid=null 2- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$ 3- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F001111" baADFAid=null 4- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$	2- Applet is triggered 3- No exception is thrown 4- Applet is triggered	
3	**Deregister EF does not affect parent DF** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F001111" baADFAid=null 2- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$ 3- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F0011116F03" baADFAid=null 4- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$ 5- Update record on MF\DF$_{TEST}$\EF$_{LARU}$	2- Applet is triggered 3- No exception is thrown 4- Applet is triggered 5- Applet is triggered	

Id	Description	API Expectation	APDU Expectation
4	**Deregister EF under ADF1** 1- Call isEventSet (EVENT_EXTERNAL_FILE_UPDATE) method 2- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="03 3F007FFF11116F03 3F007FFF11116F09 3F007FFF11116F0C" baADFAid="AID ADF1" 3- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$ 4- Increase on MF\DF$_{TEST}$\EF$_{CARU}$ 5- Update record on MF\DF$_{TEST}$\EF$_{LARU}$ 6- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="02 3F007FFF11116F03 3F007FFF11116F09" baADFAid="AID ADF1" 7- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$ 8- Increase on MF\DF$_{TEST}$\EF$_{CARU}$ 9- Update record on MF\DF$_{TEST}$\EF$_{LARU}$ 10- Call isEventSet (EVENT_EXTERNAL_FILE_UPDATE) method 11- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F007FFF11116F0C" baADFAid="AID ADF1" 12- Update record on MF\DF$_{TEST}$\EF$_{LARU}$ 13- Call isEventSet (EVENT_EXTERNAL_FILE_UPDATE) method	1- Returns false 3- Applet is triggered 4- Applet is triggered 5- Applet is triggered 6- No exception is thrown 7- Applet is not triggered 8- Applet is not triggered 9- Applet is triggered 10- Returns true 11- No exception is thrown 12- Applet is not triggered 13- Returns false	
5	**Deregister DF does not affect child EF (under ADF1)** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="02 3F007FFF11116F03 3F001111" baADFAid="AID ADF1" 2- Update binary on MF\DFTEST\EFTARU 3- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F007FFF1111" baADFAid="AID ADF1" 4- Update binary on MF\DFTEST\EFTARU	2- Applet is triggered 3- No exception is thrown 4- Applet is triggered	

Id	Description	API Expectation	APDU Expectation
6	**Deregister EF does not affect parent DF (under ADF1)** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F007FFF1111" baADFAid="AID ADF1" 2- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$ 3- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="01 3F007FFF11116F03" baADFAid="AID ADF1" 4- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$ 5- Update record on MF\DF$_{TEST}$\EF$_{LARU}$ 6- Restore EFs	1- No exception is thrown 2- Applet is triggered 3- No exception is thrown 4- Applet is triggered 5- Applet is triggered	
7	**NullPointerException Exception** Call deregisterFileEvent() method with baFileList null	Shall throw a NullPointerException	
8	**sOffset1 >= baFileList.length** Call deregisterFileEvent() method with baFileList.length = 7 sOffset1 = 8 sLength1 = 4	Shall throw a ArrayIndexOutOfBoundsException	
9	**sOffset1 < 0** Call deregisterFileEvent() method with baFileList.length = 19 sOffset1 = -1 sLength1 = 4	Shall throw a ArrayIndexOutOfBoundsException	
10	**sLength1 > baFileList.length** Call deregisterFileEvent() method with baFileList.length = 7 sOffset1 = 0 sLength1 = 10	Shall throw a ArrayIndexOutOfBoundsException	
11	**sOffset1 + sLength1 > baFileList.length** Call deregisterFileEvent() method with baFileList.length = 7 sOffset1 = 5 sLength1 = 4	Shall throw a ArrayIndexOutOfBoundsException	
12	**sLength1 < 0** Call deregisterFileEvent() method with baFileList.length = 7 sOffset1 = 0 sLength1 = -1	Shall throw a ArrayIndexOutOfBoundsException	
13	**sOffset2 >= baFileList.length** Call deregisterFileEvent() method with baADFAid.length = 15 sOffset2 = 15 bLength2 = 6	Shall throw a ArrayIndexOutOfBoundsException	
14	**sOffset2 < 0** Call deregisterFileEvent() method with baADFAid.length = 15 sOffset2 = -1 bLength2 = 6	Shall throw a ArrayIndexOutOfBoundsException	
15	**sLength2 > baFileList.length** Call deregisterFileEvent() method with baADFAid.length = 15 sOffset2 = 0 bLength2 = 16	Shall throw a ArrayIndexOutOfBoundsException	

Id	Description	API Expectation	APDU Expectation
16	**sOffset2+ sLength2 > baFileList.length** Call deregisterFileEvent() method with baADFAid.length = 15 sOffset1 = 10 bLength1 = 6	Shall throw a ArrayIndexOutOfBoundsException	
17	**ILLEGAL_VALUE Exception** 1- Call deregisterFileEvent() method with baADFAid.length = 18 sOffset2 = 0 bLength2 = 4 2- Call deregisterFileEvent() method with baADFAid.length = 18 sOffset2 = 0 bLength2 = 18	1- Shall throw a SystemException with ILLEGAL_VALUE reason code 2- Shall throw a SystemException with ILLEGAL_VALUE reason code	
18	**EVENT_MENU_SELECTION not allowed** Call deregisterFileEvent() method with event=EVENT_MENU_SELECTION	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
19	**EVENT_MENU_SELECTION_HELP_REQUEST not allowed** Call deregisterFileEvent() method with event= EVENT_MENU_SELECTION_HELP_REQUEST	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
20	**EVENT_TIMER_EXPIRATION not allowed** Call deregisterFileEvent() method with event=EVENT_TIMER_EXPIRATION	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
21	**EVENT_STATUS_COMMAND not allowed** Call deregisterFileEvent() method with event=EVENT_STATUS_COMMAND	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
22	**EVENT_NOT_SUPPORTED Exception** Call deregisterFileEvent() method with event=EVENT_PROFILE_DOWNLOAD	Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason code	

5.2.8.18 Method deregisterFileEvent(short event, FileView aFileView)

Test Area Reference: Api_2_Tkr_Drfeso.

5.2.8.18.1 Conformance requirement

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

```
public void deregisterFileEvent(short event,
                                FileView aFileView)
                         throws ToolkitException,
                                java.lang.NullPointerException,
                                javacard.framework.TransactionException
```

5.2.8.18.1.1 Normal execution

- CRRN1: The only event allowed and supported by the method is EVENT_EXTERNAL_FILE_UPDATE.
- CRRN2: The aFileView object's current file indicates the file that is no longer monitored. The applet is deregistered to the aFileView object's current file.
- CRRN3: If the current file is a dedicated file the deregistration shall not affect the monitoring of an elementary file within the dedicated file that was individually registered.
- CRRN4: If the current file is an elementary file the deregistration will not affect the monitoring of the parent dedicated file that was individually registered.
- CRRN5: A call to isEventSet() method for EVENT_EXTERNAL_FILE_UPDATE should return false if the applet has been deregistered completely to all its registered EFs and DFs.

5.2.8.18.1.2 Parameter errors

- CRRP1: Shall throw a java.lang.NullPointerException if aFileView is null.
- CRRP2: Shall throw a javacard.framework.TransactionException if the operation would cause the commit capacity to be exceeded.
- CRRP3: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_MENU_SELECTION.
- CRRP4: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_MENU_SELECTION_HELP_REQUEST.
- CRRP5: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_TIMER_EXPIRATION.
- CRRP6: Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if the event is EVENT_STATUS_COMMAND.
- CRRP7: Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason if the event is not EVENT_EXTERNAL_FILE_UPDATE.

5.2.8.18.1.3 Context errors

No requirements.

5.2.8.18.2 Test area files

Test Source: Test_Api_2_Tkr_Drfeso.java.

Test Applet: Api_2_Tkr_Drfeso_1.java.

Cap File: api_2_tkr_drfeso.cap.

5.2.8.18.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5, 6
N2	1, 4
N3	2, 5
N4	3, 6
N5	1
P1	7
P2	Not testable
P3	8
P4	9
P5	10
P6	11
P7	12

5.2.8.18.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Deregister EF under MF** 1- Call isEventSet (EVENT_EXTERNAL_FILE_UPDATE) method 2- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="03 3F0011116F03 3F0011116F09 3F0011116F0C" baADFAid=null 3- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$ 4- Increase on MF\DF$_{TEST}$\EF$_{CARU}$ 5- Update record on MF\DF$_{TEST}$\EF$_{LARU}$ 6- Call UICCView=getTheUICCView() 7- Applet selects EF$_{TARU}$. 8- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = UICCView 9- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$ 10- Applet selects EF$_{CARU}$. 11- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = UICCView 12- Increase on MF\DF$_{TEST}$\EF$_{CARU}$ 13- Call isEventSet (EVENT_EXTERNAL_FILE_UPDATE) method 14- Applet selects EF$_{LARU}$. 15- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = UICCView 16- Update record on MF\DF$_{TEST}$\EF$_{LARU}$ 17- Call isEventSet (EVENT_EXTERNAL_FILE_UPDATE) method	1- Returns false 3- Applet is triggered 4- Applet is triggered 5- Applet is triggered 8- No exception is thrown 9- Applet is not triggered 11- No exception is thrown 12- Applet is not triggered 13- Returns true 15- No exception is thrown 16- Applet is not triggered 17- Returns false	
2	**Deregister DF does not affect child EF** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="02 3F0011116F03 3F001111" baADFAid=null 2- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$ 3- Applet selects DF$_{TEST}$. 4- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = UICCView 5- Update binary on MF\DF$_{TEST}$\EF$_{TARU}$	2- Applet is triggered 4- No exception is thrown 5- Applet is triggered	

Id	Description	API Expectation	APDU Expectation
3	**Deregister EF does not affect parent DF** 1-Select DF_{TEST} 2- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = UICCView 3- Update binary on MF\$DF_{TEST}$\$EF_{TARU}$ 4- Applet selects EF_{TARU}. 5- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = UICCView 6- Update binary on MF\$DF_{TEST}$\$EF_{TARU}$ 7- Update record on MF\$DF_{TEST}$\$EF_{LARU}$	3- Applet is triggered 5- No exception is thrown 6- Applet is triggered 7- Applet is triggered	
4	**Deregister EF under ADF1** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="03 3F007FFF11116F03 3F007FFF11116F09 3F007FFF11116F0C" baADFAid="AID ADF1" 2- Update binary on ADF1\$DF_{TEST}$\$EF_{TARU}$ 3- Increase on ADF1\$DF_{TEST}$\$EF_{CARU}$ 4- Update record on ADF1\$DF_{TEST}$\$EF_{LARU}$ 5- Call ADF1View=getTheFileView() 6- Applet selects EF_{TARU}. 7- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View 8- Update binary on ADF1\$DF_{TEST}$\$EF_{TARU}$ 9- Applet selects EF_{CARU}. 10- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View 11- Increase on ADF1\$DF_{TEST}$\$EF_{CARU}$ 12- Applet selects EF_{LARU}. 13- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View 14- Update record on ADF1\$DF_{TEST}$\$EF_{LARU}$	2- Applet is triggered 3- Applet is triggered 4- Applet is triggered 7- No exception is thrown 8- Applet is not triggered 10- No exception is thrown 11- Applet is not triggered 13- No exception is thrown 14- Applet is not triggered	

Id	Description	API Expectation	APDU Expectation
5	**Deregister DF does not affect child EF (under ADF1)** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE baFileList="02 3F007FFF11116F03 3F007FFF1111" baADFAid="AID ADF1" 2- Update binary on ADF1\$DF_{TEST}$\$EF_{TARU}$ 3- Applet selects DF_{TEST}. 4- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View 5- Update binary on ADF1\$DF_{TEST}$\$EF_{TARU}$	 2- Applet is triggered 4- No exception is thrown 5- Applet is triggered	
6	**Deregister EF does not affect parent DF (under ADF1)** 1- Call registerFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View 2- Update binary on ADF1\$DF_{TEST}$\$EF_{TARU}$ 3- Applet selects EF_{TARU}. 4- Call deregisterFileEvent() method with parameters: event= EVENT_EXTERNAL_FILE_UPDATE aFileView = ADF1View 5- Update binary on ADF1\$DF_{TEST}$\$EF_{TARU}$ 6- Update record on ADF1\$DF_{TEST}$\$EF_{LARU}$ 7- Restore EFs	 1- No exception is thrown 2- Applet is triggered 4- No exception is thrown 5- Applet is triggered 6- Applet is triggered	
7	**NullPointerException Exception** Call registerFileEvent() method with null aFileView	Shall throw a NullPointerException	
8	**EVENT_MENU_SELECTION not allowed** Call registerFileEvent() method with event=EVENT_MENU_SELECTION	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
9	**EVENT_MENU_SELECTION_HELP_REQUEST not allowed** Call registerFileEvent() method with event= EVENT_MENU_SELECTION_HELP_REQUEST	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
10	**EVENT_TIMER_EXPIRATION not allowed** Call registerFileEvent() method with event=EVENT_TIMER_EXPIRATION	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
11	**EVENT_STATUS_COMMAND not allowed** Call registerFileEvent() method with event=EVENT_STATUS_COMMAND	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code	
12	**EVENT_NOT_SUPPORTED Exception** Call registerFileEvent() method with event=EVENT_PROFILE_DOWNLOAD	Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason code	

5.2.8.19 Method setMenuEntryTextAttribute

Test Area Reference: Api_2_Tkr_Smta.

5.2.8.19.1 Conformance requirement

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

```
public void setMenuEntryTextAttribute(byte id,
                                       byte[] textAttribute,
                                       short offset,
                                       short length)
                                throws java.lang.NullPointerException,
                                       java.lang.ArrayIndexOutOfBoundsException,
                                       ToolkitException,
                                       javacard.framework.TransactionException
```

5.2.8.19.1.1 Normal execution

- CRRN1: Sets the text attribute of a menu entry.
- CRRN2: The text attribute provided will be added to the text attribute list of the item text attribute list Comprehension TLV.
- CRRN3: After the invocation of this method, during the current card session, the CAT Runtime Environment shall dynamically update the menu stored in the terminal.

5.2.8.19.1.2 Parameter errors

- CRRP1: Shall throw a java.lang.NullPointerException if textAttribute is null.
- CRRP2: Shall throw a java.lang.ArrayIndexOutOfBoundsException if offset or length or both would cause access outside array bounds.
- CRRP3: Shall throw a ToolkitException with MENU_ENTRY_NOT_FOUND reason if the menu entry does not exist for this applet.
- CRRP4: Shall throw a ToolkitException with BAD_INPUT_PARAMETER reason if the length is different from 4.
- CRRP5: Shall throw a javacard.framework.TransactionException if the operation would cause the commit capacity to be exceeded.

5.2.8.19.1.3 Context errors

No requirements.

5.2.8.19.2 Test area files

Test Source: Test_ Api_2_Tkr_Smta.java.

Test Applet: Api_2_Tkr_Smta _1.java.

Cap File: api_2_tkr_smta.cap.

5.2.8.19.3 Test coverage

CRR number	Test case number
N1	1, 2
N2	1, 2
N3	1, 2
P1	3
P2	4, 5, 6, 7
P3	8
P4	9
P5	Not testable

5.2.8.19.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Text attribute update 1** Call setMenuEntryTextAttribute() with parameters: Id = '02' textAttribute= "00 0C 11 02 00 0C 10 03" Offset = 0 Length = 4	No exception shall be thrown.	The UICC shall issue a SETUP MENU proactive command which contains the text Attribute list value "00 00 03 90" “00 0C 11 02” "00 00 03 90" "00 00 03 90"
2	**Text attribute update 2** Call setMenuEntryTextAttribute() with parameters: Id = '04' textAttribute= "00 0C 11 02 00 0C 10 03" Offset = 4 Length = 4	No exception shall be thrown.	The UICC shall issue a SETUP MENU proactive command which contains the text Attribute list value "00 00 03 90" “00 0C 11 02” "00 00 03 90" "00 0C 10 03"
3	**Call setMenuEntryTextAttribute() with null textAttribute**	Shall throw a NullPointerException	
4	**DstOffset >= dstBuffer.length** setMenuEntryTextAttribute() dstBuffer.length = 8 dstOffset = 8 dstLength = 4	Shall throw a ArrayIndexOutOfBoundsException	
5	**dstOffset < 0** setMenuEntryTextAttribute() dstBuffer.length = 8 dstOffset = -1 dstLength = 4	Shall throw a ArrayIndexOutOfBoundsException	
6	**DstLength > dstBuffer.length** setMenuEntryTextAttribute() dstBuffer.length = 3 dstOffset = 0 dstLength = 4	Shall throw a ArrayIndexOutOfBoundsException	
7	**dstOffset + dstLength > dstBuffer.length** setMenuEntryTextAttribute() dstBuffer.length = 8 dstOffset = 5 dstLength = 4	Shall throw a ArrayIndexOutOfBoundsException	
8	**MENU_ENTRY_NOT_FOUND exception** Call setMenuEntryTextAttribute() with Id = 08	Shall throw a ToolkitException with MENU_ENTRY_NOT_FOUND reason code	

Id	Description	API Expectation	APDU Expectation
9	**BAD_INPUT_PARAMETER exception** `Call setMenuEntryTextAttribute() with length = 2`	Shall throw a ToolkitException with BAD_INPUT_PARAMETER reason code	

5.2.9 Interface ViewHandler

Tests are done in inheriting interfaces EnvelopeHandler, EnvelopeResponseHandler, ProactiveHandler and ProactiveResponseHandler.

5.2.10 Interface BERTLVEditHandler

5.2.10.1 Method setTag

Test Area Reference: Api_2_Bte_Sttg.

5.2.10.1.1 Conformance requirement

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

```
public void setTag(byte bBERTag)
```

5.2.10.1.1.1 Normal execution

- CRRN1: Sets the tag of the BER TLV list.

5.2.10.1.1.2 Parameter errors

No requirements.

5.2.10.1.1.3 Context errors

No requirements.

5.2.10.1.2 Test area files

Test Source: Test_Api_2_Bte_Sttg.java.

Test Applet: Api_2_Bte_Sttg_1.java.

Cap File: api_2_bte_sttg.cap.

5.2.10.1.3 Test coverage

CRR number	Test case number
1	1

5.2.10.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	`Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x10`		
1	`1- setTag(0x01)` `2- getTag()`	2- Returns 0x01	

5.2.10.2 Method getTag

Test Area Reference: Api_2_Bte_Gttg.

5.2.10.2.1 Conformance requirement

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

```
public byte getTag()
```

5.2.10.2.1.1 Normal execution

- CRRN1: Returns the BER Tag of the BER TLV list.

5.2.10.2.1.2 Parameter errors

No requirements.

5.2.10.2.1.3 Context errors

No requirements.

5.2.10.2.2 Test area files

Test Source: Test_Api_2_Bte_Gttg.java.

Test Applet: Api_2_Bte_Gttg_1.java.

Cap File: api_2_bte_gttg.cap.

5.2.10.2.3 Test coverage

CRR number	Test case number
1	Tested in API_2_BTE_STTG

5.2.10.3 Method getSize

Test Area Reference: Api_2_Bte_Gtsz.

5.2.10.3.1 Conformance requirement

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

```
public short getSize()
```

5.2.10.3.1.1 Normal execution

- CRRN1: Returns the BER TLV size, this includes the tag and the length.

5.2.10.3.1.2 Parameter errors

No requirements.

5.2.10.3.1.3 Context errors

No requirements.

5.2.10.3.2 Test area files

Test Source: Test_Api_2_Bte_Gtsz.java.

Test Applet: Api_2_Bte_Gtsz_1.java.

Cap File: api_2_bte_gtsz.cap.

5.2.10.3.3 Test coverage

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

5.2.10.3.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	`Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x110`		
1	`Fill the handler with BERTLV tag 0x01, data length 0x22` `Call getSize() method`	Returns 0x24	
2	`Fill the handler with BERTLV tag 0x01, data length 0x7F` `Call getSize() method`	Returns 0x81	
3	`Fill the handler with BERTLV tag 0x01, data length 0x80` `Call getSize() method`	Returns 0x83	
4	`Fill the handler with BERTLV tag 0x01, data length 0xFF` `Call getSize() method`	Returns 0x102	
5	`Fill the handler with BERTLV tag 0x01, data length 0x100` `Call getSize() method`	Returns 0x104	
6	`Fill the handler with BERTLV tag 0x01, data length 0x110` `Call getSize() method`	Returns 0x114	

5.2.10.4 Method getLength

Test Area Reference Api_2_Bte_Glen.

5.2.10.4.1 Conformance requirement

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

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

5.2.10.4.1.1 Normal execution

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

5.2.10.4.1.2 Parameter errors

No requirements.

5.2.10.4.1.3 Context errors

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

5.2.10.4.2 Test area files

Test Source: Test_Api_2_Bte_Glen.java.

Test Applet: Api_2_Bte_Glen_1.java.

Cap File: api_2_bte_glen.cap.

5.2.10.4.3 Test coverage

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

5.2.10.4.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	`Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01`		
1	**Clear the handler** `getLength()`	Result of getLength() is 0	
2	**Call the appendArray() method with buffer length 9** `getLength()`	Result of getLength() is 9	
3	**Call the appendArray() method, with buffer length = 253** `getLength()`	Result of getLength() is 253	
4	**Build a 7Fh Handler and fill it with appendArray() method** `getLength()`	Result of getLength() is 7Fh	
5	**Build a 80h Handler and fill it with appendArray() method** `getLength()`	Result of getLength() is 80h	
6	**Build a 100h Handler and fill it with appendArray() method** `getLength()`	Result of getLength() is 100h	

5.2.10.5 Method copy

Test Area Reference Api_2_Bte_Copy.

5.2.10.5.1 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
```

5.2.10.5.1.1 Normal execution

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

5.2.10.5.1.2 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 grater than the length of the Comprehension TLV List, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException.OUT_OF_TLV_BOUNDARIES.

5.2.10.5.1.3 Context errors

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

5.2.10.5.2 Test area files

Test Source: Test_Api_2_Bte_Copy.java.

Test Applet: Api_2_Bte_Copy _1.java.

Cap File: api_2_bte_copy.cap.

5.2.10.5.3 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 BERTLVEdit Handler

5.2.10.5.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	`Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01`		
1	**NULL as parameter to dstBuffer**	NullPointerException is thrown	
2	**Call the appendArray() method with 81 03 01 41 42 82 02 81 43**		

	DstOffset > dstBuffer.length
	`copy()`
	`dstBuffer.length = 5`
	`dstOffset = 6`
	`dstLength = 0`
		ArrayIndexOutOfBoundsException is thrown	

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

Id	Description	API Expectation	APDU Expectation
4	**DstLength > dstBuffer.length** copy() dstBuffer.length = 5 dstOffset = 0 dstLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**dstOffset + dstLength > dstBuffer.length** copy() dstBuffer.length = 5 dstOffset = 3 dstLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength < 0** copy() dstBuffer.length = 5 dstOffset = 0 dstLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**dstLength > length of the Comprehension TLV list** copy() dstBuffer.length = 10 dstOffset = 0 dstLength = 10	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**Successful call, dstBuffer is the whole buffer** copy() 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** copy() 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** copy() dstBuffer.length = 15 dstOffset = 3 dstLength = 6	Result of copy() is 9	
13	**Compare the whole buffer**	Result of arrayCompare() is 0	

5.2.10.6 Method findTLV

Test Area Reference Api_2_Bte_Find.

5.2.10.6.1 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
```

5.2.10.6.1.1 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.

5.2.10.6.1.2 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.

5.2.10.6.1.3 Context errors

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

5.2.10.6.2 Test area files

Test Source: Test_Api_2_Bte_Find.java.

Test Applet: Api_2_Bte_Find_1.java.

Cap File: api_2_bte_find.cap.

5.2.10.6.3 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 BERTLVEdit Handler

5.2.10.6.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Initialize the handler with 81 03 01 21 00 82 02 81 82**		
	Invalid input parameter findTLV() Occurrence = 0	ToolkitException.BAD_INPUT_PARAMETER is thrown	
2	**Call the init() method**		
	Search 1st TLV findTLV() Tag = 01h Occurrence = 1	Result is TLV_FOUND_CR_SET	
3	**Call the getValueLength() method**	Result is 03h	
4	**Search 2nd TLV** findTLV() 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 findTLV() Tag = 03h Occurrence = 1	Result is TLV_NOT_FOUND	
7	**Call the getValueLength() method**	ToolkitException.UNAVAILABLE_ELEMENT is thrown.	

Id	Description	API Expectation	APDU Expectation

8	**Search a tag with wrong occurrence**
	`findTLV()`
	`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
	`findTLV()`
	`Tag = 02h`
	`Occurrence = 2`
		Result is TLV_FOUND_CR_NOT_SET	

11	**Append a TLV with tag=04h**		

	Search the TLV
	`findTLV()`
	`Tag = 04h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_NOT_SET	

12	**Search tag 81h**
	`findTLV()`
	`Tag = 81h`
	`Occurrence = 1`
		Result is TLV_FOUND_CR_SET	

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

5.2.10.7 Method getValueLength

Test Area Reference Api_2_Bte_Gvle.

5.2.10.7.1 Conformance requirement

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

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

5.2.10.7.1.1 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.

5.2.10.7.1.2 Parameter errors

No requirements.

5.2.10.7.1.3 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.

5.2.10.7.2 Test area files

Test Source: Test_Api_2_Bte_Gvle.java.

Test Applet: Api_2_Bte_Gvle_1.java.

Cap File: api_2_bte_gvle.cap.

5.2.10.7.3 Test coverage

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

5.2.10.7.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Initialize the handler with 81 03 01 21 00 82 02 81 82**		
	getValueLength()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	**Call the appendTLV() method** tag = 0D valueOffset = 0 valueLength = 0		
	Search TLV 0Dh (Text String TLV)		
	getValueLength()	Result is 00h	
3	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 02 04 00** length = 1 (+ dcs byte)		
	Search TLV 0Dh (Text String TLV)		
	getValueLength()	Result is 02h	
4	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 7F 04 00 00 …**		
	Search TLV 0Dh (Text String TLV)		
	getValueLength()	Result is 7Fh	
5	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 81 80 04 00 00 …**		
	Search TLV 0Dh (Text String TLV)		
	getValueLength()	Result is 80h	
6	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 81 F1 04 00 00 …**		
	Search TLV 0Dh (Text String TLV)		
	getValueLength()	Result is F1h	

5.2.10.8 Method getValueByte

Test Area Reference Api_2_Bte_Gvby.

5.2.10.8.1 Conformance requirement

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

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

5.2.10.8.1.1 Normal execution

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

5.2.10.8.1.2 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.

5.2.10.8.1.3 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.

5.2.10.8.2 Test area files

Test Source: Test_Api_2_Bte_Gvby.java.

Test Applet: Api_2_Bte_Gvby_1.java.

Cap File: api_2_bte_gvby.cap.

5.2.10.8.3 Test coverage

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

5.2.10.8.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	`Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01`		

1	**Initialize the handler with 81 03 01 FF FE 82 02 81 FD**
	`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	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 7F 04 00 01 … 7D** `Search TLV 0Dh (Text String TLV)`		
	getValueByte(7E)	Result is 7Dh	
6	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 81 80 04 00 01 … 7E** `Search TLV 0Dh (Text String TLV)`		
	getValueByte(7E)	Result is 7Dh	

Id	Description	API Expectation	APDU Expectation
7	**getValueByte(7F)**	Result is 7Eh	

8	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 81 F1 04 00 01 … EF**
	
	`Search TLV 0Dh (Text String TLV)`
	`getValueByte(F0)`
		Result is EFh	

5.2.10.9 Method copyValue

Test Area Reference Api_2_Bte_Cpyv.

5.2.10.9.1 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
```

5.2.10.9.1.1 Normal execution

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

5.2.10.9.1.2 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 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.

5.2.10.9.1.3 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.

5.2.10.9.2 Test area files

Test Source: Test_Api_2_Bte_Cpyv.java.

Test Applet: Api_2_Bte_Cpyv_1.java.

Cap File: api_2_bte_cpyv.cap.

5.2.10.9.3 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 BERTLVEdit Handler
C2	11

5.2.10.9.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Initialize the handler with 81 03 01 21 00 82 02 81 82 Select a TLV**		
	copyValue() with a null dstBuffer	NullPointerException is thrown	
2	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 10 04 00 01 … 0E** Select Text String TLV		
	dstOffset > dstBuffer.length copyValue() dstBuffer.length = 5 dstOffset = 6 dstLength = 0	ArrayIndexOutOfBoundsException is thrown	
3	**dstOffset < 0** copyValue() dstBuffer.length = 5 dstOffset = -1 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**dstLength >dstBuffer.length** copyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**dstOffset + dstLength >dstBuffer.length** copyValue() dstBuffer.length = 5 dstOffset = 3 dstLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength < 0** copyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 06 04 00 01 … 04** Select Text String TLV		
	valueOffset > Text String Length copyValue() valueOffset = 7 dstBuffer.length = 15 dstOffset = 0 dstLength = 0	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**[Select Text String TLV]** valueOffset < 0 copyValue() valueOffset = -1 dstBuffer.length = 15 dstOffset = 0 dstLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	

Id	Description	API Expectation	APDU Expectation
9	**[Select Text String TLV]** dstLength > Text String length copyValue() 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 copyValue() valueOffset = 2 dstBuffer.length = 15 dstOffset = 0 dstLength = 5	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
11	**Initialize the handler with 81 03 01 21 00 82 02 81 82**		
	copyValue()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
12	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F** Select Text String TLV		
	Successful call copyValue() 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	**Initialize dstBuffer** dstBuffer = 55 55 … 55		
	Successful call copyValue() 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	

5.2.10.10 Method compareValue

Test Area Reference Api_2_Bte_Cprv.

5.2.10.10.1 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
```

5.2.10.10.1.1 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 Comprehension TLV List is less than that in compareBuffer.
- CRRN3: returns 1 if the first miscomparing byte in Comprehension TLV List is greater than that in compareBuffer.

5.2.10.10.1.2 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 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.

5.2.10.10.1.3 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.

5.2.10.10.2 Test area files

Test Source: Test_Api_2_Bte_Cprv.java.

Test Applet: Api_2_Bte_Cprv_1.java.

Cap File: api_2_bte_cprv.cap.

5.2.10.10.3 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 BERTLVEdit Handler
C2	11

5.2.10.10.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	`Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01`		
1	**Initialize the handler with 81 03 01 21 00 82 02 81 02 Select a TLV**		
	compareValue() with a null compareBuffer	NullPointerException is thrown	

Id	Description	API Expectation	APDU Expectation
2	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 10 04 00 01 … 0E** Select Text String TLV		
	compareOffset > compareBuffer.length compareValue() compareBuffer.length = 5 compareOffset = 6 compareLength = 0	ArrayIndexOutOfBoundsException is thrown	
3	**compareOffset < 0** compareValue() compareBuffer.length = 5 compareOffset = -1 compareLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**compareLength >compareBuffer.length** compareValue() compareBuffer.length = 5 compareOffset = 0 compareLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**compareOffset + compareLength >compareBuffer.length** compareValue() compareBuffer.length = 5 compareOffset = 3 compareLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**compareLength < 0** compareValue() compareBuffer.length = 5 compareOffset = 0 compareLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 06 04 00 01 … 04** **Select Text String TLV**		
	valueOffset > Text String Length compareValue() valueOffset = 7 compareBuffer.length = 15 compareOffset = 0 compareLength = 0	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**[Select Text String TLV]** valueOffset < 0 compareValue() 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 compareValue() 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 compareValue() valueOffset = 2 compareBuffer.length = 15 compareOffset = 0 compareLength = 5	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
11	**Initialize the handler with 81 03 01 21 00 82 02 81 82**		
	compareValue()	ToolkitException.UNAVAILABLE_ELEMENT is thrown	

Id	Description	API Expectation	APDU Expectation
12	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F** Select Text String TLV		
	Initialize compareBuffer compareBuffer = 04 00 01 … 0F		
	Compare buffers compareValue() valueOffset = 0 compareOffset = 0 compareLength = 17	Result is 00h	
13	**Initialize 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	**Initialize compareBuffer** compareBuffer = 03 00 01 … 0F		
	Compare buffers with same parameters	Result is +1	
15	**Initialize compareBuffer** compareBuffer = 55 55 55 01 02 03 04 05 06 07 08 09 0A 0B 0C 55 55 55 55 55		
	Compare buffers compareValue() valueOffset = 2 compareOffset = 3 compareLength = 12	Result is 00h	
16	**Initialize 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	**Initialize 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	**Initialize 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	

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

Test Area Reference Api_2_Bte_Facyb_Bs.

5.2.10.11.1 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
```

5.2.10.11.1.1 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.

5.2.10.11.1.2 Parameter errors

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

5.2.10.11.1.3 Context errors

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

5.2.10.11.2 Test area files

Test Source: Test_Api_2_Bte_Facyb_Bs.java.

Test Applet: Api_2_Bte_Facyb_Bs_1.java.

Cap File: api_2_bte_facyb_bs.cap.

5.2.10.11.3 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 BERTLVEdit Handler

5.2.10.11.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Initialize the handler with 81 03 01 21 00 82 02 81 02**		
	FindAndCopyValue() with a null dstBuffer	NullPointerException is thrown	
2	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 10 04 00 01 … 0E**		
	dstOffset > dstBuffer.length findAndCopyValue() tag = 0Dh dstBuffer.length = 20 dstOffset = 21	ArrayIndexOutOfBoundsException is thrown	
3	**dstOffset < 0** findAndCopyValue() dstBuffer.length = 20 dstOffset = -1	ArrayIndexOutOfBoundsException is thrown	
4	**length > dstBuffer.length** findAndCopyValue() dstBuffer.length = 15 dstOffset = 0	ArrayIndexOutOfBoundsException is thrown	
5	**DstOffset + length >dstBuffer.length** findAndCopyValue() DstBuffer.length = 20 DstOffset = 5	ArrayIndexOutOfBoundsException is thrown	
6	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F**		
	Select a TLV (tag 02h)		
	findAndCopyValue() tag = 03h	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the getValueLength() method	ToolkitException.UNAVAILABLE_ELEMENT is thrown.	
7	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F**		
	Successful call findAndCopyValue() Tag = 0Dh DstBuffer.length = 17 DstOffset = 0	Result of findAndcopyValue() is 17	
8	**Compare buffer** buffer = 04 00 01 … 0F	Result is 00h	
9	**Initialize dstBuffer** dstBuffer = 55 55 … 55		
	Successful call findAndCopyValue() 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	

Id	Description	API Expectation	APDU Expectation
11	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F**		
	append a 2nd Text String TLV		

	Successful call
	`findAndCopyValue()`
	`tag = 0Dh`
	`dstBuffer.length = 17`
	`dstOffset = 0`
		Result of findAndcopyValue() is 17	

12	**Compare buffer** `buffer = 04 00 01 … 0F`	Result is 00h	
13	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F**		

	Successful call (with tag 8Dh)
	`findAndCopyValue()`
	`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)
	`findAndCopyValue()`
	`tag = 8Fh`
	`dstBuffer.length = 16`
	`dstOffset = 0`
		Result of findAndcopyValue() is 16	

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

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

Test Area Reference Api_2_Bte_Facybbs_Bss.

5.2.10.12.1 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
```

5.2.10.12.1.1 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.

5.2.10.12.1.2 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 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.

5.2.10.12.1.3 Context errors

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

5.2.10.12.2 Test area files

Test Source: Test_Api_2_Bte_Facybbs_Bss.java.

Test Applet: Api_2_Bte_Facybbs_Bss_1.java.

Cap File: api_2_bte_facybbs_bss.cap.

5.2.10.12.3 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
P4	24
C1	Does not apply for BERTLVEdit Handler

5.2.10.12.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Initialize the handler with 81 03 01 21 00 82 02 81 82**		
	findAndCopyValue() with a null dstBuffer	NullPointerException is thrown	
2	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 10 04 00 01 … 0E**		
	dstOffset > dstBuffer.length findAndCopyValue() tag = 0Dh, occurrence = 1 valueOffset = 0 dstBuffer.length = 5 dstOffset = 6 dstLength = 0	ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
3	**dstOffset < 0** findAndCopyValue() dstBuffer.length = 5 dstOffset = -1 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**dstLength >dstBuffer.length** findAndCopyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**dstOffset + dstLength >dstBuffer.length** findAndCopyValue() dstBuffer.length = 5 dstOffset = 3 dstLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength < 0** findAndCopyValue() dstBuffer.length = 5 dstOffset = 0 dstLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 06 04 00 01 … 04**		
	valueOffset > Text String Length findAndCopyValue() tag = 0Dh, occurrence = 1 valueOffset = 7 dstBuffer.length = 15 dstOffset = 0 dstLength = 0	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**valueOffset < 0** findAndCopyValue() valueOffset = -1 dstBuffer.length = 15 dstOffset = 0 dstLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
9	**dstLength > Text String length** findAndCopyValue() valueOffset = 0 dstBuffer.length = 15 dstOffset = 0 dstLength = 7	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
10	**valueOffset + dstLength > Text String length** findAndCopyValue() valueOffset = 2 dstBuffer.length = 15 dstOffset = 0 dstLength = 5	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
11	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F**		
	Select a TLV (tag 02h)		
	findAndCopyValue() findAndCopyValue() tag = 0Dh occurrence = 2	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the getValueLength() method	ToolkitException.UNAVAILABLE_ELEMENT is thrown.	
12	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F**		
	Successful call findAndCopyValue() tag = 0Dh, occurrence = 1 valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17	Result of findAndCopyValue() is 17	

Id	Description	API Expectation	APDU Expectation
13	**Compare buffer** buffer = 04 00 01 … 0F	Result is 00h	
14	**Initialize dstBuffer** dstBuffer = 55 55 … 55		
	Successful call findAndCopyValue() 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** findAndCopyValue() tag = 0D buffer = 00 11 22 33 44 55 (no specific DCS byte)		
	Successful call findAndCopyValue() 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** findAndCopyValue() 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	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F**		
	Successful call (with tag 8Dh) findAndCopyValue() 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) findAndCopyValue() 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	**Invalid parameter** findAndCopyValue() occurrence = 0	ToolkitException.BAD_INPUT_PARAMETER is thrown	

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

Test Area Reference Api_2_Bte_Facrb_Bs.

5.2.10.13.1 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
```

5.2.10.13.1.1 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 Comprehension TLV is less than that in compareBuffer returns -1.
- CRRN5: if the first miscomparing byte in Comprehension TLV is greater than that in compareBuffer returns 1.
- CRRN6: The search method is comprehension required flag independent.

5.2.10.13.1.2 Parameter errors

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

5.2.10.13.1.3 Context errors

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

5.2.10.13.2 Test area files

Test Source: Test_Api_2_Bte_Facrb_Bs.java.

Test Applet: Api_2_Bte_Facrb_Bs_1.java.

Cap File: api_2_bte_facrb_bs.cap.

5.2.10.13.3 Test coverage

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

5.2.10.13.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Initialize the handler with 81 03 01 21 00 82 02 81 82**		
	findAndCompareValue() with a null dstBuffer	NullPointerException is thrown	
2	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 10 04 00 01 … 0E**		
	compareOffset > compareBuffer.length findAndCompareValue() tag = 0Dh compareBuffer.length = 20 compareOffset = 21	ArrayIndexOutOfBoundsException is thrown	
3	**compareOffset < 0** findAndCompareValue() compareBuffer.length = 20 compareOffset = -1	ArrayIndexOutOfBoundsException is thrown	
4	**length > compareBuffer.length** findAndCompareValue() compareBuffer.length = 15 compareOffset = 0	ArrayIndexOutOfBoundsException is thrown	
5	**compareOffset + length >** findAndCompareValue() compareBuffer.length compareBuffer.length = 20 compareOffset = 5	ArrayIndexOutOfBoundsException is thrown	
6	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F**		
	Select a TLV (tag 02h)		
	findAndCompareValue() tag = 03h	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
	Call the getValueLength() method	ToolkitException.UNAVAILABLE_ELEMENT is thrown.	
7	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F**		
	Initialize compareBuffer compareBuffer = 04 00 01 … 0F		
	Compare buffers findAndCompareValue() tag = 0Dh compareOffset = 0	Result is 00h	
8	**Verify current TLV** getValueLength()	Result is 17	

Id	Description	API Expectation	APDU Expectation
9	**Initialize compareBuffer** compareBuffer = 04 00 01 … 10		
	Compare buffers with same parameters	Result is -1	
10	**Initialize compareBuffer** compareBuffer = 03 00 01 … 0F		
	Compare buffers with same parameters	Result is +1	
11	**Initialize compareBuffer** compareBuffer = 55 55 04 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 55		
	Compare buffers findAndCompareValue() compareOffset = 2	Result is 00h	
12	**append a Text String TLV** tag = 0Dh buffer = 00 11 22 33 44 55		
	Initialize compareBuffer compareBuffer = 55 55 04 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 55		
	Compare buffers findAndCompareValue() compareOffset = 2	Result is 00h	
13	**Initialize compareBuffer** compareBuffer = 55 55 04 01 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 55		
	Compare buffers findAndCompareValue() compareOffset = 2	Result is -1	
14	**Initialize compareBuffer** compareBuffer = 55 55 04 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0D 10 55		
	Compare buffers findAndCompareValue() compareOffset = 2	Result is +1	
15	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F**		
	Initialize compareBuffer CompareBuffer = 04 00 01 … 0F		
	Successful call (with tag 8Dh) findAndCompareValue() tag = 8Dh compareBuffer.length = 17 compareOffset = 0	Result is 00h	
16	**Append tag 0Fh** buffer = 00 01 … 0F		
	Initialize compareBuffer compareBuffer = 00 01 … 0F		
	Successful call (with tag 8Fh) findAndCompareValue() tag = 8Fh compareBuffer.length = 16 compareOffset = 0	Result is 00h	

Id	Description	API Expectation	APDU Expectation
17	**Initialize compareBuffer** compareBuffer = 00 99 01 03 … 0F		
	Successful call (with tag 8Fh) findAndCompareValue() tag = 8Fh compareBuffer.length = 16 compareOffset = 0	Result is +1	

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

Test Area Reference Api_2_Bte_Facrbbs_Bss.

5.2.10.14.1 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,
                                compareLength)
                           throws java.lang.NullPointerException,
                                java.lang.ArrayIndexOutOfBoundsException,
                                ToolkitException
```

5.2.10.14.1.1 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 Comprehension TLV is less than that in compareBuffer -1 is returned.
- CRRN5: if the first miscomparing byte in Comprehension TLV is greater than that in compareBuffer 1 is returned.
- CRRN6: The search method is comprehension required flag independent.

5.2.10.14.1.2 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 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.

5.2.10.14.1.3 Context errors

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

5.2.10.14.2 Test area files

Test Source: Test_Api_2_Bte_Facrbbs_Bss.java.

Test Applet: Api_2_Bte_Facrbbs_Bss_1.java.

Cap File: api_2_bte_facrbbs_bss.cap.

5.2.10.14.3 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 BERTLVEdit Handler

5.2.10.14.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Initialize the handler with 81 03 01 21 00 82 02 81 82**		
	findAndCompareValue() with a null compareBuffer	NullPointerException is thrown	
2	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 10 04 00 01 … 0E**		
	compareOffset > compareBuffer.length findAndCompareValue() tag = 0Dh, occurrence = 1 valueOffset = 0 compareBuffer.length = 5 compareOffset = 6 compareLength = 0	ArrayIndexOutOfBoundsException is thrown	
3	**compareOffset < 0** findAndCompareValue() compareBuffer.length = 5 compareOffset = -1 compareLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**compareLength >compareBuffer.length** findAndCompareValue() compareBuffer.length = 5 compareOffset = 0 compareLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**compareOffset + compareLength >compareBuffer.length** findAndCompareValue() compareBuffer.length = 5 compareOffset = 3 compareLength = 3	ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
6	**compareLength < 0** findAndCompareValue() compareBuffer.length = 5 compareOffset = 0 compareLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 06 04 00 01 … 04**		
	valueOffset > Text String Length findAndCompareValue() tag = 0Dh, occurrence = 1 valueOffset = 7 compareBuffer.length = 15 compareOffset = 0 compareLength = 0	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	**valueOffset < 0** findAndCompareValue() valueOffset = -1 compareBuffer.length = 15 compareOffset = 0 compareLength = 1	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
9	**compareLength > Text String length** findAndCompareValue() valueOffset = 0 compareBuffer.length = 15 compareOffset = 0 compareLength = 7	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
10	**valueOffset + compareLength > Text String length** findAndCompareValue() valueOffset = 2 compareBuffer.length = 15 compareOffset = 0 compareLength = 5	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
11	**Invalid parameter** findAndCompareValue() occurrence = 0	ToolkitException.BAD_INPUT_PARAMETER is thrown	
12	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F**		
	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	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F**		
	Initialize 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	**Initialize compareBuffer** compareBuffer = 04 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 10		
	Compare buffers with same parameters	Result is -1	

Id	Description	API Expectation	APDU Expectation
16	**Initialize compareBuffer** compareBuffer = 03 00 01 … 0F		
	Compare buffers with same parameters	Result is +1	
17	**Initialize compareBuffer** compareBuffer = 55 55 55 01 02 03 04 05 06 07 08 09 0A 0B 0C 55 55 55 55 55		
	Compare buffers findAndCompareValue() valueOffset = 2 compareOffset = 3 compareLength = 12	Result is 00h	
18	**Initialize 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	**Initialize 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		
	Initialize compareBuffer compareBuffer = 04 00 01 … 0F		
	findAndCompareValue() tag = 0Dh, occurrence = 1 valueOffset = 0 compareOffset = 0 compareLength = 17	Result is 00h	
21	**Initialize compareBuffer** compareBuffer = 00 11 22 33 44 55		
	findAndCompareValue() tag = 0Dh, occurrence = 2 valueOffset = 0 compareOffset = 0 compareLength = 6	Result is 00h	
22	**Initialize compareBuffer** compareBuffer = 00 11 22 33 44 66		
	findAndCompareValue() tag = 0Dh, occurrence = 2 valueOffset = 0 compareOffset = 0 compareLength = 6	Result is -1	

Id	Description	API Expectation	APDU Expectation
23	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 11 04 00 01 … 0F**		
	Initialize compareBuffer `CompareBuffer = 04 00 01 … 0F`		

	Successful call (with tag 8Dh)
	`findAndCompareValue()`
	`tag = 8Dh, occurrence = 1`
	`valueOffset = 0`
	`compareBuffer.length = 17`
	`compareOffset = 0`
	`compareLength = 17`
		Result is 00h	

24	**Append tag 0Fh** `buffer = 00 01 … 0F`		
	Initialize compareBuffer `compareBuffer = 00 01 … 0F`		

	Successful call (with tag 8Fh)
	`findAndCompareValue()`
	`tag = 8Fh, occurrence = 1`
	`valueOffset = 0`
	`compareBuffer.length = 16`
	`compareOffset = 0`
	`compareLength = 16`
		Result is 00h	

25	**Initialize compareBuffer**
	`compareBuffer =0099 02 … 0F`
	findAndCompareValue()
	`tag = 0Dh, occurrence = 1`
	`valueOffset = 0`
	`compareOffset = 0`
	`compareLength = 17`
		Result is +1	

5.2.10.15 Method getCapacity

Test Area Reference: Api_2_Bte_Gcap.

5.2.10.15.1 Conformance requirement

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

```
public byte getCapacity()
```

5.2.10.15.1.1 Normal execution

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

5.2.10.15.1.2 Parameter errors

No requirements

5.2.10.15.1.3 Context errors

No requirements

5.2.10.15.2 Test area files

Test Source: Test_Api_2_Bte_Gcap.java.

Test Applet: Api_2_Bte_Gcap_1.java.

Cap File: api_2_bte_gcap.cap.

5.2.10.15.3 Test coverage

CRR number	Test case number
N1	1

5.2.10.15.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x10 and set its BER Tag to 0x01		
1	1- The applet calls getCapacity() on the BERTLVEditHandler 2- The applet fills the handler with the maximum capacity, using appendTLV() method 3- The applet calls clear() on the BERTLVEdit handler 4- The applet fills the handler with the maximum capacity plus one, using appendTLV() method	1- No exception is thrown, the capacity shall be 0x10 2- No exception is thrown 3- No exception is thrown 4- HANDLER_OVERFLOW exception is thrown	

5.2.10.16 Method getValueShort

Test Area Reference: Api_2_Bte_Gvsh.

5.2.10.16.1 Conformance requirement

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

```
public short getValueShort(short valueOffset)
                    throws ToolkitException
```

5.2.10.16.1.1 Normal execution

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

5.2.10.16.1.2 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.

5.2.10.16.1.3 Context errors

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

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

5.2.10.16.2 Test area files

Specific triggering: None.

Test Source: Test_Api_2_Bte_Gvsh.java.

Test Applet: Api_2_Bte_Gvsh_1.java.

Cap File: api_2_bte_gvsh.cap.

5.2.10.16.3 Test coverage

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

5.2.10.16.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	`Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01`		
1	**Initialize the handler with 81 03 01 FF FE 82 02 81 FD**		
	getValueShort(0)	ToolkitException.UNAVAILABLE_ELEMENT is thrown	
2	**Search TLV 01h (Command Details TLV)**		
	getValueShort(3)	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
3	**Search TLV 01h (Command Details TLV)**		
	getValueShort(1)	Result is FFh Feh (type, qualifier)	
4	**Search TLV 02h (Device Identities TLV)**		
	getValueShort(0)	Result is 81h FDh (Source, Destination)	
5	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 7F 04 00 01 … 7D Search TLV 0Dh (Text String TLV)**		
	getValueShort(7D)	Result is 7Ch 7Dh	
6	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 81 80 04 00 01 … 7E Search TLV 0Dh (Text String TLV)**		
	getValueShort(7D)	Result is 7Ch 7Dh	
7	**getValueShort(7E)**	Result is 7Dh 7Eh	
8	**Initialize the handler with 81 03 01 21 00 82 02 81 02 0D 81 F1 04 00 01 … EF Search TLV 0Dh (Text String TLV)**		
	getValueShort(EF)	Result is Eeh Efh	

5.2.10.17 Method appendArray

Test Area Reference: Api_2_Bte_Apda.

5.2.10.17.1 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
```

5.2.10.17.1.1 Normal execution

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

5.2.10.17.1.2 Parameter errors

- 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.

5.2.10.17.1.3 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.

5.2.10.17.2 Test area files

Test Source: Test_Api_2_Bte_Apda.java.

Test Applet: Api_2_Bte_Apda _1.java.

Cap File: api_2_bte_apda.cap.

5.2.10.17.3 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 BERTLVEdit Handler

5.2.10.17.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Null buffer** appendArray()	NullPointerException is thrown	
2	**offset > buffer.length** appendArray() buffer.length = 5 offset = 6 length = 0	ArrayIndexOutOfBoundsException is thrown	
3	**offset < 0** appendArray() buffer.length = 5 offset = -1 length = 1	ArrayIndexOutOfBoundsException is thrown	
4	**length > buffer.length** appendArray() buffer.length = 5 offset = 0 length = 6	ArrayIndexOutOfBoundsException is thrown	
5	**offset + length > buffer.length** appendArray() buffer.length = 5 offset = 3 length = 3	ArrayIndexOutOfBoundsException is thrown	
6	**length < 0** appendArray() buffer.length = 5 offset = 0 length = -1	ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
7	**Handler overflow exception** appendArray() buffer.length = getCapacity()+1 offset = 0 length = getCapacity()+1	ToolkitException.HANDLER_OVERFLOW is thrown	
8	**Initialize handler with 81 03 01 00 00 82 02 81 00**		
	Select Command Details TLV		
	Successful call appendArray() 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** appendArray() 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** appendArray() 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 appendArray() 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	

5.2.10.18 Method appendTLV(byte tag, byte value)

Test Area Reference: Api_2_Bte_Aptlbb.

5.2.10.18.1 Conformance requirement

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

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

5.2.10.18.1.1 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.

5.2.10.18.1.2 Parameter errors

No requirements.

5.2.10.18.1.3 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.

5.2.10.18.2 Test area files

Test Source: Test_Api_2_Bte_Aptlbb.java.

Test Applet: Api_2_Bte_Aptlbb_1.java.

Cap File: api_2_bte_aptlbb.cap.

5.2.10.18.3 Test coverage

CRR number	Test case number
N1	3, 4, 5
N2	2
C1	1
C2	Does not apply for BERTLVEdit Handler

5.2.10.18.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	`Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01`		
1	**Call appendArray()** `length = getCapacity()-1`		
	Handler Overflow exception: Call the appendTLV() method	ToolkitException.HANDLER_OVERFLOW is thrown	
2	**Initialize handler with 81 03 01 00 00 82 02 81 00**		
	Select Command Details TLV		
	Call the appendTLV() method		
	Verify Current TLV: Call getValueLength()	Result is 03h	
3	**Clear the handler**		

	Successful call
	`appendTLV()`
	`tag = 84h`
	`value = 00h`		

	Call copy() method		
	Compare the arrays `compareBuffer = 84 01 00`	Result of javacard.framework.Util.arrayCompare() is 00h	

Id	Description	API Expectation	APDU Expectation
4	**Successful call** appendTLV() 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	
5	**Clear the handler** **Call appendArray()** length = 250 buffer = 00 81 F7 03 04 … F9 **Successful call** appendTLV() 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	

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

Test Area Reference: Api_2_Bte_Aptlbbb.

5.2.10.19.1 Conformance requirements

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

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

5.2.10.19.1.1 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.

5.2.10.19.1.2 Parameter errors

No requirements.

5.2.10.19.1.3 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.

5.2.10.19.2 Test area files

Test Source: Test_Api_2_Bte_Aptlbbb.java.

Test Applet: Api_2_Bte_Aptlbbb_1.java.

Cap File: api_2_bte_aptlbbb.cap.

5.2.10.19.3 Test coverage

CRR number	Test case number
N1	3, 4, 5
N2	2
C1	1
C2	Does not apply for BERTLVEdit Handler

5.2.10.19.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Call the appendArray()** length = getCapacity()-1		
	Handler Overflow exception: Call the appendTLV() method	ToolkitException.HANDLER_OVERFLOW is thrown	
2	**Initialize handler with 81 03 01 00 00 82 02 81 00**		
	Select Command Details TLV		
	Call the appendTLV() method		
	Verify Current TLV: Call getValueLength()	Result is 03h	
3	**Clear the handler**		
	Successful call appendTLV() 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** appendTLV() 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 appendTLV() 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	

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

Test Area Reference: Api_2_Bte_Aptlb_Bss.

5.2.10.20.1 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
```

5.2.10.20.1.1 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.

5.2.10.20.1.2 Parameter errors

- 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.

5.2.10.20.1.3 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.
- CRRC3: if valuelength is greater than 255, a ToolkitException is thrown with reason code BAD_INPUT_PARAMETER.

5.2.10.20.2 Test area files

Test Source: Test_Api_2_Bte_Aptlb_Bss.java.

Test Applet: Api_2_Bte_Aptlb_Bss_1.java.

Cap File: api_2_bte_aptlb_bss.cap.

5.2.10.20.3 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 BERTLVEdit Handler
C3	8

5.2.10.20.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Null value** appendTLV()	NullPointerException is thrown	
2	**valueOffset > value.length** appendTLV() value.length = 5 valueOffset = 6 valueLength = 0	ArrayIndexOutOfBoundsException is thrown	
3	**valueOffset < 0** appendTLV() value.length = 5 valueOffset = -1 valueLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**valueLength > value.length** appendTLV() value.length = 5 valueOffset = 0 valueLength = 6	ArrayIndexOutOfBoundsException is thrown	
5	**valueOffset + valueLength > value.length** appendTLV() value.length = 5 valueOffset = 3 valueLength = 3	ArrayIndexOutOfBoundsException is thrown	
6	**valueLength < 0** appendTLV() value.length = 5 valueOffset = 0 valueLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Handler overflow exception** Call the appendArray() method, length = getCapacity()-1 appendTLV() value.length = 254 valueOffset = 0 valueLength = 254	ToolkitException.HANDLER_OVERFLOW is thrown	
8	**Bad parameter exception** Clear the handler appendTLV() value.length = 256 valueOffset = 0 valueLength = 256	ToolkitException.BAD_INPUT_PARAMETER is thrown	
9	**Initialize handler with 81 03 01 00 00 82 02 81 00**		
	Select Command Details TLV		
	Successful call appendTLV() tag = 04 value = FF FE … F8 valueOffset = 0 valueLength = 8		
	Verify Current TLV: Call getValueLength()	Result is 03h	
10	**Clear the handler**		
	Successful call appendTLV() 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	

Id	Description	API Expectation	APDU Expectation
11	**Successful call** appendTLV() 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** appendTLV() 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 appendTLV() 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	
14	**Clear the handler** **Successful call** appendTLV() 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	

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

Test Area Reference: Api_2_Bte_Aptlbb_Bss.

5.2.10.21.1 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
```

5.2.10.21.1.1 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.

5.2.10.21.1.2 Parameter errors

- 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.

5.2.10.21.1.3 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.

5.2.10.21.2 Test area files

Test Source: Test_Api_2_Bte_Aptlbb_Bss.java.

Test Applet: Api_2_Bte_Aptlbb_Bss_1.java.

Cap File: api_2_bte_aptlbb_bss.cap.

5.2.10.21.3 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 BERTLVEdit Handler
C3	8

5.2.10.21.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	`Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01`		
1	**Null value2** `appendTLV()`	NullPointerException is thrown	

2	**value2Offset > value2.length**
	`appendTLV()`
	`value2.length = 5`
	`value2Offset = 6`
	`value2Length = 0`
		ArrayIndexOutOfBoundsException is thrown	

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

Id	Description	API Expectation	APDU Expectation
4	**value2Length > value2.length** appendTLV() value2.length = 5 value2Offset = 0 value2Length = 6	ArrayIndexOutOfBoundsException is thrown	
5	**value2Offset + value2Length > value2.length** appendTLV() value2.length = 5 value2Offset = 3 value2Length = 3	ArrayIndexOutOfBoundsException is thrown	
6	**value2Length < 0** appendTLV() value2.length = 5 value2Offset = 0 value2Length = -1	ArrayIndexOutOfBoundsException is thrown	
7	**Handler overflow exception** Call the appendArray() method, length = getCapacity()-1 appendTLV() value2.length = 254 value2Offset = 0 value2Length = 254	ToolkitException.HANDLER_OVERFLOW is thrown	
8	**Bad parameter exception** Clear the handler appendTLV() value2.length = 256 value2Offset = 0 value2Length = 256	ToolkitException.BAD_INPUT_PARAMETER is thrown	
9	**Initialize handler with** **81 03 01 00 00 82 02 81 00**		
	Select Command Details TLV		
	Successful call appendTLV() 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 appendTLV() 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	
11	**Successful call** appendTLV() 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	

Id	Description	API Expectation	APDU Expectation
12	**Successful call** appendTLV() 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 appendTLV() 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 appendTLV() 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	

5.2.10.22 Method appendTLV(byte tag, byte value1, short value2)

Test Area Reference: Api_2_Bte_Aptlbbs.

5.2.10.22.1 Conformance requirements

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

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

5.2.10.22.1.1 Normal execution

- CRRN1: Appends a TLV element to the current TLV list (3-byte element(1-byte,1-short)).
- CRRN2: A successful append does not modify the TLV selected.

5.2.10.22.1.2 Parameter errors

No requirements

5.2.10.22.1.3 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.

5.2.10.22.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Bte_Aptlbbs.java.

Test Applet: Api_2_Bte_Aptlbbs_1.java.

Cap File: api_2_bte_aptlbbs.cap.

5.2.10.22.3 Test coverage

CRR number	Test case number
N1	3, 4, 5
N2	2
C1	1
C2	Does not apply for BERTLVEdit Handler

5.2.10.22.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Call the appendArray() length = getCapacity()-1**		
	Handler Overflow exception: Call the appendTLV() method	ToolkitException.HANDLER_OVERFLOW is thrown	
2	**Initialize handler with 81 03 01 00 00 82 02 81 00**		
	Select Command Details TLV		
	Call the appendTLV() method		
	Verify Current TLV: Call getValueLength()	Result is 03h	
3	**Clear the handler**		
	Successful call appendTLV() tag = 84h value1 = 00h **value2 = 01h 02h**		
	Call copy() method		
	Compare the arrays compareBuffer = 84 03 00 01 02	Result of javacard.framework.Util.arrayCompare() is 00h	
4	**Successful call** appendTLV() tag = 01h value1 = FEh value2 = FDh FCh		
	Call copy() method		
	Compare the arrays compareBuffer = 84 03 00 01 02 01 03 FE FD FC	Result of javacard.framework.Util.arrayCompare() is 00h	

Id	Description	API Expectation	APDU Expectation
5	**Clear the handler**		
	Call appendArray() length = 248 buffer = 00 81 F5 03 04 … F7		
	Successful call appendTLV() tag = 84h value1 = 00h value2 = 01h 02h		
	Call getLength() method	result = 253	
	Call copy() method		
	Compare handler compareBuffer = 00 81 F5 03 04 … F7 84 03 00 01 02	Result of javacard.framework.Util.arrayCompare() is 00h	

5.2.10.23 Method appendTLV(byte tag, short value)

Test Area Reference: Api_2_Bte_Aptlbs.

5.2.10.23.1 Conformance requirements

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

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

5.2.10.23.1.1 Normal execution

- CRRN1: Appends a TLV element to the current TLV list (2-byte or 1-short element).
- CRRN2: A successful append does not modify the TLV selected.

5.2.10.23.1.2 Parameter errors

No requirements

5.2.10.23.1.3 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.

5.2.10.23.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Bte_Aptlbs.java.

Test Applet: Api_2_Bte_Aptlbs_1.java.

Cap File: api_2_bte_aptlbs.cap.

5.2.10.23.3 Test coverage

CRR number	Test case number
N1	3, 4, 5
N2	2
C1	1
C2	Does not apply for BERTLVEdit Handler

5.2.10.23.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Call appendArray()** length = getCapacity()-1		
	Handler Overflow exception: **Call the appendTLV() method**	ToolkitException.HANDLER_OVERFLOW is thrown	
2	**Initialize handler with** **81 03 01 00 00 82 02 81 00**		
	Select Command Details TLV		
	Call the appendTLV() method		
	Verify Current TLV: Call getValueLength()	Result is 03h	
3	**Clear the handler**		
	Successful call appendTLV() tag = 84h value = 00h 01h		
	Call copy() method		
	Compare the arrays **compareBuffer = 84 02 00 01**	Result of javacard.framework.Util.arrayCompare() is 00h	
4	**Successful call** appendTLV() tag = 01h value = FEh FFh		
	Call copy() method		
	Compare the arrays **compareBuffer = 84 02 00 01 01 02 FE FF**	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 appendTLV() tag = 84h value = 00h 01h		
	Call getLength() method	result = 253	
	Call copy() method		
	Compare the array **compareBuffer = 00 81 F6 03 04 … F8 84 02 00 01**	Result of javacard.framework.Util.arrayCompare() is 00h	

5.2.10.24 Method appendTLV(byte tag, short value1, short value2)

Test Area Reference: Api_2_Bte_Aptlbss.

5.2.10.24.1 Conformance requirements

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

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

5.2.10.24.1.1 Normal execution

- CRRN1: Appends a TLV element to the current TLV list (4-byte element(2-short)).
- CRRN2: A successful append does not modify the TLV selected.

5.2.10.24.1.2 Parameter errors

No requirements

5.2.10.24.1.3 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.

5.2.10.24.2 Test area files

Specific triggering: Unrecognized Envelope:

Test Source: Test_Api_2_Bte_Aptlbss.java.

Test Applet: Api_2_Bte_Aptlbss_1.java.

Cap File: api_2_bte_aptlbss.cap.

5.2.10.24.3 Test coverage

CRR number	Test case number
N1	3, 4, 5
N2	2
C1	1
C2	Does not apply for BERTLVEdit Handler

5.2.10.24.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	`Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01`		
1	**Call the appendArray()** `length = getCapacity()-1` **Handler Overflow exception:** **Call the appendTLV() method**	ToolkitException.HANDLER_OVERFLOW is thrown	
2	**Initialize handler with** **81 03 01 00 00 82 02 81 00** **Select Command Details TLV** **Call the appendTLV() method** **Verify Current TLV: Call getValueLength()**	Result is 03h	

3	**Clear the handler**
	Successful call
	`appendTLV()`
	`tag = 84h`
	`value1 = 00h 01h`
	`value2 = 02h 03h`
	Call copy() method
	Compare the arrays
	compareBuffer = 84 04 00 01 02 03
		Result of javacard.framework.Util.arrayCompare() is 00h	

4	**Successful call**
	`appendTLV()`
	`tag = 01h`
	`value1 = FEh FDh`
	`value2 = FCh FBh`
	Call copy() method
	Compare the arrays
	compareBuffer = 84 04 00 01 02 03 01 04 FE FD FC FB
		Result of javacard.framework.Util.arrayCompare() is 00h	

Id	Description	API Expectation	APDU Expectation
5	**Clear the handler** **Call appendArray()** length = 247 buffer = 00 81 F4 03 04 … F6 **Successful call** appendTLV() tag = 84h value1 = 00h 01h value2 = 02h 03h **Call getLength() method** **Call copy() method** **Compare handler** **compareBuffer = 00 81 F4 03 04 … F6 84 04 00 01 02 03**	result = 253 Result of javacard.framework.Util.arrayCompare() is 00h	

5.2.10.25 Method appendTLV(byte tag, byte[] value1, short value1Offset, short value1Length, byte[] value2, short value2Offset, short value2Length)

Test Area Reference: Api_2_Bte_Aptlb_Bss_Bss.

5.2.10.25.1 Conformance requirements

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

```
public void appendTLV(byte tag,
                      byte[] value1,
                      short value1Offset,
                      short value1Length,
                      byte[] value2,
                      short value2Offset,
                      short value2Length)
               throws java.lang.NullPointerException,
                      java.lang.ArrayIndexOutOfBoundsException,
                      ToolkitException
```

5.2.10.25.1.1 Normal execution

- CRRN1: Appends a TLV element to the current TLV list (2 byte arrays format).
- CRRN2: A successful append does not modify the TLV selected.

5.2.10.25.1.2 Parameter errors

- CRRP1: If value1 or value2 is null, a NullPointerException is thrown.
- CRRP2: If value1Offset or value1Length or both would cause access outside value1 array bounds, or if value1Length is negative, an ArrayIndexOutOfBoundsException is thrown.
- CRRP3: If value2Offset or value2Length or both would cause access outside value2 array bounds, or if value2Length is negative, an ArrayIndexOutOfBoundsException is thrown.

5.2.10.25.1.3 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 value1Length or value2Length is greater than 255, a ToolkitException is thrown with reason code BAD_INPUT_PARAMETER.

5.2.10.25.2 Test area files

Test Source: Test_Api_2_Bte_Aptlb_Bss_Bss.java.

Test Applet: Api_2_Bte_Aptlb_Bss_Bss_1.java.

Cap File: api_2_bte_aptlb_bss_bss.cap.

5.2.10.25.3 Test coverage

CRR number	Test case number
N1	18, 19, 20, 21
N2	17
P1	1, 2
P2	3, 4, 5, 6, 7
P3	8, 9, 10, 11, 12
C1	13
C2	Does not apply for BERTLVEdit Handler
C3	14, 15

5.2.10.25.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Null value1** appendTLV()	NullPointerException is thrown	
2	**Null value2** appendTLV()	NullPointerException is thrown	
3	**Value1Offset ≥ value1.length** appendTLV() value1.length = 5 value1Offset = 5 value1Length = 1 value2.length = 5 value2Offset = 0 value2Length = 1	ArrayIndexOutOfBoundsException is thrown	
4	**Value1Offset < 0** appendTLV() value1.length = 5 value1Offset = -1 value1Length = 1 value2.length = 5 value2Offset = 0 value2Length = 1	ArrayIndexOutOfBoundsException is thrown	
5	**Value1Length > value1.length** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 6 value2.length = 5 value2Offset = 0 value2Length = 1	ArrayIndexOutOfBoundsException is thrown	
6	**Value1Offset + value1Length > value1.length** appendTLV() Value1.length = 5 value1Offset = 3 value1Length = 3 value2.length = 5 value2Offset = 0 value2Length = 1	ArrayIndexOutOfBoundsException is thrown	

Id	Description	API Expectation	APDU Expectation
7	**Value1Length < 0** appendTLV() value1.length = 5 value1Offset = 0 value1Length = -1 value2.length = 5 value2Offset = 0 value2Length = 1	ArrayIndexOutOfBoundsException is thrown	
8	**Value2Offset ≥ value2.length** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 value2.length = 5 value2Offset = 5 value2Length = 1	ArrayIndexOutOfBoundsException is thrown	
9	**Value2Offset < 0** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 value2.length = 5 value2Offset = -1 value2Length = 1	ArrayIndexOutOfBoundsException is thrown	
10	**Value2Length > value2.length** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 value2.length = 5 value2Offset = 0 value2Length = 6	ArrayIndexOutOfBoundsException is thrown	
11	**Value2Offset + value2Length > value2.length** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 Value2.length = 5 Value2Offset = 3 Value2Length = 3	ArrayIndexOutOfBoundsException is thrown	
12	**Value2Length < 0** appendTLV() value1.length = 5 value1Offset = 0 value1Length = 1 value2.length = 5 value2Offset = 0 value2Length = -1	ArrayIndexOutOfBoundsException is thrown	
13	**Handler overflow exception** Call the appendArray() method, length = getCapacity()-1 appendTLV() Value1.length = 256 Value1Offset = 0 Value1Length = 253 Value2.length = 256 Value2Offset = 0 Value2Length = 1	ToolkitException.HANDLER_OVERFLOW is thrown	
14	**Bad parameter exception** Clear the handler appendTLV() Value1.length = 256 Value1Offset = 0 Value1Length = 256 Value2.length = 256 Value2Offset = 0 Value2Length = 1	ToolkitException.BAD_INPUT_PARAMETER is thrown	
15	**Bad parameter exception** appendTLV() Value1.length = 256 Value1Offset = 0 Value1Length = 1 Value2.length = 256 Value2Offset = 0 Value2Length = 256	ToolkitException.BAD_INPUT_PARAMETER is thrown	

Id	Description	API Expectation	APDU Expectation
16	**clear the handler, append the handler with TLVs:** 81 03 11 22 33 82 02 99 77 **Select Command Details TLV**	No exceptions shall be thrown	
17	Successful call **Clear the handler** appendTLV() tag = 04 value1 = FF FE … F8 value1Offset = 0 value1Length = 8 value2 = F7 F6 … F0 value2Offset = 0 value2Length = 8 **Verify Current TLV: Call getValueLength()** **Clear the handler**	Result is 10h	
18	**Successful call** appendTLV() tag = 04 value1 = FF FE … F8 value1Offset = 0 value1Length = 8 value2 = F7 F6 … F0 value2Offset = 0 value2Length = 8 **Call copy() method** **Compare handler** **CompareBuffer = 04 10 FF FE … F0**	Result is 00	
19	**Successful call** appendTLV() tag = 85h value1 = 00 01 … 07 value1Offset = 2 value1Length = 6 value2 = 08 09 … 0F value2Offset = 2 value2Length = 6 **Call copy() method** **Compare handler** **compareBuffer = 04 10 FF FE … F0 85 0C 02 03 04 05 06 07 0A 0B 0C 0D 0E 0F**	Result is 00	
20	**Successful call** appendTLV() tag = 01 value1 = 11 22 … 88 value1Offset = 2 value1Length = 4 value2 = 99 AA … FF 00 value2Offset = 2 value2Length = 4 **Call copy() method** **Compare handler** **compareBuffer = 04 10 FF FE … F0 85 0C 02 03 04 05 06 07 0A 0B 0C 0D 0E 0F 01 08 33 44 55 66 BB CC DD EE** **Clear the handler**	Result is 00	
21	**Successful call** appendTLV() tag = 04 value1 = 00 01 … 7F value1Offset = 0 value1Length = 80h value2 = 80 81 … FC value2Offset = 0 value2Length = 7Dh **Call copy() method** **Compare handler** **compareBuffer = 04 81 FD 00 01…FC**	Result is 00	

5.2.10.26 Method clear

Test Area Reference: Api_2_Bte_Cler.

5.2.10.26.1 Conformance requirement

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

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

5.2.10.26.1.1 Normal execution

- CRRN1: Clears the TLV list of an EditHandler.
- CRRN2: Resets the current TLV selected.

5.2.10.26.1.2 Parameter errors

No requirements.

5.2.10.26.1.3 Context errors

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

5.2.10.26.2 Test area files

Test Source: Test_Api_2_Bte_Cler.java.

Test Applet: Api_2_Bte_Cler_1.java.

Cap File: api_2_bte_cler.cap.

5.2.10.26.3 Test coverage

CRR number	Test case number
N1	1
N2	2
C1	Does not apply for BERTLVEdit Handler

5.2.10.26.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Get a BERTLVEditHandler using buildTLVHandler() with type BER_EDIT_HANDLER and capacity 0x100 and set its BER Tag to 0x01		
1	**Initialize the handler with 81 03 01 00 00 82 02 81 00** 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	

5.2.11 Interface BERTLVViewHandler

Tests are done in inheriting interfaces BERTLVEditHandler and envelopeHandler.

5.2.12 Class EnvelopeHandlerSystem

5.2.12.1 Method getTheHandler

Test Area Reference: Api_2_Ehs_Gthd.

5.2.12.1.1 Conformance requirements

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

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

5.2.12.1.1.1 Normal execution

- CRRN1: The method shall return the single system instance of the class implementing the EnvelopeHandler interface.
- CRRN2: The EnvelopeHandler is a Temporary JCRE Entry Point Object (see Javacard 2.2.1 Runtime Environment (JCRE) Specification [2]).

5.2.12.1.1.2 Parameter errors

No requirements.

5.2.12.1.1.3 Context errors

- CRRC1: The method shall thrown ToolkitException.HANDLER_NOT_AVAILABLE if the handler is not available.

5.2.12.1.2 Test area files

Test Source: Test_Api_2_Ehs_Gthd.java.

Test Applet: Api_2_Ehs_Gthd_1.java.

Cap File: api_2_ehs_gthd.cap.

5.2.12.1.3 Test coverage

CRR number	Test case number
N1	1, 2, 3
N2	checked in CAT Runtime Environment: Cre_Api_Hepo (Test case 1 and 2)
C1	checked in CAT Runtime Environment: Cre_Mha_Enhd (Test case 1)

5.2.12.1.4 Test procedure

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

5.2.13 Class EnvelopeResponseHandlerSystem

5.2.13.1 Method getTheHandler

Test Area Reference: Api_2_Ers_Gthd.

5.2.13.1.1 Conformance requirement

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

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

5.2.13.1.1.1 Normal execution

- CRRN1: The method shall return the single system instance of the class implementing the EnvelopeResponseHandler interface.
- CRRN2: The EnvelopeResponseHandler is a Temporary JCRE Entry Point Object (see Javacard 2.2.1 Runtime Environment (JCRE) Specification [2]).

5.2.13.1.1.1 Parameter errors

No requirements.

5.2.13.1.1.3 Context errors

- CRRC1: The method shall thrown ToolkitException.HANDLER_NOT_AVAILABLE if the handler is not available.
- CRRC2: After the first invocation of the ProactiveHandler.send method the EnvelopeResponseHandler is no more available.

5.2.13.1.2 Test area files

Test Source: Test_Api_2_Ers_Gthd.java.

Test Applet: Api_2_Ers_Gthd_1.java.

Cap File: api_2_ers_gthd.cap.

5.2.13.1.3 Test coverage

CRR number	Test case number
N1	1, 2, 3
N2	checked in CAT Runtime Environment: Cre_Api_Hepo (Test case 3 and 4)
C1	checked in CAT Runtime Environment: Cre_Mha_Erhd (Test case 1)
C2	4

5.2.13.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call getTheHandler() twice**	The returned objects shall be the same	
2	**Verify that getTheHandler returns an EnvelopeResponseHandler** `Call getTheHandler() method`	The reference returned shall be an object implementing the EnvelopeResponseHandler interface (check cast)	
3	**Verify the returned value is not null** `Call getTheHandler() method`	The reference returned shall not be null.	

Id	Description	API Expectation	APDU Expectation
4	**Send a proactive command, and then, Call getTheHandler()**	ToolkitException.HANDLER_NOT_AVAILABLE is thrown	

5.2.14 Class ProactiveHandlerSystem

5.2.14.1 Method getTheHandler

Test Area Reference: Api_2_Phs_Gthd.

5.2.14.1.1 Conformance requirement

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

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

5.2.14.1.1.1 Normal execution

- CRRN1: The method shall return the single system instance of the class implementing the ProactiveHandler interface.
- CRRN2: The ProactiveHandler is a Temporary JCRE Entry Point Object (see Javacard 2.2.1 Runtime Environment (JCRE) Specification [2]).

5.2.14.1.1.2 Parameter errors

No requirements.

5.2.14.1.1.3 Context errors

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

5.2.14.1.2 Test area files

Test Source: Test_Api_2_Phs_Gthd.java.

Test Applet: Api_2_Phs_Gthd_1.java.

Cap File: api_2_phs_gthd.cap.

5.2.14.1.3 Test coverage

CRR number	Test case number
N1	1, 2, 3
N2	checked in CAT Runtime Environment: Cre_Api_Hepo (Test case 5 and 6)
C1	checked in CAT Runtime Environment: Cre_Mha_Pahd (Test case 1)

5.2.14.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call getTheHandler() method twice**	The returned objects shall be the same	
2	**Call getTheHandler() method**	The reference shall be an objetc implementing the ProactiveHandler interface	
3	**Call getTheHandler() method**	The reference shall not be null	

5.2.15 Class ProactiveResponseHandlerSystem

5.2.15.1 Method getTheHandler

Test Area Reference: Api_2_Prs_Gthd.

5.2.15.1.1 Conformance requirement

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

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

5.2.15.1.1.1 Normal execution

- CRRN1: The method shall return the single system instance of the object implementing the ProactiveHandler interface.
- CRRN2: The ProactiveResponseHandler is a Temporary JCRE Entry Point Object (see Javacard 2.2.1 Runtime Environment (JCRE) Specification [2]).

5.2.15.1.1.2 Parameter errors

No requirements.

5.2.15.1.1.3 Context errors

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

5.2.13.1.2 Test area files

Test Source: Test_Api_2_Prs_Gthd.java.

Test Applet: Api_2_Prs_Gthd_1.java.

Cap File: api_2_prs_gthd.cap.

5.2.15.1.3 Test coverage

CRR number	Test case number
N1	1, 2, 3
N2	checked in CAT Runtime Environment: Cre_Api_Hepo (Test case 7 and 8)
C1	checked in CAT Runtime Environment: Cre_Mha_Prhd (Test case 1)

5.2.15.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Build and send a Proactive Command**		Proactive Command
	Terminal Response		
	Call getTheHandler() twice method	The returned objects shall be the same	
2	**Call getTheHandler() method**	The reference shall be an object implementing the ProactiveResponseHandler interface	
3	**Call getTheHandler() method**	The reference shall not be null	

5.2.16 Class TerminalProfile

5.2.16.1 Method check(byte index)

Test Area Reference: Api_2_Tep_Checb.

5.2.16.1.1 Conformance requirement

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

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

5.2.16.1.1.1 Normal execution

- CRRN1: The method checks a facility in the handset profile: returns true if supported and false otherwise.
- CRRN2: returns false if facility-index is outside Terminal Profile data.

5.2.16.1.1.2 Parameter errors

- CRRP1: shall throw BAD_INPUT_PARAMETER ToolkitException if index has a negative value.

5.2.16.1.1.3 Context errors

- CRRC1: shall throw TERMINAL_PROFILE_NOT_AVAILABLE ToolkitException if Terminal Profile data are not available.

5.2.16.1.2 Test area files

Specific triggering: EVENT_STATUS_COMMAND:

Test Source: Test_Api_2_Tep_Checb.java.

Test Applet: Api_2_Tep_Checb_1.java.

Cap File: api_2_tep_checb.cap.

5.2.16.1.3 Test coverage

CRR number	Test case number
N1	2,3
N2	4
P1	5
C1	1

5.2.16.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**No Terminal Profile is registered** 1- Trigger applet with status command 2- Call check() method with Index = 1	1- applet is triggered 2- TERMINAL_PROFILE_NOT_AVAILABLE ToolkitException is thrown	
2	**Terminal Profile, Facility is supported** 1- Trigger applet with EVENT_UNRECOGNIZED_ENVELOPE 2- Call check() method with index = 0	1- applet is triggered 2- returns true	

Id	Description	API Expectation	APDU Expectation
3	**Facility is not supported** `Call check() method with index = 15`	returns false	
4	**Facility index is outside TerminalProfile data** `Call check() method with index = 0x7F`	Returns false	
5	**Index has a negative value** `Call check() method with index = -1`	Throws a ToolkitException with BAD_INPUT_PARAMETER reason code.	

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

Test Area Reference: Api_2_Tep_Chec_Bss.

5.2.16.2.1 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
```

5.2.16.2.1.1 Normal execution

- CRRN1: The method checks all the facilities corresponding to bits set to 1 in the mask buffer: returns true if the bitwise AND of the TerminalProfile data padded with 0 and the mask is equal to the mask , false otherwise.
- CRRN2: The method returns true if the length is equal to 0.

5.2.16.2.1.2 Parameter errors

- CRRP1: The method shall throw java.lang.NullPointerException if mask is null.
- CRRP2: The method shall throw java.lang.ArrayIndexOutOfBoundsException if check would cause access of data outside mask array bounds.
- CRRP3: If offset or length parameter is negative an ArrayIndexOutOfBoundsException exception is thrown and no check is performed.
- CRRP4: If offset+length is greater than mask.length, the length of the mask array an ArrayIndexOutOfBoundsException exception is thrown and no check is performed.

5.2.16.2.1.3 Context errors

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

5.2.16.2.2 Test area files

Specific triggering: MENU_SELECTION:

Test Source: Test_Api_2_Tep_Chec_Bss.java.

Test Applet: Api_2_Tep_Chec_Bss_1.java.

Cap File: api_2_tep_chec_bss.cap.

5.2.16.2.3 Test coverage

CRR number	Test case number
N1	9, 10, 11
N2	8
P1	2
P2	3, 4, 5, 6
P3	4, 7
P4	6
C1	1

5.2.16.2.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**No Terminal Profile is registered** Triggered by Menu Selection Call check() method: mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F offset = 0 length = 16	TERMINAL_PROFILE_NOT_AVAILABLE ToolkitException is thrown	
2	**NULL as parameter to check** Call check() method: mask= NULL	NullPointerException is thrown	
3	**Offset > mask.length** Call check() method: mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F offset = 17	ArrayIndexOutOfBoundsException is thrown	
4	**Offset < 0** Call check() method: mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F offset = -1	ArrayIndexOutOfBoundsException is thrown	
5	**Length > mask.length** Call check() method: mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F offset = 0 length = 18	ArrayIndexOutOfBoundsException is thrown	
6	**Offset + length > mask.length** Call check() method: mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F offset = 9 length = 9	ArrayIndexOutOfBoundsException is thrown	
7	**Length < 0** Call check() method: mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F offset = 0 length = -1	ArrayIndexOutOfBoundsException is thrown	
8	**length = 0** call check() method: mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F offset = 0 length = 0	Returns true	
9	**Check all the Terminal Profile** Call check() method: mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F offset = 0 length = 16	Returns false because facility 15 is not supported	
10	**Check a part of the Terminal Profile** Call check() method: mask = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F offset = 15 length = 2	Returns true: the 16 first facilities except facility 15 have been successfully checked	

Id	Description	API Expectation	APDU Expectation

11	**Check a part of the Terminal Profile**
	
	`Call check() method:`
	`mask = 0x0080`
	`offset = 0`
	`length = 2`
		Returns false: only facility 15 is checked and not supported.	

5.2.16.3 Method check(short index)

Test Area Reference: Api_2_Tep_Checs.

5.2.16.3.1 Conformance requirement

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

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

5.2.16.3.1.1 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 TerminalProfile data.
- CRRN2: returns false if facility-index is outside Terminal Profile data.

5.2.16.3.1.2 Parameter errors

- CRRP1: shall throw BAD_INPUT_PARAMETER ToolkitException if index has a negative value.

5.2.16.3.1.3 Context errors

- CRRC1: shall throw TERMINAL_PROFILE_NOT_AVAILABLE ToolkitException if Terminal Profile data are not available.

5.2.16.3.2 Test area files

Specific triggering: EVENT_STATUS_COMMAND.

Test Source: Test_Api_2_Tep_Checs.java.

Test Applet: Api_2_Tep_Checs_1.java.

Cap File: api_2_tep_checs.cap.

5.2.16.3.3 Test coverage

CRR number	Test case number
N1	2, 3, 4
N2	
P1	
C1	1

5.2.16.3.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**No Terminal Profile is registered** 1- Triggerapplet with status command 2- Call check() method with index = 1	1- applet is triggered 2- TERMINAL_PROFILE_NOT_AVAILABLE ToolkitException is thrown	
2	**Terminal Profile, Facility is supported** 1- Trigger applet with unrecognized envelope 2- Call check() method with index = 0	Returns true	
3	**Facility is not supported** Call check() method with index = 15	Returns false	
4	**Facility index is outside TerminalProfile data** Call check() method with index = 0x0099	Returns false	
5	**Index has a negative value** Call check() method with index = -1	Throws a ToolkitException with BAD_INPUT_PARAMETER reason code.	

5.2.16.4 Method getValue(short indexMSB, short indexLSB)

Test Area Reference: Api_2_Tep_Gval.

5.2.16.4.1 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
```

5.2.16.4.1.1 Normal execution

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

5.2.16.4.1.2 Parameter errors

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

5.2.16.4.1.3 Context errors

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

5.2.16.4.2 Test area files

Specific triggering: EVENT_STATUS_COMMAND.

Test Source: Test_Api_2_Tep_Gval.java.

Test Applet: Api_2_Tep_Gval_1.java.

Cap File: api_2_tep_gval.cap.

5.2.16.4.3 Test coverage

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

5.2.16.4.4 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 Call getValue() method: indexMSB = 15, indexLSB = 0	TERMINAL_PROFILE_NOT_AVAILABLE ToolkitException is thrown	
2	**Retrieve number of character down Terminal display in Terminal Profile which is 13** Call getValue() method: indexMSB = 108, indexLSB = 104	Returns 13	
3	**Retrieve byte 3 and byte 4 from terminal profile.** Byte 3 = 0xD2, Byte 4 = 0xF0 Call getValue() method: indexMSB = 31, indexLSB = 16	Returns 0xF0D2	
4	**indexMSB is negative** call getValue() method: indexMSB = 0xFFFF, indexLSB = 0xFFFD	BAD_INPUT_PARAMETER ToolkitException is thrown	
5	**indexLSB is negative** Call getValue()method: indexMSB = 0x0002, indexLSB = 0xFFFD	BAD_INPUT_PARAMETER ToolkitException is thrown	
6	**indexMSB < indexLSB** Call getValue() method: indexMSB = 0x0002, indexLSB = 0x0003	BAD_INPUT_PARAMETER ToolkitException is thrown	
7	**indexMSB > indexLSB + 16** Call getValue() method: indexMSB = 0x0021, indexLSB = 0x0010	BAD_INPUT_PARAMETER ToolkitException is thrown	
8	**indexMSB = indexLSB + 16** Call getValue() method: indexMSB = 0x0020, indexLSB = 0x0010	BAD_INPUT_PARAMETER ToolkitException is thrown	
9	**indexMSB is outside data available** Call getValue() method: indexMSB = 121, indexLSB = 115	Returns 0x001F	

5.2.16.5 Method copy(short startOffset, byte[] dstBuffer, short dstOffset, short dstLength)

Test Area Reference: Api_2_Tep_Copy.

5.2.16.5.1 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
```

5.2.16.5.1.1 Normal execution

- CRRN1: The method copies a part of the handset profile in a buffer.
- CRRN2: The method returns dstOffset + dstLength.

5.2.16.5.1.2 Parameter errors

- 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.*

5.2.16.5.1.3 Context errors

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

5.2.16.5.2 Test area files

Specific triggering: EVENT_STATUS_COMMAND.

Test Source: Test_Api_2_Tep_Copy.java.

Test Applet: Api_2_Tep_Copy_1.java.

Cap File: api_2_tep_copy.cap.

5.2.16.5.3 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

5.2.16.5.4 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 Call copy() method: startOffset = 0 dstBuffer.length = 6 dstOffset = 0 dstLength = 6	TERMINAL_PROFILE_NOT_AVAILABLE ToolkitException is thrown	
2	**dstBuffer is null**	NullPointerException is thrown	
3	**dstOffset ≥ dstBuffer.length** Call copy() method: startOffset = 0 dstBuffer.length = 5 dstOffset = 5 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	
4	**dstOffset < 0** Call copy() method: startOffset = 0 dstBuffer.length = 5 dstOffset = -1 dstLength = 1	ArrayIndexOutOfBoundsException is thrown	
5	**dstLength < 0** Call copy() method: startOffset = 0 dstBuffer.length = 5 dstOffset = 1 dstLength = -1	ArrayIndexOutOfBoundsException is thrown	
6	**dstLength >dstBuffer.length** Call copy() method: startOffset = 0 dstBuffer.length = 5 dstOffset = 0 dstLength = 6	ArrayIndexOutOfBoundsException is thrown	
7	**dstOffset + dstLength >dstBuffer.length** Call copy() method: startOffset = 0 dstBuffer.length = 5 dstOffset = 3 dstLength = 3	ArrayIndexOutOfBoundsException is thrown	
8	**Successful call extreme values** Call copy() method: startOffset = 0 dstBuffer.length = 6 dstOffset = 0 dstLength = 6	Result of copy() is 6	
9	**Successful call any values** Call copy() method: startOffset = 1 dstBuffer.length = 20 dstOffset = 3 dstLength = 4	Result of copy() is 7	
10	**Successful call, copy with length =0** Call copy() method: startOffset = 0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0	Result of copy() is 20	
11	**Value outside ProfileDownload data available** Call copy() method: startOffset = 13 dstBuffer.length = 6 dstOffset = 0 dstLength = 6	Result of copy() is 6	

5.2.17 Class ToolkitRegistrySystem

5.2.17.1 Method getEntry

Test Area Reference: Api_2_Trs_Gety.

5.2.17.1.1 Conformance requirement:

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

```
public static ToolkitRegistry getEntry()
                              throws ToolkitException
```

5.2.17.1.1.1 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.

5.2.17.1.1.2 Parameter errors

No requirements.

5.2.17.1.1.3 Context errors

- CRRC1: This method returns null if the Applet.register() has not yet been invoked.
- CRRC2: This method returns null if the server does not exist.
- CRRC3: This method returns null if the server returns null.
- CRRC4: ToolkitException with REGISTRY_ERROR reason code shall be thrown in any case of register error.

5.2.17.1.2 Test area files

Test Source: Test_Api_2_Trs_Gety.java.

Test Applet: Api_2_Trs_Gety_1.java.

Cap File: api_2_trs_gety.cap.

5.2.17.1.3 Test coverage

CRR number	Test case number
N1	2
N2	3
C1	1
C2	Not testable
C3	Not testable
C4	Not testable

5.2.17.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call before register() method** In the constructor, the applet instance calls the getEntry() method.	Returns null	
2	**Call after register() method** In the install method and after the call to register() method, call the getEntry() method.	A not null reference is returned. No exception shall be thrown	
3	**Check it returns the same entry** The applet calls the getEntry() method again, in the processToolkit() method.	Returns the same ToolkitRegistry object reference as for test case 2.	

5.2.18 Class ToolkitException

5.2.18.1 ToolkitException Constructor

Test Area Reference: Api_2_Tke_Coor.

5.2.18.1.1 Conformance requirement:

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

```
public ToolkitException(short reason)
```

5.2.18.1.1.1 Normal execution

- CRRN1: Construct a ToolkitException instance with the specified reason.

5.2.18.1.1.2 Parameter errors

No requirements.

5.2.18.1.1.3 Context errors

No requirements.

5.2.18.1.2 Test area files

Test Source: Test_Api_2_Tke_Coor.java.

Test Applet: Api_2_Tke_Coor_1.java.

Cap File: api_2_tke_coor.cap.

5.2.18.1.3 Test coverage

CRR number	Test case number
N1	1

5.2.18.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**reason = (short) 19**	ToolkitException.getReason() = (short)19	

5.2.18.2 Method throwIt

Test Area Reference: Api_2_Tke_Thit.

5.2.18.2.1 Conformance requirement

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

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

5.2.18.2.1.1 Normal execution

- CRRN1: Throws the JCRE instance of the ToolkitException class with the specified reason.
- CRRN2: extends javacard.framework.CardRuntimeException.

5.2.18.2.1.2 Parameter errors

No requirements.

5.2.18.2.1.3 Context errors

No requirements.

5.2.18.2.2 Test area files

Test Source: Test_Api_2_Tke_Thit.java.

Test Applet: Api_2_Tke_Thit_1.java.

Cap File: api_2_tke_thit.cap.

5.2.18.2.3 Test coverage

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

5.2.18.2.4 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 = 0xA55A	
4	**ToolkitException extends javacard.framework.CardRuntimeException**	Reason = 0	
5	**ToolkitException extends javacard.framework.CardRuntimeException**	Reason = 1	
6	**ToolkitException extends javacard.framework.CardRuntimeException**	Reason = 0xA55A	

5.2.18.3 Reason Codes

Test Area Reference: Api_1_Tke_Cons.

5.2.18.3.1 Conformance Requirement

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

5.2.18.3.1.1 Normal execution

- CRRN1: The Constants of the class ToolkitException shall all have the same name and value defined in the ETSI TS 102 241 [9].
- CRRN2: Constructs ToolkitException an Exception with the specified reason.

5.2.18.3.1.2 Parameter errors

No requirements.

5.2.18.3.1.3 Context errors

No requirements.

5.2.18.3.2 Test area files

None.

5.2.18.3.3 Test Coverage

CRR number	Test case number
N1 & N2	The constants in Java are resolved at compilation time, therefore a runtime test is not useful. No test of constants will be performed

5.2.18.3.4 Test Procedure

None.

5.3 Package uicc.access.fileadministration

5.3.1 Interface AdminFileView

5.3.1.1 Method createFile(ViewHandler viewHandler)

Test Area Reference: Api_4_Afv_Crtf.

5.3.1.1.1 Conformance requirement

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

```
public void createFile(ViewHandler viewHandler)
               throws java.lang.NullPointerException,
                      UICCException,
                      AdminException,
                      javacard.framework.TransactionException
```

5.3.1.1.1.1 Normal execution

- CRRN1: This method creates a new file under the current DF or ADF, as described in ETSI TS 102 222 [7].

5.3.1.1.1.2 Parameter errors

- CRRP1: If viewHandler is null, an instance of java.lang.NullPointerException shall be thrown.
- CRRP2: If the viewHandler parameter includes incorrect parameters, an instance of AdminException shall be thrown. The reason code shall be AdminException.INCORRECT_PARAMETERS.

5.3.1.1.1.3 Context errors

- CRRC1: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.
- CRRC2: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.
- CRRC3: If the file identifier of the EF being created already exists, an instance of AdminException shall be thrown. The reason code shall be AdminException.FILE_ALREADY_EXISTS.
- CRRC4: If the DF name already exists, an instance of AdminException shall be thrown. The reason code shall be AdminException.DF_NAME_ALREADY_EXISTS.
- CRRC5: If there is not enough memory, an instance of AdminException shall be thrown. The reason code shall be AdminException.NOT_ENOUGH_MEMORY_SPACE.
- CRRC6: If the access conditions are not satisfied, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.

5.3.1.1.2 Test area files

Test Source: Test_Api_4_Afv_ Crtf.java.

Test Applet: Api_4_Afv_ Crtf _1.java.

Cap File: api_4_Afv_ Crtf.cap.

5.3.1.1.3 Test coverage

CRR number	Test case number
N1	1, 2
P1	3
P2	4
C1	Not testable
C2	Not testable
C3	5
C4	6
C5	Not testable
C6	7

5.3.1.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Create an EF** 1- applet is triggered by sending an unrecognized envelope 2- get an AdminFileView AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR_ON_RESET) 3- select MF/DF_{TEST} 4- create transparent EF_{RFU0} (6F29) 5- select EF_{RFU0}, update binary 12 34 56 6- create Linear fixed EF_{RFU1} 7- select EF_{RFU1}, update record 01 8- create Cyclic EF_{RFU2} 9- select EF_{RFU2}, update record 01 record:1 value: 12 34 56 Applet finalizes 10- select MF/DF_{TEST}/EF_{RFU0}, read binary 11- select MF/DF_{TEST}/EF_{RFU1}, read record 1 12- select MF/DF_{TEST}/EF_{RFU2} ,read record 1	4- no exception shall be thrown	10- returns: 12 34 56 11- returns: 02 12- returns: 12 34 56
2	**Create a DF in ADF1** 1- applet is triggered by sending an unrecognized envelope 2- get an AdminFileView AdminFileViewBuilder.getTheAdminFileView(AID_ADF1,CLEAR_ON_RESET) 3- select ADF1/DF_{TEST} 4- create DF_{RFU1} (5F01) 5- select DF_{RFU1} 6- create EF_{RFU1} (6F2A) 7- select EF_{RFU1}, update binary 12 34 56 Applet finalizes 8- select ADF1/DF_{TEST}/EF_{RFU1} (6F2A), read binary 9- Reset		8- returns: 12 34 56
3	**Call createFile with a null viewHandler** 1- call createFile() with null.	1- java.lang.NullPointerException shall be thrown	
4	**Call createFile with incorrect parameters** 1- call createFile with incorrect parameters.	1- AdminException.INCORRECT_PARAMETERS shall be thrown	
5	**EF already exists** 1- Select MF/DF_{TEST} 2- Call createFile(EF_{TARU})	2- AdminException.FILE_ALREADY_EXISTS	

Id	Description	API Expectation	APDU Expectation
6	**DF already exists** 1- Call createFile(DF_{TEST})	1- AdminException.DF_NAME_ALREADY_EXISTS	
7	**Security status not satisfied** 1- Select MF/DF_{TEST}/DF_{ARR2} 2- Call createFile() to create some transparent file.	2- UICCException.SECURITY_STATUS_NOT_SATISFIED	

5.3.1.2 Method deleteFile(short fid)

Test Area Reference: Api_4_Afv_Dltf.

5.3.1.2.1 Conformance requirement

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

```
public void deleteFile(short fid)
                throws UICCException,
                       javacard.framework.TransactionException
```

5.3.1.2.1.1 Normal execution

- CRRN1: This method initiates the deletion of an EF immediately under the current DF, or a DF with its complete subtree, as described in ETSI TS 102 222 [7].

5.3.1.2.1.2 Parameter errors

Not applicable

5.3.1.2.1.3 Context errors

- CRRC1: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.
- CRRC2: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.
- CRRC3: If the file cannot be found in the current directory, an instance of UICCException. shall be thrown. The reason code shall be UICCException. FILE_NOT_FOUND.
- CRRC4: If the access conditions are not satisfied, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.
- CRRC5: If the operation would cause the commit capacity to be exceeded, an instance of javacard.framework.TransactionException shall be thrown.

5.3.1.2.2 Test area files

Test Source: Test_Api_4_Afv_ Dltf.java.

Test Applet: Api_4_Afv_ Dltf _1.java.

Cap File: api_4_Afv_ Dltf.cap.

5.3.1.2.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4
C1	Not testable
C2	Not testable
C3	5
C4	6
C5	Not testable

5.3.1.2.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	**Initialization** 1- applet is triggered by sending an unrecognized envelope 2- get an AdminFileView AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR_ON_RESET) 3- select MF/DF_{TEST} 4- create DF_{RFU1} 5- select DF_{RFU1} 6- create DF_{RFU2} , create EF_{RFU1}, create EF_{RFU2} 7- select DF_{RFU2} 8- create EF_{RFU1} 9- get an AdminFileView AdminFileViewBuilder.getTheAdminFileView(AID_ADF1,CLEAR_ON_RESET) 10- select MF/DF_{TEST} 11- create DF_{RFU1} 12- select DF_{RFU1} 13- create DF_{RFU2} , create EF_{RFU1}, create EF_{RFU2} 14- select DF_{RFU2} 15- create EF_{RFU1}		
1	**Delete EF** 1- Select MF/DF_{TEST}/DF_{RFU1} 2- call deleteFile(EF_{RFU1}) 3- Select MF/DF_{TEST}/DF_{RFU1}/EF_{RFU1}	2- no exception shall be thrown 3- UICCException.FILE_NOT_FOUND is thrown	
2	**Delete EF in ADF1** 1- Select ADF1/DF_{TEST}/DF_{RFU1} 2- call deleteFile(EF_{RFU1}) 3- Select ADF1/DF_{TEST}/DF_{RFU1}/EF_{RFU1}	2- no exception shall be thrown 3- UICCException.FILE_NOT_FOUND is thrown	
3	**Delete DF and its subtree** 1- Select MF/DF_{TEST} 2- call deleteFile(DF_{RFU1}) 3- Select MF/DF_{TEST}/DF_{RFU1}	2- no exception shall be thrown 3- UICCException.FILE_NOT_FOUND is thrown	
4	**Delete DF and its subtree in ADF1** 1- Select ADF1/DF_{TEST} 2- call deleteFile(DF_{RFU1}) 3- Select ADF1/DF_{TEST}/DF_{RFU1}	2- no exception shall be thrown 3- UICCException.FILE_NOT_FOUND is thrown	
5	**File not found** 1- Select MF/DF_{TEST} 2- call deleteFile(DF_{RFU1}) 3- Select ADF1/DF_{TEST} 4- call deleteFile(EF_{RFU1})	2- UICCException.FILE_NOT_FOUND 4- UICCException.FILE_NOT_FOUND	
6	**Security status not satisfied** 1- Select MF/DFTEST/DFARR2 2- call deleteFile(EF_{TAR2T})	2- UICCException.SECURITY_STATUS_NOT_SATISFIED	

5.3.1.3 Method resizeFile(ViewHandler viewHandler)

Test Area Reference: Api_4_Afv_Rszf.

5.3.1.3.1 Conformance requirement

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

```
public void resizeFile(ViewHandler viewHandler)
                throws java.lang.NullPointerException,
                       UICCException,
                       AdminException,
                       javacard.framework.TransactionException
```

5.3.1.3.1.1 Normal execution

- CRRN1: This method resizes a file under the current DF or ADF, as described in ETSI TS 102 222 [7].

5.3.1.3.1.2 Parameter errors

- CRRP1: If viewHandler is null, an instance of java.lang.NullPointerException shall be thrown.
- CRRP2: If the viewHandler parameter includes incorrect parameters, an instance of AdminException shall be thrown. The reason code shall be AdminException.INCORRECT_PARAMETERS.

5.3.1.3.1.3 Context errors

- CRRC1: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.
- CRRC2: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.
- CRRC3: If the file cannot be found in the current directory, an instance of UICCException. shall be thrown. The reason code shall be UICCException. FILE_NOT_FOUND.
- CRRC4: If the access conditions are not satisfied, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.
- CRRC5: If the operation would cause the commit capacity to be exceeded, an instance of javacard.framework.TransactionException shall be thrown.
- CRRC6: If there is not enough memory, an instance of AdminException shall be thrown. The reason code shall be AdminException.NOT_ENOUGH_MEMORY_SPACE.
- CRRC7: If the conditions of use are not satisfied, an instance of AdminException shall be thrown. The reason code shall be AdminException.CONDITIONS_OF_USE_NOT_SATISFIED.
- CRRC8: If the method resizeFile() is applied to a non compatible file, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND_INCOMPATIBLE.
- CRRC9: If the method resizeFile() is applied to invalidated data, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF_DATA_INVALIDATED.

5.3.1.3.2 Test area files

Test Source: Test_Api_4_Afv_ Rszf.java.

Test Applet: Api_4_Afv_ Rszf _1.java.

Cap File: api_4_Afv_ Rszf.cap.

5.3.1.3.3 Test coverage

CRR number	Test case number
N1	1, 2
P1	3
P2	4
C1	Not testable
C2	Not testable
C3	5
C4	6
C5	Not testable
C6	Not testable
C7	Not testable
C8	7
C9	8

5.3.1.3.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Resize a Transparent EF** 1- Trigger the applet with an unrecognized envelope 2- Select MF/DF_{TEST} 3- Call ResizeFile() on EF_{TARU} new size: 10 bytes Applet finalizes 4- Select MF/DF_{TEST}/EF_{TARU} and check size in the returned FCP template. 5- Restore EF_{TARU}		4- should return a size of 10 bytes
2	**Resize a Linear Fixed EF** 1- Trigger the applet with an unrecognized envelope 2- Select MF/DF_{TEST} 3- Call ResizeFile() on EFLARU add 2 records. Applet finalizes 4- Select MF/DF_{TEST}/EF_{LARU} and check size in the returned FCP template. 5- Restore EF_{LARU}		4- should return a size of 16 bytes
3	**Call resizeFile with a null viewHandler** 1- Call resizeFile with null.	1- java.lang.NullPointerException shall be thrown	
4	**Call createFile with incorrect parameters** 1- Call createFile with incorrect parameters.	1- AdminExceptiogn.INCORRECT_PARAMETERS shall be thrown	
5	**File not found** 1- Select MF/DF_{TEST} 2- Call resizeFile(DF_{RFU1}) 3- Select ADF1/DF_{TEST} 4- Call resizeFile(EF_{RFU1})	2- UICCException.FILE_NOT_FOUND shall be thrown 4- UICCException.FILE_NOT_FOUND shall be thrown	
6	**Security status not satisfied** 1- Select MF/DFTEST/DFARR2 2- Call resizeFile(EF_{TAR2T})	2- UICCException.SECURITY_STATUS_NOT_SATISFIED shall be thrown	

Id	Description	API Expectation	APDU Expectation
7	**Command incompatible** 1- Select MF/DF_{TEST}, call resizeFile(EF_{CARU})	1- UICCException.COMMAND_INCOMPATIBLE shall be thrown	
8	**Invalidated data** 1- Select MF/DF_{TEST} 2- Invalidate EF_{TARU} 3- Call resizeFile(EF_{TARU}) 4- Validate EF_{TARU}	3- UICCException.REF_DATA_INVALIDATED shall be thrown.	

5.3.1.4 Method select (byte sfi)

Test Area Reference: Api_4_Afv_Slctb.

5.3.1.4.1 Conformance requirement

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

```
public void select(byte sfi)
            throws UICCException
```

5.3.1.4.1.1 Normal execution

- CRRN1: Selects a file by its Short File Identifier in the current directory of the FileView.
- CRRN2: Allows to update the current file without handling the Select Response.
- CRRN3: The current EF it self can be selected.
- CRRN4: The file context associated with the FileView object is changed after successful execution.

5.3.1.4.1.2 Parameter errors

- CRRP1: If the file which sfi matches is not in the current directory or no file matches the sfi, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE_NOT_FOUND.

5.3.1.4.1.3 Context errors

- CRRC1: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM
- CRRC2: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.3.1.4.2 Test area files

Test Source: Test_Api_4_Afv_ Slctb.java.

Test Applet: Api_4_Afv_ Slctb _1.java.

Cap File: api_4_Afv_ slctb.cap.

5.3.1.4.3 Test coverage

CRR number	Test case number
N1	1, 2, 4
N2	Not testable
N3	3
N4	5
P1	4
C1	Not testable
C2	Not testable

5.3.1.4.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Selection possibilities, UICC file system** 1- get an AdminFileView AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR_ON_RESET) 2- select DF_{TEST}, fid=1111 3- select with sfi EF_{TNR}, sfi=0x01 4- select with sfi EF_{TNU}, sfi=0x02 5- select with sfi EF_{CNU}, sfi=0x05	2- no exception shall be thrown 3- no exception shall be thrown 4- no exception shall be thrown 5- no exception shall be thrown	
2	**Selection possibilities, ADF1** 1- get an AdminFileView AdminFileViewBuilder.getTheAdminFileView(AID_ADF1,CLEAR_ON_RESET) 2- select DF_{TEST}, fid=1111 3- select with sfi EF_{TNR}, sfi=0x01 4- select with sfi EF_{TNU}, sfi=0x02 5- select with sfi EF_{CNU}, sfi=0x05	2- no exception shall be thrown 3- no exception shall be thrown 4- no exception shall be thrown 5- no exception shall be thrown	
3	**Current EF itself can be selected** 1- get an AdminFileView AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR_ON_RESET) 2- select DF_{TEST}, fid=1111 3- select with sfi EF_{TNR}, sfi=0x01 4- select with sfi EF_{TNR}, sfi=0x01	4- no exception shall be thrown	
4	**FILE_NOT_FOUND** 1- try to select a file with sfi=0x55	1- shall throw an uicc.access.UICCException with reason code FILE_NOT_FOUND	
5	**File context changed** 1- select EF_{TARU}, sfi=0x03 read 3 first bytes 2- select EF_{TNU}, sfi=0x02 read file content	1- file content should be {0xFF,0xFF,0xFF} 2- file content should be {0x55,0x55,0x55}	

5.3.1.5 Method select(short fid, byte[] fcp, short fcpOffset, short fcpLength)

Test Area Reference: Api_4_Afv_Slctb_bss.

5.3.1.5.1 Conformance requirement

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

```
public short select(short fid,
                    byte[] fcp,
                    short fcpOffset,
                    short fcpLength)
             throws java.lang.NullPointerException,
                    java.lang.ArrayIndexOutOfBoundsException,
                    UICCException
```

5.3.1.5.1.1 Normal execution

- CRRN1: Selects a file of the UICC file system or of an ADF file system as defined in ETSI TS 102 221 [5].
- CRRN2: The method returns the FCP information in a form of a TLV structure as specified in ETSI TS 102 221 [5].
- CRRN3: If the fcpLength is greater than the length of the response, the whole response is copied into the fcp buffer and the length of the response is returned by the method.
- CRRN4: If the fcpLength is smaller than the length of the response, the first part of the response is copied into the fcp buffer and the fcpLength is returned by the method.
- CRRN5: After selecting a ADF/MF/DF no EF is selected.
- CRRN6: After selecting a linear fixed EF no record is selected.
- CRRN7: After selecting a cyclic EF the last updated record is the first record.
- CRRN8: Any file can be selected by FID which is an immediate child of the current directory.
- CRRN9: Any DF can be selected by FID which is an immediate child of the parent of the current DF.
- CRRN10: The parent of the current directory can be selected by the FID.
- CRRN11: The ADF of the current active application can be selected by the FID.
- CRRN12: The ADF/MF can always be selected.
- CRRN13: The file context associated with the FileView object is changed after successful execution.
- CRRN14: The current file context of any other applets shall not be changed. This will be tested during the testing of the framework.

5.3.1.5.1.2 Parameter errors

- CRRP1: If the array fcp is null, an instance of NullPointerException shall be thrown.
- CRRP2: If fcpOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP3: If fcpLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP4: If fcpOffset plus fcpLength is greater than the length of the array fcp.length an instance of ArrayIndexOutOfBoundsException shall be thrown.

5.3.1.5.1.3 Context errors

- CRRC1: If the file with a File-ID which matches fid could not be found according to the selection rules, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE_NOT_FOUND.
- CRRC2: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.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 UICCException.INTERNAL_ERROR.

5.3.1.5.2 Test area files

Test Source: Test_Api_4_Afv_Slctb_bss.java.

Test Applet: Api_4_Afv_Slctb_bss_1.java.

Cap File: Api_4_Afv_slctb_bss.cap.

5.3.1.5.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5, 6, 7, 8, 14, 18, 20, 21
N2	1, 2, 3, 4, 5, 6, 7, 8, 14, 18, 20, 21
N3	1
N4	2, 3, 4 ,5, 6, 7, 8
N5	15,19
N6	17
N7	18
N8	14
N9	14
N10	14
N11	19, 20
N12	20
N13	20
N14	1, 2, 3, 4, 5, 6, 7, 8, 14, 18, 20
P1	9
P2	10
P3	11
P4	12, 13
C1	16
C2	Not testable
C3	Not testable

5.3.1.5.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	**Get an AdminFileView object, UICC file system** get an AdminFileView FileView.getTheUICCAdminFileView(CLEAR_ON_RESET)		
1	**Select EFTARU in MF (Transparent EF)** Select DF_{TEST} select EF_{TARU}, fid=6F03 byte[] fcp = new byte[132] fcpOffset = 0 fcpLength = 127	Shall return at least 19. fcp[] shall contain following TLVs: 1. 82 02 41 21 //file descriptor 2. 83 02 2F E2 //file id 3. 8A 01 05 //life cycle status 4. 80 02 00 0A // file size	
2	**Select EFTARU in MF (Transparent EF)** select EF_{TARU}, fid=6F03 fcpOffset = 0 fcpLength = 7 select()	Shall return 7. fcp[] shall contain the first 7 bytes of the FCP structure and contain following TLV. 1. 82 02 41 21 //file descriptor	
3	**Select DFTEST in MF** fid = DF_{TEST}, fid=1111 fcpOffset = 0 fcpLength = 127 select()	Shall return at least 17. fcp[] shall contain following TLVs 1. 82 02 78 21 //file descriptor 2. 83 02 11 11 //file id 3. 8A 01 05 //life cycle status	
4	**Select EFCARU in DFTEST (Cyclic EF)** select EF_{CARU}, fid=6f09 fcpOffset = 0 fcpLength = 11 select()	Shall return: 11 fcp[] shall contain following TLV: 82 05 46 21 00 03 02	
5	**Select ADF1** select ADF fid=7FFF fcp[0:5]=0x00 fcpOffset=5 fcpLength=127 select	Shall return: at least 27 The first 5 bytes of fcp[] shall be 0x00 and contains following TLVs: 1. 82 02 78 21 //file descriptor 2. 84 10 A0 00 00 00 09 00 05 FF FF FF FF 89 60 00 00 00 //DF Name 3. 8A 01 05 //life cycle	

Id	Description	API Expectation	APDU Expectation
6	**Select MF** select MF, fid= 3F00 fcpOffset = 0 fcpLength = 11 select()	Shall return: 11 fcp[] shall contain following TLVs: 1. 82 02 38/78 21 //file descriptor 2. 83 02 3F 00 //file ID	
7	**Select DFTELECOM in MF** select $DF_{TELECOM}$, fid=7F10 fcp[0] = fcp[1] = 0x05 fcpOffset = 2 fcpLength = 13 select()	Shall return 13. The first 2 bytes of fcp[] shall be 0x05 and fcp[] shall contain following TLVs 1. 82 02 38/78 21 //file descriptor 2. 83 02 7F 10 //file id	
8	**Select EFLARU in DFTELECOM (Linear FixedEF)** select EF_{LARU}, fid = 6F0C fcpOffset = 0 fcpLength = 14	Shall return 14. fcp[] shall contain following TLVs: 1. 82 05 42 21 00 04 02 2. 83 02 6F 0C	
9	**fcp is null** select EF_{LARU}, fid = 6F0C byte[] nullBuffer = null fcpOffset = 0 fcpLength = 15	Shall throw java.lang.NullPointerException	
10	**fcpOffset < 0** select EF_{LARU}, fid = 6F0C fcpOffset = -1 fcpLength = 15	Shall throw java.lang.ArrayIndexOutOfBoundsException	
11	**fcpLength < 0** select EF_{LARU}, fid = 6F0C fcpOffset = 0 fcpLength = -1	Shall throw java.lang.ArrayIndexOutOfBoundsException	
12	**fcpOffset + fcpLength > fcp.length** select EF_{LARU}, fid = 6F0C fcpOffset = 115 fcpLength = 18	Shall throw java.lang.ArrayIndexOutOfBoundsException	
13	**fcpOffset + fcpLength > fcp.length** select EF_{LARU}, fid = 6F0C fcpOffset = fcp.length+1 fcpLength = 0	Shall throw java.lang.ArrayIndexOutOfBoundsException	
14	**Selection possibilities** 0- select MF, fid=3F00 1- select EF_{UICC}, fid=2FF0 2- select DF_{TEST}, fid=1111 3- select EF_{CNU}, fid=6F05 4- select EF_{TAAA}, fid=6F16 5- select DF_{SUB_TEST}, fid=2211 6- select DF_{TEST}, fid=1111 7- select EF_{TAAA}, fid=6F16 8- select DF_{TEST}, fid=1111 9- select MF, fid=3F00 10- select DF_{TEST}, fid=1111 11- select EF_{TAAA}, fid=6F16 12- select MF, fid=3F00	No exception shall be thrown.	
15	**EF not selected after MF/DF selection** 1- select MF, fid = 3F00 select EF_{ICCID}, fid = 2FE2 2 - select MF fid = 3F00 select() readBinary()	2 - Shall throw uicc.access.UICCException with reason code NO_EF_SELECTED.	

Id	Description	API Expectation	APDU Expectation
16	**No selection of non-reachable file** 1 - select MF, fid = 3F00 2 - select EF_{CARU}, fid= 0x6F09	2 - Shall throw uicc.access.UICCException with reason code FILE_NOT_FOUND.	
17	**No record is selected after selecting linear fixed EF** 1- select MF, fid = 3F00 2- select DF_{TEST}, fid=1111 3- select EF_{LARU}, fid=6F0C 4 - recNumber = 0 mode = REC_ACC_MODE_ CURRENT readRecord()	4 - Shall throw uicc.access.UICC Exception with reason code RECORD_NOT_FOUND.	
18	**Record pointer in selected cyclic EF** 1- select MF, fid = 3F00 2- select DF_{TEST}, fid=1111 3- select EF_{CARU}, fid=6F09 4- byte[] data1 = { 1,2,3 } mode = REC_ACC_MODE_PREVIOUS updateRecord(data1) 5- select EF_{CARU} fid = 6F09 select() mode = REC_ACC_MODE_PREVIOUS readRecord() readRecord(data2) compare data1 to data2 6- restore original data of EF_{CARU}	5 - The contents of data1 and data2 shall be identical.	
19	**EF not selected after ADF/DF selection** 1- get an AdminFileView AdminFileViewBuilder.getTheAdminFileView (AID_ADF1,CLEAR_ON_RESET) 2- select ADF, fid = 7FFF select EF_{UICC}, fid = 2FF0 3 - select ADF fid = 7FFF select() readBinary()	3 - Shall throw uicc.access.UICCException with reason code NO_EF_SELECTED.	
20	**Reselection** 1- Using the ADF FileView select ADF, fid=7FFF select ADF, fid=7FFF 2- Using the UICC FileView select MF, fid=3F00 select MF, fid=3F00 3- select DF_{TEST}, fid=1111 select DF_{TEST}, fid=1111 5- select EF_{TAAA}, fid=6F16 select EF_{TAAA}, fid=6F16	No exceptions shall be thrown	
21	**Security attributes** 1- Using the ADF FileView select ADF, fid=7FFF select DF_{TEST}, fid=1111 select EF_{LARR1}, fid=6FA1 2- Using the UICC FileView select MF, fid=3F00 select DF_{TEST}, fid=1111 select EF_{TARR3}, fid=6FB3	1- fcp[] shall contain the following TLV 8B 03 AC 00 01 or 8B 06 AC 00 00 01 01 01 2- fcp[] shall contain the following TLV 8B 03 AC 00 03 or 8B 06 AC 00 00 03 01 03	

5.3.1.6 Method select (short fid)

Test Area Reference: Api_4_Afv_Slcts.

5.3.1.6.1 Conformance requirement

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

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

5.3.1.6.1.1 Normal execution

- CRRN1: Selects a file of the UICC file system or of an ADF file system by file identifier.
- CRRN2: Allows to update the current file without handling the Select Response.
- CRRN3: After selecting a ADF/MF/DF no EF is selected.
- CRRN4: After selecting a linear fixed EF no record is selected.
- CRRN5: Any file can be selected by FID which is an immediate child of the current directory.
- CRRN6: Any DF can be selected by FID which is an immediate child of the parent of the current DF.
- CRRN7: The parent of the current directory can be selected by the FID.
- CRRN8: The ADF of the current active application can be selected by the FID.
- CRRN9: The ADF/MF/EF can always be self selected.
- CRRN10: The file context associated with the FileView object is changed after successful execution.

5.3.1.6.1.2 Parameter errors

No requirements.

5.3.1.6.1.3 Context errors

- CRRC1: If the file with a File Identifier which matches fid could not be found according to the selection rule listed in CCRN3, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE_NOT_FOUND.
- CRRC2: If the file with a File Identifier which matches fid could not be found according to the selection rule listed in CCRN4, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE_NOT_FOUND.
- CRRC3: If the file with a File Identifier which matches fid could not be found according to the selection rule listed in CCRN5, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE_NOT_FOUND.
- CRRC4: If the file with a File Identifier which matches fid could not be found according to the selection rule listed in CCRN6, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE_NOT_FOUND.
- CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.
- CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.3.1.6.2 Test area files

Test Source: Test_Api_4_Afv_Slcts.java.

Test Applet: Api_4_Afv_Slcts_1.java.

Cap File: api_4_Afv_slcts.cap.

5.3.1.6.3 Test coverage

CRR number	Test case number
N1	1,2
N2	5
N3	5
N4	6
N5	1, 2
N6	1, 2
N7	1, 2
N8	1,2
N9	4
N10	Tested in Api_1_Cont, test case 1 and 2
C1	3
C2	3
C3	3
C4	3
C5	Not testable
C6	Not testable

5.3.1.6.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	**Get an AdminFileView object, UICC file system** 1- get an AdminFileView AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR_ON_RESET)		
1	**Selection possibilities** 1- select EF_{UICC}, fid=2FF0 2- select DF_{TEST}, fid=1111 3- select EF_{CNU}, fid=6F05 4- select EF_{TAAA}, fid=6F16 5- select DF_{SUB_TEST}, fid=2211 6- select DF_{TEST}, fid=1111 7- select EF_{TAAA}, fid=6F16 8- select DF_{TEST}, fid=1111 9- select MF, fid=3F00 10- select DF_{TEST}, fid=1111 11- select EF_{TAAA}, fid=6F16 12- select MF, fid=3F00	No exception shall be thrown	
2	**Selection possibilities, ADF1** 1- get an AdminFileView AdminFileViewBuilder.getTheAdminFileView(AID_ADF1,CLEAR_ON_RESET) 2- select EF_{UICC}, fid=2FF0 3- select DF_{TEST}, fid=1111 4- select EF_{CNU}, fid=6F05 5- select EF_{TAAA}, fid=6F16 6- select DF_{SUB_TEST}, fid=2211 7- select DF_{TEST}, fid=1111 8- select EF_{TAAA}, fid=6F16 9- select DF_{TEST}, fid=1111	No exception shall be thrown	

Id	Description	API Expectation	APDU Expectation
3	**No selection of unreachable file** 1- get an AdminFileView AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR_ON_RESET) 2- select EF_{CNU}, fid=6F05 3- select DF_{TEST}, fid=1111 4- select EF_{TAA}, fid=2222 5- select EF_{CNU}, fid=6F05 6- select DF_{SUB_TEST}, fid=2211 7- select EF_{TAA}, fid=2222 8- select $DF_{TELECOM}$, fid=7F10	2- A UICCException.FILE_NOT_FOUND shall be thrown. 3- No exception shall be thrown 4- A UICCException.FILE_NOT_FOUND shall be thrown. 5- No exception shall be thrown 6- No exception shall be thrown 7- No exception shall be thrown 8- A UICCException.FILE_NOT_FOUND shall be thrown.	
4	**Self selection** 1- select MF, fid=3F00 2- select MF, fid=3F00 3- select DF_{TEST}, fid=1111 4- select DF_{TEST}, fid=1111 5- select EF_{TAAA}, fid=6F16 6- select EF_{TAAA}, fid=6F16 7- get an AdminFileView AdminFileViewBuilder.getTheAdminFileView(AID_ADF1,CLEAR_ON_RESET) 8- select ADF, fid=7FFF 9- select ADF, fid=7FFF	2- No exception shall be thrown 4- No exception shall be thrown 6- No exception shall be thrown 8- No exception shall be thrown 9- No exception shall be thrown	
5	**EF not selected after MF/DF selection** 1- select MF, fid=3F00 2- updateRecord() 3- select DF_{TEST}, fid=1111 4- updateRecord()	2- A UICCException.NO_EF_SELECTED shall be thrown 4- A UICCException.NO_EF_SELECTED shall be thrown	
6	**No record is selected after selecting linear fixed EF** 1- select MF, fid = 3F00 2- select DF_{TEST}, 3- select EF_{LARU}, 4 - recNumber = 0 mode = REC_ACC_MODE_ CURRENT readRecord() 5- select EF_{CARU}, 6 - recNumber = 0 mode = REC_ACC_MODE_ CURRENT readRecord()	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - No exception shall be thrown. 4 - Shall throw uicc.access.UICCException with reason code RECORD_NOT_FOUND. 5 - No exception shall be thrown. 6 - Shall throw uicc.access.UICCException with reason code RECORD_NOT_FOUND.	

5.3.1.7 Method status

Test Area Reference: Api_4_Afv_Stat.

5.3.1.7.1 Conformance requirement

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

```
public short status(byte[] fcp,
                    short fcpOffset,
                    short fcpLength)
             throws java.lang.NullPointerException,
                    java.lang.ArrayIndexOutOfBoundsException,
                    UICCException
```

5.3.1.7.1.1 Normal execution

- CRRN1: The method returns the File Control Parameter of the current selected DF/MF or ADF as defined in TS 102 221 [5].
- CRRN2: If the `fcpLength` is greater than the length of the response, the whole response is copied into the `fcp` buffer and the length of the response is returned by the method.
- CRRN3: f the `fcplength` is smaller than the length of the response, the first part of the response is copied into the `fcp` buffer and the `fcpLength` is returned by the method.

5.3.1.7.1.2 Parameter errors

- CRRP1: If the array fcp is null, an instance of NullPointerException shall be thrown.
- CRRP2: If fcpOffset is negative, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP3: If fcpLength is negative, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP4: If fcpOffset+fcpLength is greater than fcp.length an ArrayIndexOutOfBoundsException shall be thrown.

5.3.1.7.1.3 Context errors

- CRRC1: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.
- CRRC2: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.3.1.7.2 Test area files

Test Source: Test_Api_4_Afv_Stat.java.

Test Applet: Api_4_Afv_Stat_1.java.

Cap File: Api_4_Afv_Stat.cap.

5.3.1.7.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5, 11
N2	2, 3
N3	1, 4
P1	6
P2	7
P3	8
P4	9, 10
C1	Not testable
C2	Not testable

5.3.1.7.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Status of MF** 1- Get an AdminFileView object, UICC file get an AdminFileView FileView.getTheUICCAdminFileView(CLEAR_ON_RESET) 2- select MF byte[] fcp = new byte[127] fcp[0:2]= 0xCC fcpOffset = 3 fcpLength = 11 status()	2- Shall return 11. The first 3 bytes of fcp[] shall contain 0xCC. fcp[] shall contain following TLVs: 1. 82 02 38/78 21 //file descriptor 2. 83 02 3F 00 //file ID	
2	**Status after select EF_{TARU} in MF** 1 - select DF_{TEST} select EF_{TARU}, fid = 6F03 fcpOffset = 0 fcpLength = 127 select() status()	1- Shall return at least 19. fcp[] shall contain following TLVs: 1. 82 02 38/78 21 //file descriptor 2. 83 02 11 11 //file ID	
3	**Status of $DF_{TELECOM}$** 1 - fid = 7F10 fcpOffset = 0 fcpLength = 127 status()	1 - Shall return at least 17 and the entire structure of the file control parameters. The file identifier shall be contain the fid of $DF_{TELECOM}$.	
4	**Status $DF_{TELECOM}$** Select $DF_{TELECOM}$, fid=7F10 fcpOffset = 0 fcpLength = 11 status()	Shall return 11. fcp shall contain the first 11 bytes of the FCP structure starting at index 0. fcp[] shall contain following TLVs: 1. 82 02 38/78 21 //file descriptor 2. 83 02 7F 10 //file id	
5	**Status ADF1** Select ADF, fid=7FFF fcpOffset = 0 fcpLength = 127	Shall return at least 27 fcp[] shall contain the entire FCP structure fcp[] shall contain following TLVs: 1. 82 02 78 21 //file descriptor 2. 84 10 A0 00 00 00 09 00 05 FF FF FF FF 89 60 00 00 00 //DF Name 3. 8A 01 05 //life cycle	
6	**fcp is null** byte[] nullBuffer = null fcpOffset = 0 fcpLength = 34 status()	Shall throw java.lang.NullPointerException.	
7	**fcpOffset < 0** fcpOffset = -1 fcpLength = 34 status()	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
8	**fcpLength < 0** fcpOffset = 0 fcpLength = -1 status()	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
9	**fcpOffset + fcpLength > fcp.length** fcpOffset = fcp.length-1 fcpLength = 15 status()	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
10	**fcpOffset + fcpLength > fcp.length** fcpOffset = fcp.length+1 fcpLength = 0 status()	Shall throw java.lang.ArrayIndexOutOfBoundsException.	

Id	Description	API Expectation	APDU Expectation
11	**Security attributes** 1- Using the ADF FileView select ADF, fid=7FFF select DF_{TEST}, select DF_{ARR2}, 2- Using the UICC FileView select MF, fid=3F00 select DF_{TEST}, select DF_{ARR4},	1- fcp[] shall contain the following TLV 8B 03 AC 00 02 or 8B 06 AC 00 00 02 01 02 2- fcp[] shall contain the following TLV 8B 03 AC 00 04 or 8B 06 AC 00 00 04 01 04	

5.3.1.8 Method readBinary

Test Area Reference: Api_4_Afv_Redb.

5.3.1.8.1 Conformance requirement

The method with 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,
                        UICCException
```

5.3.1.8.1.1 Normal execution

- CRRN1:. Reads the data bytes of the current transparent EF, as defined in ETSI TS 102 221 [5].
- CRRN2: The sum of respOffset plus respLength is returned. and the data bytes of the currently selected transparent file are returned in resp.

5.3.1.8.1.2 Parameter errors

- CRRP1: If fileOffset is negative, an instance of UICCException.OUT_OF_FILE_BOUNDARIES shall be thrown.
- CRRP2: If respOffset is negative, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP3: If respLength is negative, an instace of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP4: If the array resp is null, an instance of NullPointerException shall be thrown.
- CRRP5: If respOffset plus respLength is greater than the length of the array resp.length, an instance of ArrayIndexOutOfBoundsException shall be thrown and no read is performed.
- CRRP6: If fileOffset plus respLength exceeds the length of the file, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT_OF_FILE_BOUNDARIES.

5.3.1.8.1.3 Context errors

- CRRC1: If the method call causes an error to occur that is not expected and thus not handled, an instace of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.
- CRRC2: If the currently selected EF is not transparent, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND_INCOMPATIBLE.
- CRRC3: If the calling applet does not fulfil the access condition, READ, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.

- CRRC4: If the currently selected EF is deactivated and the file status of the EF does not allow for the reading of an deactivated file, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF_DAT_INVALIDATED.

- CRRC5:If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.

5.3.1.8.2 Test area files

Test Source: Test_Api_4_Afv_Redb.java.

Test Applet: Api_4_Afv_Redb_1.java.

Cap File: Api_4_Afv_redb.cap.

5.3.1.8.3 Test coverage

CRR number	Test case number
N1	1, 2
N2	1, 2
P1	3
P2	6
P3	7
P4	5
P5	8
P6	4
C1	Not testable
C2	9
C3	10
C4	11
C5	12

5.3.1.8.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Read from EF_{TARU}** 1- select DF_{TEST}, fid=1111 select EF_{TARU}, fid=6F03 2- fileOffset = 0 resp.length = 260 resp[0:259] = 0x55 respOffset = 10 respLength = 250 readBinary()	2 - shall return 20. resp shall contain the contents of EF_{TARU} starting at index 10. <Description of resp: 55 55 55 55 55 55 55 55 55 55 FF FF FF FF FF FF .. .. FF>	
2	**Read from EF_{TARU}** fileOffset = 0x80 resp.length = 260 resp[0:259] = 0x55 respOffset = 5 respLength = 0x80 readBinary()	shall return 15 resp shall contain the last 5 bytes of EF_{TARU} 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 FF FF … FF >	
3	**FileOffset is negative** fileOffset = -1 respOffset = 0 respLength = 10 readBinary()	Shall throw uicc.access.UICC Exception with reason code OUT_OF_FILE_BOUNDARIES.	
4	**FileOffset + respLength > EF length** fileOffset = 259 respOffset = 0 respLength = 2 readBinary()	Shall throw uicc.access.UICC Exception with reason code OUT_OF_FILE_BOUNDARIES.	

Id	Description	API Expectation	APDU Expectation
5	**resp[] is null** fileOffset = 0 resp = null 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 resp.length = 20 respOffset = 10 respLength = 11 readBinary()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
9	**EF is not Transparent** 1- select EF_{LARU}, fid=6F0C 2- fileOffset = 0 respOffset = 0 respLength = 1 readBinary()	2 - Shall throw uicc.access.UICC Exception with reason code COMMAND_INCOMPATIBLE.	
10	**Access condition not fulfilled** 1- select EF_{TRAC}, fid=6F0E 2- fileOffset = 0 respOffset = 0 respLength = 1 readBinary()	2- Shall throw uicc.access.UICC Exception with reason code SECURITY_STATUS_NOT_SATISFIED.	
11	**EF is deactivated** 1 - select EF_{TARU}, fid=6F03 2 - deactivateFile() 3 - readBinary() 4 - activateFile())	3 - Shall throw uicc.access.UICCException with reason code REF_DATA_INVALIDATED.	
12	**No EF selected** 1- select DF_{TEST} fid=1111 2 readBinary()	2 - Shall throw uicc.access.UICCException with reason code NO_EF_SELECTED.	

5.3.1.9 Method updateBinary

Test Area Reference: Api_4_Afv_Updb.

5.3.1.9.1 Conformance requirement

The method with 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,
                         UICCException
```

5.3.1.9.1.1 Normal execution

- CRRN1: Updated the data bytes of the current selected transparent EF.

5.3.1.9.1.2 Parameter errors

- CRRP1: If recOffset is less than 0, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT_OF_FILE_BOUNDARIES.
- CRRP2: : If fileOffset plus dataLength exceeds the length of the file, an instance of UICCException shall be thrown. The reason code shall be UICCException.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.

5.3.1.9.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the currently selected EF is not transparent, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND_INCOMPATIBLE.
- CRRC3: If the calling applet does not fulfil the access condition, UPDATE, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.
- CRRC4: If the currently selected EF is deactivated and the file status of the EF does not allow for updating of a deactivated file, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF_DATA_INVALIDATED.
- CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.
- CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.3.1.9.2 Test area files

Test Source: Test_Api_4_Afv_Updb.java.

Test Applet: Api_4_Afv_Updb _1.java.

Cap File: Api_4_Afv_updb.cap.

5.3.1.9.3 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,	Not Testable
C6	Not Testable

5.3.1.9.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	**Get the UICC FileView** AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR_ON_RESET)		0
1	**No EF selected** fileOffset = 0 byte[] data = new byte[20] data[0] = '55' dataOffset = 0 dataLength = 10 updateBinary()	Shall throw uicc.access.UICC Exception with reason code NO_EF_SELECTED.	1
2	**Update Transparent EF** 1- select DF_{TEST}, fid = 1111 2- select EF_{TARU}, fid = 6F03 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	**fileOffset = 254** 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 uicc.access.UICCException with reason code OUT_OF_FILE_BOUNDARIES.	
5	**fileOffset + dataLength > EF length** fileOffset = 259 dataOffset = 0 dataLength = 2 updateBinary()	Shall throw uicc.access.UICC Exception 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.	

Id	Description	API Expectation	APDU Expectation
10	**EF is not Transparent** 1- select DF_{TEST}, fid = 1111 2- select DF_{LARU}, fid = 6F0C 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 uicc.access.UICC Exception with reason code COMMAND_INCOMPATIBLE.	
11	**Access condition not fulfilled** 1- select DF_{TEST}, fid = 1111 2- select EF_{TNU}, fid = 6F02 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 uicc.access.UICCException with reason code SECURITY_STATUS_NOT_SATISFIED.	
12	**EF is deactivated** 1- select EF_{TNR}, fid = 6F01 deactiveFile() 2- fileOffset = 0 data[0] = '55' dataOffset = 0 dataLength = 1 updateBinary() 3- activateFile()	1- No exception shall be thrown. 2- Shall throw uicc.access.UICCException with reason code REF_DATA_INVALIDATED 3- No exception shall be thrown.	

5.3.1.10 Method readRecord

Test Area Reference: Api_4_Afv_Redr.

5.3.1.10.1 Conformance requirement

The method with 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,
                        UICCException
```

5.3.1.10.1.1 Normal execution

- CRRN1: Reads a record or a part of record of a current linear fixed or cyclic EF into byte array resp and the sum of respOffset plus respLength is returned.
- CRRN2: If the access mode is REC_ACC_MODE_CURRENT the current record will be read and the current record pointer shall not be changed.
- CRRN3: If the access mode is REC_ACC_MODE_ABSOLUTE the record addressed by recNumber will be read and the current record pointer shall not be changed.
- CRRN4: If the access mode is REC_ACC_MODE_NEXT the next record relative to the current selected record will be selected and read. The record pointer will be incremented.
- CRRN5: If the access mode is REC_ACC_MODE_NEXT and no current record is selected, the first record will be selected and read. The record pointer will be incremented.
- CRRN6: If the access mode is REC_ACC_MODE NEXT and the current record pointer, of a cyclic EF, is set to the last record, the record pointer is set to the first record and the record is read.
- CRRN7: If the access mode is REC_ACC_MODE_PREVIOUS the previous record relative to the current selected record will be selected and read.

- CRRN8: If the access mode is REC_ACC_MODE PREVIOUS and no current record is selected, the last record will be selected and read.
- CRRN9:If the access mode is REC_ACC_MODE_PREVIOUS and the current record pointer of a cyclic EF is set to the first record, the record pointer is set to the last record in this EF and this record shall be read.
- CRRN10: The current record pointer of any other applet shall not be changed.

5.3.1.10.1.2 Parameter errors

- CRRP1: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_ABSOLUTE and recNumber is less than 0 or greater than records available, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD_NOT_FOUND.
- CRRP2: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_CURRENT, recNumber is 0 and there is no current record selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD_NOT_FOUND.
- 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 UICCException shall be thrown. The reason code shall be UICCException.RECORD_ NOT_FOUND.
- 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 UICCException shall be thrown. The reason code shall be UICCException.RECORD_ NOT_FOUND.
- CRRP5: If the specified offset into the selected record recOffset is less than 0, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT_OF_RECORD_BOUNDARIES.
- CRRP6: If recOffset plus respLength is greater than the record length, an instance of UICCException shall be thrown. The reason code shall be UICC Exception.OUT_OF_RECORD_BOUNDARIES.
- CRRP7: If the access mode is not between 2 and 4 (2 = REC_ACC_MODE_NEXT, etc.), an instance of UICCException shall be thrown. The reason code shall be UICC Exception.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.

5.3.1.10.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the currently selected EF is neither linear fixed nor cyclic, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND_INCOMPATIBLE.
- CRRC3: If the calling applet does not fulfil the access condition, READ, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.
- 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 UICCException shall be thrown. The reason code shall be UICCException.REF_DATA_INVALIDATED.
- CRRC5: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.3.1.10.2 Test area files

Test Source: Test_Api_4_Afv_Redr.java.

Test Applet: Api_4_Afv_Rredr_1.java.

Cap File: Api_4_Afv_redr.cap.

5.3.1.10.3 Test coverage

CRR number	Test case number
N1	2, 3, 4, 5,6, 8, 9, 10, 11, 12, 13
N2	3, 9
N3	2, 8
N4	4, 5, 10, 11
N5	4, 11
N6	11
N7	6, 7, 12, 13
N8	6, 13
N9	12
N10	
P1	14
P2	15
P3	5
P4	7
P5	16
P6	17
P7	18
P8	19
P9	20
P10	21
P11	22
C1	1
C2	23
C3	24
C4	25
C5	Not testable

5.3.1.10.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**No EF selected** 1- select DF_{TEST},fid=1111 2- recNumber = 1 mode = REC_ACC_MODE_ABSOLUTE recOffset = 0 byte[] resp = new byte[20] respOffset = 0 respLength = 10 readRecord()	2-Shall throw uicc.access.UICCException with reason code NO_EF_SELECTED.	
2	**Read Absolute from Linear Fixed EF** 1 - select EF_{LARU}, fid=6F0c // Record pointer not set. 2 - recNumber = 2 mode = REC_ACC_MODE_ABSOLUTE recOffset = 0 respOffset = 0 respLength = 4 readRecord() 3- recNumber = 0 mode = REC_ACC_MODE_NEXT recOffset = 0 respOffset = 0 respLength = 4 readRecord()	2 - resp shall be: resp={0xAA,0xAA,0xAA,0xAA} 3 - resp shall be: Resp={0x55,0x55,0x55,0x55}	

Id	Description	API Expectation	APDU Expectation
3	**Read Current from Linear Fixed EF** //record pointer shall not be changed 1- recNumber = 0 mode = REC_ACC_MODE_CURRENT recOffset = 0 respOffset = 0 respLength = 4 readRecord()	resp shall be: resp={0x55,0x55,0x55,0x55}	

4	**Read Next from Linear Fixed EF**
	1- select EF_{LARU}, fid=6F0c
	//no record selected
	recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	respOffset = 0
	respLength = 4
	readRecord()
	2- recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	respOffset = 0
	respLength = 4
	readRecord()
		1- resp shall be:
		resp={0x55,0x55,0x55,0x55}
		
		2- resp shall be:
		resp={0xAA,0xAA,0xAA,0xAA}	

5	**Read Next from Linear Fixed EF** recNumber = 0 mode = REC_ACC_MODE_NEXT recOffset = 0 respOffset = 0 respLength = 4 readRecord()	Shall throw uicc.access.UICC Exception with reason code RECORD_NOT_FOUND.	

6	**Read Previous from Linear Fixed EF**
	1- recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	respOffset = 0
	respLength = 4
	readRecord()
	2- select EF_{LARU}, fid=6F0c
	//no record selected
	recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	respOffset = 0
	respLength = 4
	readRecord()
	3- Set the record to the first record by reading the file
	recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	respOffset = 0
	respLength = 4
		1- resp shall be:
		resp={0x55,0x55,0x55,0x55}
		
		2- resp shall be:
		resp={0xAA,0xAA,0xAA,0xAA}
		3- resp={0x55,0x55,0x55,0x55}	

7	**Read Previous from Linear Fixed EF** recNumber = 0 mode = REC_ACC_MODE_PREVIOUS recOffset = 0 respOffset = 0 respLength = 4 readRecord()	Shall throw uicc.access.UICCException with reason code RECORD _NOT_FOUND.	

Id	Description	API Expectation	APDU Expectation
8	**Read Absolute from Cyclic EF** 1 select EF_{CARU}, fid = 6F09 2- recNumber = 2 mode = REC_ACC_MODE_ABSOLUTE recOffset = 0 respOffset = 0 respLength = 3 readRecord() 3- recNumber = 1 readRecord() 4- Read the file in next mode to set the record pointer to the first position. recNumber = 0 mode = REC_ACC_MODE_NEXT recOffset = 0 respOffset = 0 respLength = 3 readRecord()	2 - resp shall be: resp={0xAA,0xAA,0xAA} 3 - resp shall be: resp={0xAA,0xAA,0xAA} 4 - resp shall be: resp={0xAA,0xAA,0xAA}	

9	**Read Current from Cyclic EF**
	//record pointer shall not be changed
	//from testcase before
	1- recNumber = 0
	mode = REC_ACC_MODE_CURRENT
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
		1- resp shall be:
		resp={0xAA,0xAA,0xAA}	

10	**Read Next from Cyclic EF** recNumber = 0 mode = REC_ACC_MODE_NEXT recOffset = 0 respOffset = 0 respLength = 3 readRecord()	resp shall be: resp={0xAA,0xAA,0xAA}	

11	**Read Next from Cyclic EF**
	1- recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
	2- select EF_{CARU}, fid = 6F09
	//no rec selected
	recNumber = 0
	mode = REC_ACC_MODE_NEXT
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
		1- resp shall be:
		resp={0x55,0x55,0x55}
		
		2- Shall throw uicc.access.UICCException with reason code RECORD_NOT_FOUND.	

12	**Read Previous from Cyclic EF** 1- recNumber = 0 mode = REC_ACC_MODE_PREVIOUS recOffset = 0 respOffset = 0 respLength = 3 readRecord()	1- resp shall be: resp={0xAA,0xAA,0xAA}	

13	**Read Previous from Cyclic EF**
	1- recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
	2- select EF_{CARU}, fid = 6F09
	// no rec selected
	recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
		1- resp shall be:
		resp={0x55,0x55,0x55}
		
		2- resp shall be:
		resp={0xAA,0xAA,0xAA}	

Id	Description	API Expectation	APDU Expectation
14	**Read Absolute from Linear Fixed EF beyond Records** 1- select EF_{LARU}, fid=6F0C 2- recNumber = -1 mode = REC_ACC_MODE_ABSOLUTE recOffset = 0 respOffset = 0 respLength = 4 readRecord() 3- recNumber = 3 readRecord()	2- Shall throw an uicc.access.UICCException with reason code UICCException.RECORD_NOT_FOUND. 3- Shall throw an uicc.access.UICCException with reason code UICCException.RECORD_NOT_FOUND.	

15	**No current record in linear fixed EF, read current**
	1- select EF_{LARU}, fid=6F0C // No current record
	2- recNumber = 0 // curr rec
	mode = REC_ACC_MODE_CURRENT
	recOffset = 0
	respOffset = 0
	respLength = 4
	readRecord()
		2 - Shall throw uicc.access.UICC Exception with reason code RECORD_NOT_FOUND.	

16	**recOffset < 0**
	1- select EF_{LARU}, fid=6F0C
	2- recNumber = 1 // rec 1
	mode = REC_ACC_MODE_ABSOLUTE
	recOffset = -1
	respOffset = 0
	respLength = 4
	readRecord()
		2 - Shall throw uicc.access.UICCException with reason code OUT_OF_RECORD_BOUNDARIES.	

17	**recOffset + respLength > Record Length** 1- select EF_{LARU}, fid=6F0C 2- recNumber = 1 mode = REC_ACC_MODE_ABSOLUTE recOffset = 2 respOffset = 0 respLength = 4 readRecord()	2 - Shall throw sim.access.SIMViewException with reason code OUT_OF_RECORD_BOUNDARIES.	
18	**Reading with invalid mode** 1- select EF_{LARU}, fid=6F0C 2- recNumber = 0 mode = 1 recOffset = 0 respOffset = 0 respLength = 4 readRecord() 3- mode = 5 readRecord()	2 - Shall throw uicc.access.UICCException with reason code COMMAND_INCOMPATIBLE. 3 - Shall throw uicc.access. UICCException with reason code COMMAND_INCOMPATIBLE.	
19	**resp is null** resp[] = null mode = REC_ACC_MODE_CURRENT respOffset = 0 respLength = 10 readRecord()	Shall throw java.lang.NullPointerException.	
20	**respOffset < 0** respOffset = -1 respLength = 10 readRecord ()	Shall throw java.lang.ArrayIndexOutOfBoundsException.	
21	**respLength < 0** respOffset = 0 respLength = -1 readRecord ()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
22	**respOffset + respLength > resp.length** respOffset = 11 respLength = 4 readRecord ()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
23	**EF is neither Cyclic nor Linear Fixed** 1- select EF_{TNU}, fid=6F02 2- respOffset = 0 respLength = 4 readRecord()	2 - Shall throw uicc.access.UICCException with reason code COMMAND_INCOMPATIBLE.	

Id	Description	API Expectation	APDU Expectation
24	**Access condition not fulfilled** 1- select EF_{CNR}, fid=6F04 2 - respLength = 3 readRecord()	2 - Shall throw uicc.access.UICCException with reason code SECURITY_STATUS_NOT_SATISFIED.	
25	**EF is deactivated** 1 - select EF_{CNU}, fid=6F05 deactivateFile() 2 - readRecord() 3 - activateFile	2 - Shall throw uicc.access.UICCException with reason code REF_DATA_INVALIDATED	

5.3.1.11 Method updateRecord

Test Area Reference: Api_4_Afv_Updr.

5.3.1.11.1 Conformance requirement

The method with 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,
                         UICCException
```

5.3.1.11.1.1 Normal execution

- CRRN1: Reads a record or a part of record of the current linear fixed or cyclic EF into byte array data[].
- CRRN2: If the access mode is REC_ACC_MODE_ CURRENT the current selected record will be updated. The current record pointer shall not be changed.
- CRRN3:If the access mode is REC_ACC_MODE_ABSOLUTE and the file is linear fixed EF, the record addresss by recNumber will be updated. The current record pointer shall not be changed.
- CRRN4: 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. The record pointer shall be updated.
- CRRN5: If the access mode is REC_ACC_MODE_NEXT and the record pointer has not been previously set within the selected EF, the record pointer shall be set to the first record and this record should be updated.
- CRRN6: If the access mode is REC_ACC_MODE_PREVIOUS the previous record relative to the current selected record will be selected and updated. The record pointer shall be updated.
- CRRN7: If the access mode is REC_ACC_MODE_PREVIOUS and the record pointer has not been previously set within the selected EF, then the record pointer should be set to the last record in this EF. This record should be updated.
- CRRN8: If the access mode is REC_ACC_MODE_PREVIOUS, the file is a cyclic EF, the oldest record will be updated independent of the current record pointer and this record becomes record number 1 and the current record.

5.3.1.11.1.2 Parameter errors

- CRRP1: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE_ABSOLUTE and recNumber is less than 0 or greater than records available, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD_NOT_FOUND.
- CRRP2: If the currently selected EF is linear fixed and the access mode is REC_ACC_MODE _CURRENT, recNumber is 0 and there is no current record selected, an instance of UICCException shall be thrown. The reason code shall be UICCException. RECORD_NOT_FOUND.
- 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 UICCException shall be thrown. The reason code shall be UICCException. RECORD_NOT_FOUND.
- 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 UICCException shall be thrown. The reason code shall be UICCException. RECORD_NOT_FOUND.
- CRRP5: If the specified offset into the selected record recOffset is less than 0, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT_OF_RECORD_BOUNDARIES.
- CRRP6: If recOffset plus dataLength is greater than the record length, an instance of UICCException shall be thrown. The reason code shall be UICCException.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 UICCException shall be thrown. The reason code shall be UICCException.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 UICCException shall be thrown. The reason code shall be UICCException.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.

5.3.1.11.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the currently selected EF is neither linear fixed nor cyclic, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND_INCOMPATIBLE.
- CRRC3: If the calling applet does not fulfil the access condition, UPDATE, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException. SECURITY_STATUS_NOT_SATISFIED.
- CRRC4: If the currently selected EF is deactivated and the file status of the EF does not allow for updating an deactivated file, an instance of UICCException shall be thrown. The reason code shall be UICCException. REF_DATA_INVALIDATED.
- CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.
- CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.3.1.11.2 Test area files

Test Source: Test_Api_4_Afv_Updr.java.

Test Applet: Api_4_Afv_ Updr_1.java.

Cap File: Api_4_Afv_updr.cap.

5.3.1.11.3 Test coverage

CRR number	Test case number
N1	2, 3, 4, 5, 7, 8, 10
N2	2
N3	3
N4	5
N5	4
N6	7, 8, 9, 10
N7	7
N8	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	Not testable
C6	Not testable

5.3.1.11.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	**Get the UICC AdminFileView** AdminFileViewBuilder.getTheUICCAdminFileView (CLEAR_ON_RESET)		
1	**No EF selected** RecNumber = 1 mode = REC_ACC_MODE_ABSOLUTE recOffset = 0 byte[] data = new byte[20] dataOffset = 0 dataLength = 10 updateRecord()	Shall throw uicc.access.UICC Exception with reason code NO_EF_SELECTED.	
2	**Update Absolute from Linear Fixed EF** 1- select DF_{TEST}, fid = 1111 2- select EF_{LARU}, fid = 6F0C // Record pointer not set. 3- recNumber = 2 mode = REC_ACC_MODE_ABSOLUTE data[0:3] = '11' recOffset = 0 dataOffset = 0 dataLength = 4 updateRecord() respOffset = 0 respLength = 0 readRecord() 4- // verify result read respOffset = 0 respLength = 4 recNumber = 0 mode = REC_ACC_MODE_CURRENT readRecord()	1- No exception shall be thrown. 2- No exception shall be thrown. 3- No exception shall be thrown. 4- Resp shall be: 11 11 11 11	

3	**Update Current from Linear Fixed EF**
	1- select DF_{TEST}, fid = 1111
	2- select EF_{LARU}, fid = 6F0C
	// 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_MODECURRENT
	updateRecord()
	5- //verify result
	respOffset = 0
	respLength = 4
	recNumber = 0
	mode = REC_ACC_MODE_ CURRENT
	readRecord()
		1- No exception shall be thrown.
		2- No exception shall be thrown.
		3- No exception shall be thrown.
		5- 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- select DF_{TEST}, fid = 1111 2- select EF_{LARU}, fid = 6F0C 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 4-// verify result readRecord()	1- No exception shall be thrown. 2- No exception shall be thrown. 4- No exception shall be thrown. resp shall be: resp[0] = '33' resp[1] = '33' resp[2] = '33' resp[3] = '33'	

Id	Description	API Expectation	APDU Expectation

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- //verify result
	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 uicc.access.UICCException with reason code RECORD_NOT_FOUND.	

7	**Update Previous from Linear Fixed EF, no record pointer set**
	1- select DF_{TEST}, fid = 1111
	2- select EF_{LARU}, fid = 6F0C
	3- recNumber = 0
	mode = REC_ACC_MODE_PREVIOUS
	recOffset = 0
	data[0:3] = '66'
	dataOffset = respOffset = 0
	dataLength = respLength = 4
	updateRecord()
	4- //verify result
	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'	

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()
	2- //verify result
	readRecord()
		1- No exception shall be thrown
		
		2- No exception shall be thrown.
		esp shall be:
		resp[0] = '77'
		resp[1] = '77'
		resp[2] = '77'
		resp[3] = '77'	

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	Shall throw sim.access.SIMViewException with reason code RECORD_NOT_FOUND.	

Id	Description	API Expectation	APDU Expectation

10	**Update Previous from Cyclic EF**
	1- select DF_{TEST}, fid = 1111
	2- select EF_{CARU}, fid = 6F09
	3- recNumber = 2
	mode = REC_ACC_MODE_ABSOLUTE
	recOffset = 0
	respOffset = 0
	respLength = 3
	readRecord()
	4- recNumber = 2
	mode = REC_ACC_MODE_PREVIOUS
	data[0:2] = 'FF'
	dataOffset = 0
	dataLength = 3
	updateRecord()
	5- //verify result
	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] = 'FF'resp[1] = 'FF'
		resp[2] = 'FF'	

11	**Update Absolute from Linear Fixed EF beyond Records** 1- select EF_{LARU}, fid = 6F0C 2-recNumber = -1 mode = REC_ACC_MODE_ABSOLUTE recOffset = 0 dataOffset = 0 dataLength = 4 updateRecord() 3- recNumber = 3 updateRecord()	1- No exception shall be thrown. 2- Shall throw uicc.access.UICCException with reason code RECORD_NOT_FOUND. 3- Shall throw uicc.access.UICCException with reason code RECORD_NOT_FOUND.	

12	**No current record in linear fixed EF, update current**
	1- select EF_{LARU}, fid = 6F0C
	// No curr rec
	2- recNumber = 0 // curr rec
	mode = REC_ACC_MODE _CURRENT
	recOffset = 0
	dataOffset = 0
	dataLength = 4
	updateRecord()
		1 - No exception shall be thrown.
		
		2 - Shall throw uicc.access.UICCException with reason code RECORD_NOT_FOUND.	

13	**recOffset < 0**
	1- select EF_{LARU}, fid = 6F0C
	2- recNumber = 1 // rec 1
	mode = REC_ACC_MODE_ABSOLUTE
	dataOffset = 0
	dataLength = 4
	updateRecord()
		1- No exception shall be thrown.
		2- Shall throw uicc.access.UICC Exception with reason code OUT_OF_RECORD_BOUNDARIES.	

14	**recOffset + dataLength > record.length** 1- select EF_{LARU}, fid = 6F0C 2- recNumber = 1 mode = REC_ACC_MODE_ABSOLUTE recOffset = 2 dataOffset = 0 dataLength = 4 updateRecord()	1- No exception shall be thrown. 2- Shall throw uicc.access.UICC Exception with reason code OUT_OF_RECORD_BOUNDARIES.	
15	**Updating with invalid mode** 1- select EF_{LARU}, fid = 6F0C 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 uicc.access.UICC Exception with reason code INVALID_MODE. 3 - Shall throw uicc.access. UICC Exception with reason code INVALID_MODE.	

Id	Description	API Expectation	APDU Expectation
16	**Updating Cyclic EF with invalid mode** 1- select DF_{TEST}, fid = 1111 2- select EF_{CARU}, fid = 6F09 set record pointer to record nr 1 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 updateRecord() 5- recNumber = 2 mode = REC_ACC_MODE_ABSOLUTE updateRecord()	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - Shall throw uicc.access. UICC Exception with reason code INVALID_MODE. 4 - Shall throw uicc.access.UICC Exception with reason code INVALID_MODE. 5 - Shall throw uicc.access. UICC Exception with reason code INVALID_MODE.	
17	**data[] is null** data[] = 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- select DF_{TEST}, fid = 1111 2- select EF_{TARU}, fid = 6F03 3- dataOffset = 0 dataLength = 4 updateRecord()	1- No exception shall be thrown. 2- No exception shall be thrown. 3- Shall throw uii.access.UICC Exception with reason code COMMAND_INCOMPATIBLE.	
22	**Access condition not fulfilled** 1- select EF_{CNU}, fid = 6F05 2- recOffset = 0 dataOffset = 0 dataLength = 1 mode = REC_ACC_MODE_PREVIOUS updateRecord() 3- fid = EFLNU select() 4- recNumber = 1 mode = REC_ACC_MODE_ CURRENT recOffset = 0 dataOffset = 0 dataLength = 1 updateRecord()	1- No exception shall be thrown. 2- Shall throw uicc.access.UICC Exception with reason code SECURITY_STATUS_NOT_SATISFIED. 3- No exception shall be thrown. 4- Shall throw uicc.access.UICC Exception with reason code SECURITY_STATUS_NOT_SATISFIED	
23	**EF is deactivated** 1- select EF_{CNR}, fid = 6F04 invalidate() 2- updateRecord() 3- activateFile() 4- restore the file content EF_{LARU}, EF_{CARU} **Restore the file content** 1- restore the file content of EF_{LARU}: record 1 = 0x55,0x55,0x55,0x55 record 2 = 0xAA,0xAA,0xAA,0xAA 2- restore the file content of EF_{CARU}: record 1 = 0x55,0x55,0x55 record 2 = 0xAA,0xAA,0xAA	1- No exception shall be thrown. 2- Shall throw uicc.access.UICCException with reason codeREF_DATA_INVALIDATED 3- No exception shall be thrown.	

5.3.1.12 Method searchRecord

Test Area Reference: Api_4_Afv_Sear.

5.3.1.12.1 Conformance requirement

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

```
public short searchRecord(byte mode,
                          short recordNum,
                          short searchIndication,
                          byte[] patt,
                          short pattOffset,
                          short pattLength,
                          short[] response,
                          short respOffset,
                          short respLength)
                   throws java.lang.NullPointerException,
                          java.lang.ArrayIndexOutOfBoundsException,
                          UICCException
```

5.3.1.12.1.1 Normal execution

- CRRN1: Search a given pattern in byte array patt[] of a current linear fixed or cyclic EF.
- CRRN2: If the pattern is found, the number of each record is stored in byte array response[] and the total number of updated bytes in the array response[] buffer is returned.
- CRRN3: If the value of respLength is greater than the number of records found, the whole response is copied into the response buffer and the number of elements copied is returned by the method.
- CRRN4: If the value of respLength is smaller than the number of found patterns, the first record numbers are copied into the response array and the value of respLength is returned.
- CRRN5: If mode is SIMPLE_SEARCH_START_FORWARD, the search starts at the given record number forward towards the end of the file.
- CRRN6: If mode is SIMPLE_SEARCH _START_BACKWARD, the search starts at a given record number backward towards to the beginning of the file.
- CRRN7: If mode is ENHANCED_SEARCH and SEARCH_INDICATION_START_BACKWARD_FROM_PREVIOUS is set in searchIndication, the search is backward starting from previous record towards to the beginning of the file.
- CRRN8: If the mode is ENHANCED_SEARCH and SEARCH_INDICATION_START_BACKWARD_FROM_PREVIOUS_GR is set in searchIndication, the search is backward starting at a given record from previous record towards to the beginning of the file.
- CRRN9: If the mode is ENHANCED_SEARCH and SEARCH_INDICATION_START_FORWARD_FROM_NEXT is set in searchIndication, the search is forward starting at the next record towards the end of the file.
- CRRN10: If the mode is ENHANCED_SEARCH and SEARCH_INDICATION_FORWARD_FROM_NEXT_GR is set in searchIndication, the search is forward starting at a given record number towards to the end of the file.
- CRRN11: If the mode is ENHANCED_SEARCH, and bit 4 of the most significant byte of the searchIndication is not set, the search starts in the record from the offset (absolute position) given in the less significant byte of searchIndication.
- CRRN12: If the mode is ENHANCED_SEARCH, and bit 4 of the most significant byte of the searchIndication is set, the search starts in the record after the first occurrence of the value contained in the less significant byte of searchIndication.
- CRRN13: If pattern given in patt[] is not found, the method returns 0.

- CRRN14: If one or more matches are found the record pointer shall be set to the first record where the search pattern was found.

5.3.1.12.1.2 Parameter errors

- CRRP1: If mode is not 4, 5 or 6, an instance of UICCException shall be thrown. The reason code shall be UICCException.INVALID_MODE.
- CRRP2: If the pattern array patt is null, an instance of java.lang.NullPointerException shall be thrown.
- CRRP3: If the response array response is null, an instance of java.lang.NullPointerExceptino shall be thrown.
- CRRP4: If parameter pattOffset is negative, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
- CRRP5: If parameter pattLength is negative, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
- CRRP6: If parameter respOffset is negative, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
- CRRP7: If parameter respLength negative, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
- CRRP8: If parameter pattOffset plus pattLength are greater than the length of array patt, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
- CRRP9: If parameter respOffset plus respLength are greater than the length of array response a ArrayIndexOutOfBoundsException shall be thrown.
- CRRP10: If parameter recordNum is negative, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD_NOT_FOUND.
- CRRP11: If parameter recordNum is greather than, the total number of records from the currently selected EF, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD_NOT_FOUND.
- CRRP12 If pattLength is greater than record size of the currently selected EF an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT_OF_FILE_BOUNDARIES.

5.3.1.12.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the currently selected EF is not linear fixed or cyclic, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND_INCOMPATIBLE.
- CRRC3: If the calling applet does not fulfil the access condition, READ, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.
- CRRC4: If the currently selected EF is deactivated and the file status of the EF does not allow for reading a deactivated file, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF_DATA_INVALIDATED.
- CRRC5: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.3.1.12.2 Test area files

Test Source: Test_Api_4_Afv_Sear.java.

Test Applet: Api_4_Afv_Sear_1.java.

Cap File: Api_4_Afv_sear.cap.

5.3.1.12.3 Test coverage

CRR number	Test case number
N1	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37
N2	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37
N3	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37
N4	12
N5	2, 28
N6	3, 29
N7	6, 7, 34, 35
N8	8, 9, 36, 37
N9	10, 11, 30, 31
N10	12, 13, 32, 33
N11	6, 8, 10, 12, 30, 32, 34, 36
N12	7, 9, 11, 13, 31, 33, 35, 37
N13	2, 3, 5, 7, 9, 11, 28, 31
N14	6, 7, 10, 11, 30, 31, 34, 35
P1	13
P2	14
P3	15
P4	16
P5	17
P6	18
P7	19
P8	20
P9	21
P10	22
P11	23
P12	24
C1	1
C2	25
C3	26
C4	27
C5	Not testable

5.3.1.12.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**No EF selected** 1- select DF_{TEST}, fid=1111 2- searchRecord()	2-shall throw uicc.access.UICC Exception with reason code NO_EF_SELECTED.	

Id	Description	API Expectation	APDU Expectation
2	**Fixed linear EF,** **Simple mode search forward** 1- select EF_{LSEA}, fid=6F1A 2- mode = SIMPLE_SEARCH_START_FORWARD recordNum = 1 patt[]={0x10,0x03,0x04} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 3- Simple mode search forward mode = SIMPLE_SEARCH_START_FORWARD recordNum = 2 patt[]={0x10,0x03,0x04} pattOffset = 1 pattLength = 2 resp.length = 4 respOffset = 1 respLength = 3 searchRecord()	2- no exception shall be thrown Shall return 0. response shall be: response={0,0,0,0} 3- Shall return 2. response shall be: response={0,2,4,0}	
3	**Simple mode, search backward** 1- mode = SIMPLE_SEARCH_START_BACKWARD recordNum = 1 patt[]={0x08,0x0A,0x0B} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 2 respLength = 2 searchRecord() 2-mode = SIMPLE_SEARCH_START_BACKWARD recordNum = 6 patt[]={0x08,0x09,0x0A,0x0B } pattOffset = 1 pattLength = 2 response[] = {0,0,0,0} respOffset = 1 respLength = 3 searchRecord()	1- shall return 0. response shall be: response={0,0,0,0} 2- shall return 3. response shall be: response={0,4,3,1}	
4	**Enhanced Mode, search backward from previous record, start from an offset in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS + 0x0009 recordNum = 0 patt[]={0x01,0x02,0x03,0x04} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS +0x0000 recordNum = 0 patt[]={0x0C,0x0D,0x0E,0x0F,0x01,0x02} pattOffset = 0 pattLength = 5 response[] = {0,0,0,0} respOffset = 2 respLength = 2 searchRecord()	1- shall return 1, response shall be: resp={3,0,0,0} 2- shall return 1 response shall be: response={0,0,2,0}	

Id	Description	API Expectation	APDU Expectation
5	**Enhanced Mode, search backward from previous record, start from a value in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS + 0x0810 recordNum = 0 patt[]={0x01,0x02,0x03,0x04} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 2- perform 3 readRecord() in next mode to set current pointer to pointer 5 3- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS +0x080E recordNum = 0 patt[]={0x01,0x02,0x03,0x04} pattOffset = 3 pattLength = 1 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 0, response shall be: resp={0,0,0,0} 3- shall return 2 response shall be: response={4,2,0,0}	
6	**Enhanced Mode, search backward from previous given record, start from an offset in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS _GR + 0x0000 recordNum = 1 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS _GR + 0x0004 recordNum = 6 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 1, response shall be: resp={1,0,0,0} 2- shall return 4 response shall be: response={5,4,3,2}	

Id	Description	API Expectation	APDU Expectation
7	**Enhanced Mode, search backward from previous given record, start from a value in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS _GR + 0x080D recordNum = 1 patt[]={0x0E,0x0E,0x0E} pattOffset = 1 pattLength = 1 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_BACKWARD_FROM_PREVIOUS _GR + 0x0800 recordNum = 6 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 1, response shall be: resp={1,0,0,0} 2- shall return 0 response shall be: response={0,0,0,0}	
8	**Enhanced Mode, search forward from next record, start from an offset in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT + 0x0003 recordNum = 0 patt[]={0x00,0x0A,0x0B} pattOffset = 1 pattLength = 2 response[] = {0,0,0} respOffset = 2 respLength = 2 searchRecord() 2- Perform readRecord() in previous mode 3- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT + 0x0003 recordNum = 0 patt[]={0x00,0x0A,0x0B} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 2 response shall be: resp={0,0,3,4} 3- shall return 1 response shall be: response={4,0,0,0}	

Id	Description	API Expectation	APDU Expectation
9	**Enhanced Mode, search forward from next record, start from a value in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT + 0x0804 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0} respOffset = 2 respLength = 2 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT + 0x0801 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 2 pattLength = 1 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 0, response shall be: resp={0,0,0,0} 2- shall return 2 response shall be: response={5,6,0,0}	
10	**Enhanced Mode, search forward from next given record, start from an offset in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT_GR + 0x0007 recordNum = 1 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 1 respLength = 3 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT_GR + 0x000C recordNum = 3 patt[]={0x03,0x02,0x01} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 3, response shall be: resp={0,3,4,5} 2- shall return 1 response shall be: response={6,0,0,0}	

Id	Description	API Expectation	APDU Expectation
11	**Enhanced Mode, search forward from next given record, start from a value in record.** 1- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT_GR + 0x080D recordNum = 5 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT_GR + 0x080C recordNum = 5 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 0, response shall be: resp={0,0,0,0} 2- shall return 1 response shall be: response={5,0,0,0}	
12	**Simple mode, total number of found patterns exceed response[]** 1- mode = SIMPLE_SEARCH_START_FORWARD recordNum = 1 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0} respOffset = 0 respLength = 4 searchRecord() 2- mode = SIMPLE_SEARCH_START_FORWARD recordNum = 1 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0,0} respOffset = 0 respLength = 4 searchRecord()	1- shall return 4 response shall be: response={1,2,3,4} 2- shall return 4 response shall be: response={1,2,3,4,0}	
13	**Invalid mode** mode = 0x14 (simple search forward with SFI) searchIndication= 0 recordNum = 2 patt[]={0x01,0x02,0x03} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 2 respLength = 2 searchRecord()	shall throw an uicc.access.UICC Exception with reason code INVALID_MODE.	
14	**Pattern array is null** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]= null pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 2 respLength = 2 searchRecord()	shall throw an java.lang.NullPointerException.	

Id	Description	API Expectation	APDU Expectation
15	**Response array is null** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = null respOffset = 0 respLength = 5 searchRecord()	shall throw an instance of java.lang.NullPointerException.	
16	**pattOffset<0** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = -1 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	shall throw an instance of java.lang.ArrayIndexOutOfBoundsException.	
17	**pattLength<0** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = -1 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	shall throw an instance of java.lang.ArrayIndexOutOfBoundsException.	
18	**respOffset <0** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = -1 respLength = 5 searchRecord()	shall throw an instance of java.lang.ArrayIndexOutOfBoundsException.	
29	**respLength <0** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 0 respLength = -1 searchRecord()	shall throw an instance of java.lang.ArrayIndexOutOfBoundsException.	
20	**PattOffset + pattLength > patt[]** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 2 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 1 respLength = 5 searchRecord()	shall throw an instance of java.lang.ArrayIndexOutOfBoundsException.	

Id	Description	API Expectation	APDU Expectation
21	**RespOffset + respLength > response[]** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 3 respLength = 3 searchRecord()	shall throw an instance of java.lang.ArrayIndexOutOfBoundsException.	
22	**recordNum < 0** mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = -1 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	shall throw an uicc.access.UICC Exception with reason code RECORD_NOT_FOUND	
23	**RecordNum > total number of file records** 1- mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 7 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord() 2- mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	1- shall throw an uicc.access.UICC Exception with reason code RECORD_NOT_FOUND 2- shall throw an uicc.access.UICC Exception with reason code RECORD_NOT_FOUND	
24	**pattlength > record length** 1- mode = SIMPLE_SEARCH_START_FORWARD searchIndication= 0 recordNum = 3 patt[16]={0x55,0x55,…,0x55} pattOffset = 0 pattLength = 16 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord() 2- mode = ENHANCED_SEARCH searchIndication= SEARCH_INDICATION_FORWARD_FROM_NEXT_GR + 0x000E recordNum = 3 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	1- shall throw an uicc.access.UICCException with reason code OUT_OF_FILE_BOUNDARIES. 2- shall throw an uicc.access.UICCException with reason code OUT_OF_FILE_BOUNDARIES.	

Id	Description	API Expectation	APDU Expectation
25	**Wrong file structure** 1- select EF_{TDAC}, fid=6F0F 2- searchRecord()	2- shall throw an uicc.access.UICCException with reason code COMMAND_INCOMPATIBLE	
26	**Security status not satisfied** 1- select EF_{LNR}, fid=6F0A 2- searchRecord()	2- shall throw an uicc.access.UICCException with reason code SECURITY_STATUS_NOT_SATISFIED	
27	**File deactivated** 1- select EF_{LARU}, fid=6F10 2- deactivateFile EF_{LARU} 3- searchRecord() 4- activateFile()	3- shall throw an uicc.access.UICCException with reason code DATA_INVALIDATED	
28	**Cyclic EF, Simple mode search forward** 1- select EF_{CSEA}, fid=6F1B 2- mode = SIMPLE_SEARCH_START_FORWARD recordNum = 1 patt[]={0x10,0x03,0x04} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord() 3- mode = SIMPLE_SEARCH_START_FORWARD recordNum = 2 patt[]={0x10,0x03,0x04} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 2 respLength = 3 searchRecord() 4- updateRecord() in previous mode with value {0x03,0x02,0x01,0x03,0x02,0x01,0x03,0x02,0x01,0x03,0x02,0x01,0x03,0x02,0x01} (new record 1 is set to previous record 6) 5- mode = SIMPLE_SEARCH_START_FORWARD recordNum = 2 patt[]={0x10,0x03,0x04} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 2 respLength = 3 searchRecord()	2- shall return 0 response shall be: response={0,0,0,0,0} 3- Shall return 3. response shall be: response={0,0,2,4,1} 5- Shall return 3. response shall be: response={0,0,2,3,5}	
29	**Cyclic EF, Simple mode search backward** mode = SIMPLE_SEARCH_START_BACKWARD recordNum = 3 patt[]={0x10,0x03,0x04} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 1 respLength = 4 searchRecord()	shall return 3 response shall be: response={0,3,2,5,0}	

Id	Description	API Expectation	APDU Expectation
30	**Cyclic EF, Enhanced mode, search forward from next record, start from an offset in record** mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_FORWARD_FROM_NEXT + 0x0009 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0,0} respOffset = 2 respLength = 3 searchRecord()	shall return 3 response shall be: response={0,0,4,5,6}	
31	**Cyclic EF, Enhanced mode, search forward from next record, start from a value in record** mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_FORWARD_FROM_NEXT + 0x0810 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 3 response[] = {0,0,0,0,0} respOffset = 2 respLength = 3 searchRecord()	shall return 0 response shall be: response={0,0,0,0,0}	
32	**Cyclic EF, Enhanced mode, search forward from next given record, start from an offset in record** mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_FORWARD_FROM_NEXT_GR + 0x0005 recordNum = 3 patt[]={0x01,0x02,0x03} pattOffset = 0 pattLength = 1 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	shall return 5 response shall be: response={3,4,5,6,1}	
33	**Cyclic EF, Enhanced mode, search forward from next given record, start from a value in record** 1- mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_FORWARD_FROM_NEXT_GR + 0x0805 recordNum = 6 patt[]={0x0E,0x0F,0x00} pattOffset = 0 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord() 2- Restore EF initial state (record 1 shall be assigned to the record that content is {0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0A, 0x0B,0x0C,0x0D,0xE,0x0F}) using 5 updateRecord() in previous mode	1- shall return 2 response shall be: response={2,4,0,0,0}	

Id	Description	API Expectation	APDU Expectation
34	**Cyclic EF, Enhanced mode, search backward from previous record, start from an offset in record** 1- Set current record pointer to record 6 using 5 readRecord() in next mode 2- mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_BACKWARD_FROM_PREVIOUS + 0x0003 recordNum = 0 patt[]={0x02,0x01,0x00} pattOffset = 0 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 3 respLength = 2 searchRecord()	2- shall return 1 response shall be: response={0,0,0,6,0}	
35	**Cyclic EF, Enhanced mode, search backward from previous record, start from a value in record** mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_BACKWARD_FROM_PREVIOUS + 0x0801 recordNum = 0 patt[]={0x01,0x02,0x03} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	shall return 5 response shall be: response={5,4,3,2,1}	
36	**Cyclic EF, Enhanced mode, search backward from given record, start from an offset in record** mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_BACKWARD_FROM_PREVIOUS_GR + 0x0003 recordNum = 5 patt[]={0x02,0x01,0x00} pattOffset = 0 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 3 respLength = 2 searchRecord()	shall return 1 response shall be: response={0,0,0,6,0}	
37	**Cyclic EF, Enhanced mode, search backward from given record, start from a value in record** mode = ENHANCED_MODE searchIndication= SEARCH_INDICATION_START_BACKWARD_FROM_PREVIOUS + 0x0801 recordNum = 3 patt[]={0x01,0x02,0x03} pattOffset = 1 pattLength = 2 response[] = {0,0,0,0,0} respOffset = 0 respLength = 5 searchRecord()	shall return 5 response shall be: response={3,2,1,5,4}	

5.3.1.13 Method increase

Test Area Reference: Api_4_Afv_Incr.

5.3.1.13.1 Conformance requirement

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

```
public short increase(byte[] incr,
                      short incrOffset,
                      short incrLength,
                      byte[] resp,
                      short respOffset)
               throws java.lang.NullPointerException,
                      java.lang.ArrayIndexOutOfBoundsException,
                      UICCException
```

5.3.1.13.1.1 Normal execution

- CRRN1: This method increases the current cyclic EF record.
- CRRN2: The response buffer will only contain the value of the increased record.

5.3.1.13.1.2 Parameter errors

- CRRP1: If the array incr is null, an instance of NullPointerException shall be thrown.
- CRRP2: If the array resp is null, an instance of NullPointerException shall be thrown.
- CRRP3: If incrOffset is negative, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP4: If incrLength is negative, an instance of ArrayIndexOutOfBoundsException shall be thrown.
- CRRP5: If respOffset is negative, an instance of ArrayIndexOutOfBoundsExceptoin shall be thrown.
- CRRP6: If incrOffset plus incrLength, is greater than the length of array incr, an instance of ArrayIndexOutOfBoundsException shall be thrown and no increase is performed.
- CRRP7: If respOffset is greater than the length of array resp, an instance of ArrayIndexOutOfboundsException shall be thrown.
- CRRP8: 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 UICCException shall be thrown. The reason code shall be UICCException.MAX_VALUE_REACHED.
- CRRP9: If incrLength is greater than 127, and exception shall be thrown.

5.3.1.13.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException. shall be thrown. The reason code shall be UICCException.MEMORY_PROBLEM.
- CRRC3: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.
- CRRC4: If file is not a cyclic one, an instance of the UICCException shall be thrown. The reason code shall be UICCException.COMMAND_INCOMPATIBLE.
- CRRC5: If the calling applet does not fulfil the access condition, INCREASE, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOTSATISFIED.

- CRRC6: If the currently selected EF is invalidated, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF_DATA_INVALIDATED.
- CRRC7: If the currently selected cyclic EF has no record, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_RECORD_FOUND.

5.3.1.13.2 Test areafiles

Test Source: Test_Api_4_Afv_Incr.java.

Test Applet: Api_4_Afv_Incr_1.java.

Cap File: Api_4_Afv_incr.cap.

5.3.1.13.3 Test coverage

CRR number	Test case number
N1	1, 2, 3, 15
N2	2, 3, 15
P1	4
P2	9
P3	6
P4	5
P5	10
P6	7
P7	11
P8	8
P9	15
C1	1
C2	Not testable
C3	Not testable
C4	12
C5	13
C6	14
C7	Not testable

5.3.1.13.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**No EF selected** 1- select DF_{TEST} fid=1111 2- byte[] incr = new byte[4] byte[] resp = new byte[4] incrOffset = 0 incrLength = 2 respOffset = 0 increase()	2- An UICCException.NO_EF_SELECTED should be thrown	

Id	Description	API Expectation	APDU Expectation

2	**increase , verify response**
	
	1- select EF_{CARU}, fid=6F09
	set the record pointer with
	readRecord() in PREVIOUS mode
	2-//Set both record to 00 00 00
	mode = REC_ACC_MODE_PREVIOUS
	data[] = {0x00,0x00,0x00}
	recOffset = 0
	dataOffset = 0
	dataLength = 3
	updateRecord() //update Record 1
	updateRecord() //update Record 2
	3- incr[] = {0x00,0x00,0x01}
	incrOffset = 0
	incrLength = 3
	resp.length = 4
	respOffset = 0
	ret = 3
	increase()
		3- resp[] = {0x00,0x00,0x01,0x00}	

3	**increase, verify file** 1- incr[]={0x00,0x00,0x00,0x02} incrOffset = 1 incrLength = 3 resp.length = 4 respOffset = 1 increase() 2- resp[] = {0x00,0x00,0x00,0x00} recNumber = 0 mode = REC_ACC_MODE_ABSOLUTE_CURRENT recOffset = 0 resp.length =4 respOffset = 0 respLength = 3 readRecord()	1- resp[] = {0x00,0x00,0x00,0x03} 2- resp[] = {0x00,0x00,0x03,0x00}	
4	**incr[] is null** incr[] = null incrOffset = 0 incrLength = 1 resp.length = 4 respOffset = 0 increase()	Shall throw java.lang.NullPointerException.	
5	**incrLength< 0** incr.length = 4 incrOffset = 0 incrLength = -1 resp.length = 4 respOffset = 0 increase()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
6	**incrOffset < 0** incr.length = 4 incrOffset = -1 incrLength = 1 resp.length = 4 respOffset = 0 increase()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
7	**IncrOffset + incrLength > incr.length** incr.length = 4 incrOffset = 1 incLength = 4resp.length = 4 respOffset = 0 increase()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	

Id	Description	API Expectation	APDU Expectation

8	**Reach Maximum Value**
	
	1- incr[0:3] = 0xFF
	incrOffset = 0
	incrLength = 3
	resp.length = 4
	respOffset = 0
	increase()
	
	2- //Set both record to FF FF FF
	mode = REC_ACC_MODE_PREVIOUS
	data[] = {0xFF,0xFF,0xFF}
	recOffset = 0
	dataOffset = 0
	dataLength = 3
	updateRecord() //update Record 1
	updateRecord() //update Record 2
	
	3- incr[] = {0x00,0x00,0x01}
	incrOffset = 0
	incrLength = 3
	resp.length = 4
	respOffset = 0
	increase()
		1- Shall throw uicc.access.UICCException with reason code MAX_VALUE_REACHED.
		
		2- Shall throw uicc.access.UICCException with reason code MAX_VALUE_REACHED.	

9	**resp[] is null** incr.length = 4 incrOffset = 0 incrLength = 1 resp[] = null respOffset = 0 increase()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
10	**respOffset < 0** incr.length = 4 incrOffset = 0 incrLength = 1 resp.length = 4 respOffset = -1 increase()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
11	**respOffset + recordLength > resp.length** incr.length = 4 incrOffset = 0 incrLength = 3 resp.length = 3 respOffset = 2 increase()	Shall throw java.lang. ArrayIndexOutOfBoundsException.	
12	**EF is not Cyclic** 1- select EF_{TARU} fid= 6F03 2- incr.length= 3 incrOffset = 0 incrLength = 3 resp.length = 3 respOffset = 0 increase() 3 - select EF_{LARU}, fid=6F0C 4 - incr.length= 3 incrOffset = 0 incrLength = 3 resp.length = 3 respOffset = 0 increase()	2 - Shall throw uicc.access.UICCException with reason code COMMAND_INCOMPATIBLE. 4 - Shall throw uicc.access.UICCException with reason code COMMAND_INCOMPATIBLE.	
13	**Access condition not fulfilled** 1- select EF_{CNIC}, fid=6F06 2- incr.length= 3 incrOffset = 0 incrLength = 3 resp.length = 3 respOffset = 0 increase()	2 - Shall throw uicc.access.UICCException with reason code SECURITY_STATUS_NOT_SATISFIED.	

Id	Description	API Expectation	APDU Expectation
14	**EF is invalidated** 1-select EF_{CARU}, fid=6F09 2 - invalidate() 3 - incr.length= 3 incrOffset = 0 incrLength = 3 resp.length = 3 respOffset = 0 increase() 4 - rehabilitate() 5- Restore initial content of EF_{CARU}	3 - Shall throw uicc.access.UICCException with reason code REF_DATA_INVALIDATED	
15	**incrLength out of range** 1- Create an EF Cyclic with 1 record of 0x7F length, fid=0x2C7F 2- Select EF Cyclic, fid=0x2C7F 3- Set record to following value rec[0] = 0; rec[1..126] = 0xFF with an update record. 4- incr.length=128 incrOffset = 1 incrLength = 127 resp.length = 255 respOffset = 0 Incr[] initialized to = {0x00,…,0x00,0x01} respOffset = 0 ret = 0x7F increase() 5- incr.length=128 incrOffset = 0 incrLength = 128 resp.length = 255 respOffset = 0 Incr[] initialized to 0 respOffset = 0 increase() 6- Delete EF Cyclic with fid=0x2C7F	4- resp[0..126] = {0x01,0x00,0x00,…,0x00} 5- Shall throw an exception	

5.3.1.14 Method deactivateFile

Test Area Reference: Api_4_Afv_Dacf.

5.3.1.14.1 Conformance requirement

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

```
public void deactivateFile()
                    throws UICCException
```

5.3.1.14.1.1 Normal execution

- CRRN1: The currently selected EF of the calling applet shall be deactivated, as defined in ETSI TS 102 222 [7].

5.3.1.14.1.2 Parameter errors

No requirements.

5.3.1.14.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the calling applet does not fulfil the access condition, activate, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.
- CCRC3: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.3.1.14.2 Test area files

Test Sourec: Test_Api_4_Afv_Dacf.java.

Test Applet: Api_4_Afv_Dacf_1.java.

Cap File: Api_4_Afv_dacf.cap.

5.3.1.14.3 Test coverage

CRR number	Test case number
N1	2, 3
C1	1
C2	4
C3	Not testable

5.3.1.14.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Test applet is installed with no access right on Application Pin2		
1	**No EF is selected** 1- select DF_{TEST} fid=1111 2- call deactivateFile()	2- An UICCException NO_EF_SELECTED is thrown	
2	**Deactivate activated File** 0- Select root directory 1- Select EF_{UICC} fid=2FF0 2- ReadBinary EF_{UICC} 3- Deactivate EF_{UICC} 4- ReadBinary EF_{UICC}	2- No Exception shall be thrown 4- UICCException.REF_DATA_INVALIDATED is thrown	
3	**Deactivate deactivated File** 1- deactivateFile EF_{UICC} 2- activateFile EF_{UICC}	1- No Exception shall be thrown	
4	**Access condition not fulfilled** 1- select DF_{TEST} fid=1111 2- select EF_{LADA} fid=6F15 3- deactivateFile EF_{LADA}	3- An UICCException SECURITY_STATUS_NOT_SATISFIED is thrown	

5.3.1.15 Method activateFile

Test Area Reference: Api_4_Afv_Actf.

5.3.1.15.1 Conformance requirement

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

```
public void activateFile()
                throws UICCException
```

5.3.1.15.1.1 Normal execution

- CRRN1: The currently selected EF of the calling applet shall be activated, as defined in ETSI TS 102 222 [7].

5.3.1.15.1.2 Parameter errors

No requirements.

5.3.1.15.1.3 Context errors

- CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO_EF_SELECTED.
- CRRC2: If the calling applet does not fulfil the access condition, activate, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY_STATUS_NOT_SATISFIED.
- CRRC3: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL_ERROR.

5.3.1.15.2 Test area files

Test Source: Test_Api_4_Afv_Actf.java.

Test Applet: Api_4_Afv_Actf_1.java.

Cap File: Api_4_Afv_actf.cap.

5.3.1.15.3 Test coverage

CRR number	Test case number
N1	2, 3
C1	1
C2	4
C3	Not testable

5.3.1.15.4 Test procedure

Id	Description	API Expectation	APDU Expectation
0	Test applet is installed with no access right on Application Pin2		
1	**No EF is selected** 1- Select DF_{TEST} fid=1111 2- Call activateFile()	 2- A UICCException NO_EF_SELECTED is thrown	
2	**Activate deactivated File** 0- Select Root directory 1- Select EF_{UICC} fid=2FF0 2- ReadBinary EF_{UICC} 3- Deactivate EF_{UICC} 4- ReadBinary EF_{UICC} 5 -ActivateFile EF_{UICC} 6- ReadBinary EF_{UICC}	 2- No Exception shall be thrown 4- UICCException.REF_DATA_INVALIDATED is thrown 6- No Exception shall be thrown	

Id	Description	API Expectation	APDU Expectation
3	**Activate activated File** ActiveFile EF_{UICC}	No Exception shall be thrown	
4	**Access condition not fulfilled** 1- Select DF_{TEST} fid=1111 2- Select EF_{LADA} fid=6F15 3- ActivateFile EF_{LADA}	3- A UICCException SECURITY_STATUS_NOT_SATISFIED is thrown	

5.3.2 Class AdminFileViewBuilder

5.3.2.1 Method getTheUICCAdminFileView

Test Area Reference: Api_4_Afb_Gtafb.

5.3.2.1.1 Conformance requirement

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

```
public static AdminFileView getTheUICCAdminFileView(byte event)
                throws javacard.framework.SystemException
```

5.3.2.1.1.1 Normal execution

- CRRN1: returns a reference to class which implements the FileView interface on the UICC file system.
- CRRN2: return null if one of the javacard.framework.Applet.register(..) method has not previously invoked by the applet invoking this method or the filesystem server returns null.
- CRRN3: It is not possible to get access to files which are located under any ADF with this FileView.
- CRRN4: After a successful invocation of the method, the MF is the current selected file.
- CRRN5: A separate and independent file context shall be associated with each and every FileView object: the operation performed on files in a given FileView object shall not affect the file context associated with any other FileView object. This context can be transient or persistent depending on what was required by the Applet during the creation of the FileView object.

5.3.2.1.1.2 Parameter errors

- CRRP1: If event is not one of the following values JCSystem.NOT_A_TRANSIENT_OBJECT, JCSystem.CLEAR_ON_DESELECT, or JCSystem.CLEAR_ON_RESET a SystemException is thrown. The value of the SystemException shall be SystemException.ILLEGAL_VALUE.

5.3.2.1.1.3 Context errors

- CRRC1: If event is JCSystem.CLEAR_ON_RESET or JCSystem.CLEAR_ON_DESELECT and not enough transient memory space is available a SystemException shall be thrown with reason code SystemException.NO_TRANSIENT_SPACE.
- CRRC2: If event is JCSystem.CLEAR_ON_DESELECT and the applet is not a currently selected applet a SystemException with reason code SystemException.ILLEGAL_TRANSIENT shall be thrown.

5.3.2.1.2 Test area files

Test Source: Test_ Api_4_Afb_Gtafb.java.

Test Applet: Api_4_Afb_Gtafb_1.java.

Cap File: Api_4_Afb_Gtafb.cap.

5.3.2.1.3 Test coverage

CRR number	Test case number
N1	2
N2	1
N3	2
N4	2
N5	3
P1	7
C1	5, 6 Testable only if available transient space is lower than 32767
C2	4

5.3.2.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Method returns null** Install Applet1 with full access rights on the UICC file system Invoke the method getTheUICCView before the javacard.framework.Applet.register(..) method invocation	returns null	
2	**Normal execution** 1- Envelope menu selection is sent to the UICC Invoke the method getTheUICCView() with the event JCSystem.NOT_A_TRANSIENT_OBJECT and stores the result in a class variable FV1 Applet1 calls status() command Select DF_{Test} using FV1 Select EF_{TARU} using FV1 Read first 3 bytes using FV1 Reset Terminal profile 2- Envelope menu selection is sent to the UICC Applet1 calls FV1.status() command Applet1 calls FV1.select(0x7FFF) Invoke the method getTheUICCView() with the event JCSystem.CLEAR_ON_RESET and stores the result in a class variable FV2 Applet1 calls FV2.status() command Select DF_{Test} using FV2 Select EF_{TARU} using FV2 Read first 3 bytes using FV2 Reset	1- Applet1 is triggered No exception shall be thrown Current selected DF is the MF Expected value is {FF FF FF} 2- Applet1 is triggered Current selected DF is DF_{Test} UICCException.FILE_NOT_FOUND is thrown No exception shall be thrown Current selected DF the MF Expected value is {FF FF FF}	

Id	Description	API Expectation	APDU Expectation
	Terminal profile 4 - Envelope menu selection is sent to the UICC Applet1 calls status() command Applet1 calls select(0x7FFF) 5- Select the Applet by AID Invoke the method in the method process() getTheUICCView() with the event JCSystem.CLEAR_ON_DESELECT and stores the result in a class variable Applet1 calls status() command Select DF_{Test} using FV3 Select EF_{TARU} using FV3 Read first 3 bytes using FV3 Select ADF2 by AID 6- Select the Applet by AID Applet1 calls status() command Applet1 calls select(0x7FFF)	4- Applet1 is triggered Current selected DF is the MF UICCException.FILE_NOT_FOUND is thrown 5- Applet1 is selected No exception shall be thrown Current selected DF the MF Expected value is {FF FF FF} 6- Applet1 is selected Current selected DF is the MF UICCException.FILE_NOT_FOUND is thrown	
3	**Fileview context independency** 1- Envelope menu selection is sent to the UICC 2- Check that previous fileviews are different (FV1 != FV2 != FV3) 3- Select DF_{Test}/EF_{LARU} using FV1 4- Select DF_{Test}/EF_{CARU} using FV2 5- Select DF_{Test}/EF_{CARU} using FV3 6- Read record number 1 using FV1 (in absolute mode) 7- Read record number 2 using FV2 (in absolute mode)	1- Applet1 is triggered 3- No exception shall be thrown 4- No exception shall be thrown 5- An exception is thrown 6- Expected value is “55 55 55 55” 7- Expected value is “AA AA AA”	
4	**ILLEGAL_TRANSIENT SystemException** 1- Envelope menu selection is sent to the UICC 2- Applet1 calls getTheUICCView() method with the event JCSystem. CLEAR_ON_DESELECT	1- Applet1 is triggered 2- SystemException. ILLEGAL_TRANSIENT is thrown	

Id	Description	API Expectation	APDU Expectation
5	**NO_TRANSIENT_SPACESystemException with CLEAR_ON_RESET Fileview object** 1- Get the available transient memory space using method length=JCSystem.getAvailableMemory(MEMORY_TYPE_TRANSIENT_RESET) 2- If length < 32767, (test case could be performed) 2.1- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR_ON_RESET) 2.2- Applet calls method getTheUICCView() with event JCSystem.CLEAR_ON_RESET 4- Reset	1- No exception shall be thrown 2.1- No exception shall be thrown 2.2- SystemException. NO_TRANSIENT_SPACE is thrown	
6	**NO_TRANSIENT_SPACE SystemException with CLEAR_ON_DESELECT Fileview object** 1- Select the Applet by AID 2- Get the available transient memory space using method length=JCSystem.getAvailableMemory(MEMORY_TYPE_TRANSIENT_DESELECT) 3- If length < 32767, (test case could be performed) 3.1- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR_ON_DESELECT) 3.2- Applet calls method getTheUICCView() with event JCSystem.CLEAR_ON_DESELECT 4- Reset	1- Applet1 is selected 2- No exception shall be thrown 3.1- No exception shall be thrown 3.2- SystemException. NO_TRANSIENT_SPACE is thrown	
7	**ILLEGAL_VALUE SystemException** Invoke the method getTheUICCView() with every event codes except 0,1,2	1- SystemException.ILLEGAL_VALUE is thrown	

5.3.2.2 Method getTheAdminFileView(javacard.framework.AID aid, byte event)

Test Area Reference: Api_4_Afb_Gtafob.

5.3.2.2.1 Conformance requirement:

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

```
public static AdminFileView getTheAdminFileView(javacard.framework.AID aid, byte event)
                throws NullPointerException,
                       javacard.framework.SystemException
```

5.3.2.2.1.1 Normal execution

- CRRN1: returns a reference to class which implements the AdminFileView interface on an ADF file system defined by is AID.
- CRRN2: returns null if the ADF with the AID does not exist.

- CRRN3: returns null if one of the javacard.framework.Applet.register(..) method has not previously invoked by the applet invoking this method or the filesystem server does not exist or the filesystem server returns null.
- CRRN4: After a successful invocation of the method the ADF is the currently selected file.
- CRRN5: A separate and independent file context shall be associated with each and every AdminFileView object: the operation performed on files in a given AdminFileView object shall not affect the file context associated with any other AdminFileView object.This context can be transient or persistent depending on what was required by the Applet during the creation of the AdminFileView object.
- CRRN6: It is not possible to access files which are not located under the ADF.

5.3.2.2.1.2 Parameter errors

- CRRP1: If event is not one of the following values JCSystem.NOT_A_TRANSIENT_OBJECT, JCSystem.CLEAR_ON_DESELECT, or JCSystem.CLEAR_ON_RESET a SystemException is thrown. The value of the SystemException shall be SystemException.ILLEGAL_VALUE.
- CRRP2: If the AID is null a NullPointerException shall be thrown.

5.3.2.2.1.3 Context errors

- CRRC1: If event is JCSystem.CLEAR_ON_RESET or JCSystem.CLEAR_ON_DESELECT and not enough transient memory space is available a SystemException shall be thrown with reason code SystemException.NO_TRANSIENT_SPACE.
- CRRC2: If event is JCSystem.CLEAR_ON_DESELECT and the applet is not a currently selected applet a SystemException with reason code SystemException.ILLEGAL_TRANSIENT shall be thrown.

5.3.2.2.2 Test area files

Test Source: Test_ Api_4_Afb_Gtafob.java.

Test Applet: Api_4_Afb_Gtafob.java.

Cap File: Api_4_Afb_Gtafob.cap.

5.3.2.2.3 Test coverage

CRR number	Test case number
N1	1 to 3
N2	1
N3	1
N4	2
N5	3
N6	2
P1	7
P2	8
C1	5, 6 Testable only if available transient space is lower than 32767
C2	4

5.3.2.2.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Method returns null** 1- Install Applet1 with full access rights on the UICC file system 2- Invoke the method getTheAdminFileView before the javacard.framework.Applet.register(..) method invocation 3- Envelope menu selection is sent to the UICC 4- Invoke the method getTheAdminFileView() with AID = unknown ADF AID	 2- returns null 3- applet is triggered 4- returns null	
2	**Normal execution** 1- Envelope menu selection is sent to the UICC Invoke the method getTheAdminFileView() with AID = ADF1 with the event JCSystem.NOT_A_TRANSIENT_OBJECT and stores the result in a class variable FV1 Applet1 calls FV1.status() command Select DFTest using FV1 Select EFTARU using FV1Read first 3 bytes using FV1 Reset Terminal profile 2 - Envelope menu selection is sent to the UICC Applet1 calls FV1.status() command Read first 3 bytes using FV1 Applet1 calls FV1.select(EFRFU1) Invoke the method getTheAdminFileView() with the event JCSystem.CLEAR_ON_RESET and stores the result in a class variable FV2 Applet1 calls FV2.status() command Select DFTest using FV2 Select EFTARU using FV2 Read first 3 bytes using FV2 Reset Terminal profile 4 - Envelope menu selection is sent to the UICC Applet1 calls FV2.status() command Read first 3 bytes using FV2 Applet1 calls FV2.select(EFRFU1) 5- Select the Applet by AID Invoke the method getTheAdminFileView() with AID = ADF1 with the event:	1- Applet1 is triggered No Exception shall be thrown Current selected DF is ADF1 Expected value is {FF FF FF} 2- Applet1 is triggered Current selected DF is DF_{Test} Expected value is {FF FF FF} UICCExceptionUICCException.FILE_NOT_FOUND is thrown No exception shall be thrown Current selected DF is the ADF1 Expected value is {FF FF FF} 4- Applet1 is triggered Current selected DF is the ADF1 UICCException. NO_EF_SELECTED UICCException.FILE_NOT_FOUND is thrown 5- Applet1 is selected	

Id	Description	API Expectation	APDU Expectation
	JCSystem.CLEAR_ON_DESELECT and stores the result in a class variable FV3 Applet1 calls FV3.status() command Select DFTest using FV3 Select EFTARU using FV3 Read first 3 bytes using FV3 6- Select the Applet by AID Applet1 calls FV3.status() command Read first 3 bytes using FV3 Applet1 calls FV3.select(EFRFU1)	No Exception shall be thrown Current selected DF is ADF1 Expected value is {FF FF FF} 6- Applet1 is selected Current selected DF is ADF1 UICCException.NO_EF_SELECTED shall be thrown UICCException.FILE_NOT_FOUND shall be thrown	
3	**FileView context independency** 1- Envelope menu selection is sent to the UICC 2- Check that previous fileviews are different (FV1 != FV2 != FV3) 3- Select DFTest/EFLARU using FV1 4- Select DFTest/EFCARU using FV2 5- Select DFTest/EFCARU using FV3 6- Read record number 1 using FV1 (in absolute mode) 7- Read record number 2 using FV2 (in absolute mode)	1- Applet1 is triggered 3- No exception shall be thrown 4- No exception shall be thrown 5- An exception shall be thrown 6- Expected value is “55 55 55 55” 7- Expected value is “AA AA AA”	
4	**ILLEGAL_TRANSIENT SystemException** 1- Envelope menu selection is sent to the UICC 2- Applet1 calls getTheAdminFileView() method with the event JCSystem.CLEAR_ON_DESELECT	1- Applet1 is triggered 2- SystemException. ILLEGAL_TRANSIENT is thrown	
5	**NO_TRANSIENT_SPACE SystemException with CLEAR_ON_RESET FileView object** 1 Get the available transient memory space using method length = JCSystem.getAvailableMemory(MEMORY_TYPE_TRANSIENT_RESET) 2- If length < 32767, (test case could be performed) 2.1- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR_ON_RESET) 2.2- Applet calls method getTheAdminFileView() with AID = ADF1 with event JCSystem.CLEAR_ON_RESET	1- No Exception shall be thrown 2.1- No Exception shall be thrown 2.2- SystemException.NO_TRANSIENT_SPACE is thrown	

Id	Description	API Expectation	APDU Expectation
6	**NO_TRANSIENT_SPACE SystemException with CLEAR_ON_DESELECT FileView object** 1 - Select the Applet by AID 2- Get the available transient memory space using method length = JCSystem.getAvailableMemory(MEMORY_TYPE_TRANSIENT_DESELECT) 3- If length < 32767, (test case could be performed) 3.1- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR_ON_DESELECT) } 3.2- Applet calls method getTheAdminFileView() with AID = ADF1 with event: JCSystem.CLEAR_ON_DESELECT 4- Reset	1- Applet1 is triggered 2- No Exception shall be thrown 3.1- No Exception shall be thrown 3.2- SystemException. NO_TRANSIENT_SPACE is thrown	
7	**ILLEGAL_VALUE SystemException** 1- Invoke the method getTheAdminFileView() with every event codes except: 0, 1, 2	1- SystemException.ILLEGAL_VALUE is thrown	
8	**NullPointerException** Invoke the method getTheAdminFileView() with AID = NULL with event: 1 - JCSystem.CLEAR_ON_RESET	1- Shall be thrown java.lang.NullPointerException	

5.3.2.3 Method getTheAdminFileView(byte[] buffer, short bOffset, short bLength, byte event)

Test Area Reference: Api_4_Afb_Gtaf_Bsbb.

5.3.2.3.1 Conformance requirement:

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

```
public static AdminFileView getTheAdminFileView(byte[] buffer, short bOffset, short bLength, byte
event)
                throws NullPointerException,
                       javacard.framework.SystemException,
                       ArrayIndexOutOfBoundException
```

5.3.2.3.1.1 Normal execution

- CRRN1: returns a reference to class which implements the AdminFileView interface on an ADF file system defined by its AID.
- CRRN2: returns null if the ADF with the full AID given in the buffer does not exist.
- CRRN3: returns null if one of the javacard.framework.Applet.register(..) method has not previously invoked by the applet invoking this method or the filesystem server does not exist or the filesystem server returns null.
- CRRN4: After a successful invocation of the method the ADF is the currently selected file.

- CRRN5: A separate and independent file context shall be associated with each and every AdminFileView object: the operation performed on files in a given FileView object shall not affect the file context associated with any other AdminFileView object. This context can be transient or persistent depending on what was required by the Applet during the creation of the AdminFileView object.

- CRRN6: It is not possible to access files which are not located under the ADF.

5.3.2.3.1.2 Parameters error

- CRRP1: If event is not one of the following values JCSystem.NOT_A_TRANSIENT_OBJECT, JCSystem.CLEAR_ON_DESELECT, or JCSystem.CLEAR_ON_RESET a SystemException is thrown. The value of the SystemException shall be SystemException.ILLEGAL_VALUE.

- CRRP2: If the buffer is null a NullPointerException shall be thrown.

- CRRP3: if bLength is less then 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.

- CRRP4: if bOffset plus bLength is greater than the length of the array buffer.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

- CRRP5: if bLength is not in the range of 5 - 16 bytes a SystemException.ILLEGAL_VALUE shall be thrown.

5.3.2.3.1.3 Context errors

- CRRC1: If event is JCSystem.CLEAR_ON_RESET or JCSystem.CLEAR_ON_DESELECT and not enough transient memory space is available a SystemException shall be thrown with reason code SystemException.NO_TRANSIENT_SPACE.

- CRRC2: If event is JCSystem.CLEAR_ON_DESELECT and the applet is not a currently selected applet a SystemException with reason code SystemException.ILLEGAL_TRANSIENT shall be thrown.

5.3.2.3.2 Test area files

Test Source: Test_ Api_4_Afb_Gtaf_Bsbb.java.

Test Applet: Api_4_Afb_Gtaf_Bsbb.java.

Cap File: Api_4_Afb_Gtaf_Bsbb.cap.

5.3.2.3.3 Test coverage

CRR number	Test case number
N1	2
N2	3
N3	1
N4	2
N5	11, 12
N6	5
P1	2
P2	5
P3	7
P4	8
P5	9
C1	11, 12 Testable only if available transient space is lower than 32767
C2	6

5.3.2.3.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Method returns null** 1- Install Applet1 with full access rights on the UICC file system 2- Invoke the method getTheFileView before the javacard.framework.Applet.register(..) method invocation Invoke the method getTheAdminFileView() with buffer[20] = {ADF1,…} bOffset= 0 bLength= 16 3- Envelope menu selection is sent to the UICC 4- Invoke the method getTheAdminFileView before the javacard.framework.Applet.register(..) method invocation Invoke the method getTheAdminFileView() with buffer[] = unknown aid	 2- returns null 3- Applet is triggered 4- returns null	
2	**Normal execution** 1- Envelope menu selection is sent to the UICC Invoke the method getTheAdminFileView() with buffer[20] = {ADF1,…} bOffset= 0 bLength= 16 JCSystem.NOT_A_TRANSIENT_OBJECT and stores the result in a class variable FV1 Applet1 calls FV1.status() command Select DFTest using FV1 Select EFTARU using FV1 Read first 3 bytes using FV1 Reset Terminal profile 2 - Envelope menu selection is sent to the UICC Applet1 calls FV1.status() command Read first 3 bytes using FV1 Applet1 calls FV1.select(EFRFU1) Invoke the method getTheAdminFileView() with the event JCSystem.CLEAR_ON_RESET and stores the result in a class variable FV2 Applet1 calls FV2.status() command Select DFTest using FV2 Select EFTARU using FV2 Read first 3 bytes using FV2 Reset Terminal profile 4 - Envelope menu selection is sent to the UICC Applet1 calls FV2.status() command	 1- Applet1 is triggered No Exception shall be thrown Current selected DF is ADF1 Expected value is {FF FF FF} 2- Applet1 is triggered Current selected DF is DFTest Expected value is {FF FF FF} UICCExceptionUICCException.FILE_NOT_FOUND is thrown No exception shall be thrown Current selected DF is ADF1 Expected value is {FF FF FF} 4- Applet1 is triggered Current selected DF is ADF1	

Id	Description	API Expectation	APDU Expectation
	Read first 3 bytes using FV2	UICCException.NO_EF_SELECTED.	
	Applet1 calls FV2.select(EFRFU1)	UICCException.FILE_NOT_FOUND is thrown	
	5- Select the Applet by AID Invoke the method getTheAdminFileView() with AID = ADF1 with buffer[20] = {ADF1,…} bOffset= 0 bLength= 16 the event:JCSystem.CLEAR_ON_DESELECT and stores the result in a class variable FV3	5- Applet1 is selected No Exception shall be thrown	
	Applet1 calls FV3.status() command Select DFTest using FV3 Select EFTARU using FV3 Read first 3 bytes using FV3	Current selected DF is ADF1 Expected value is {FF FF FF}	
	6- Select the Applet by AID Applet1 calls FV3.status() command	6- Applet1 is selected Current selected DF is ADF1	
	Read first 3 bytes using FV3	UICCException.NO_EF_SELECTED.	
	Applet1 calls FV3.select(EFRFU1)	UICCException.FILE_NOT_FOUND is thrown	
3	**FileView context independency** 1- Envelope menu selection is sent to the UICC 2- Check that previous fileviews are different (FV1 != FV2 != FV3) 3- Select DFTest/EFLARU using FV1 4- Select DFTest/EFCARU using FV2 5- Select DFTest/EFCARU using FV3 6- Read record number 1 using FV1 (in absolute mode) 7- Read record number 2 using FV2 (in absolute mode)	1- Applet1 is triggered 3- No exception shall be thrown 4- No exception shall be thrown 5- An exception shall be thrown 6- Expected value is “55 55 55 55” 7- Expected value is “AA AA AA”	
4	**ILLEGAL_TRANSIENT SystemException** 1- Envelope menu selection is sent to the UICC 2- Applet1 calls getTheAdminFileView() method with buffer[20] = {ADF1,…} bOffset= 0 bLength= 16 with the event JCSystem.CLEAR_ON_DESELECT	1- Applet1 is triggered 2- SystemException. ILLEGAL_TRANSIENT is thrown	

Id	Description	API Expectation	APDU Expectation
5	**NO_TRANSIENT_SPACE SystemException with CLEAR_ON_RESET FileView object** 1- Get the available transient memory space using method length = JCSystem.getAvailableMemory(MEMORY_TYP_TRANSIENT_RESET) 2- If length < 32767, (test case could be performed) 2.1- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR_ON_RESET) 2.2- Applet calls method getTheAdminFileView() with buffer[20] = {ADF1,…} bOffset= 0 bLength= 16 with the event JCSystem.CLEAR_ON_RESET	1- No Exception shall be thrown 2.1- No Exception shall be thrown 2.2- SystemException. NO_TRANSIENT_SPACE is thrown	
6	**NO_TRANSIENT_SPACE SystemException with CLEAR_ON_DESELECT FileView object** 1 - Select the Applet by AID 2- Get the available transient memory space using method length = JCSystem.getAvailableMemory(MEMORY_TYPE_TRANSIENT_DESELECT) 3- If length < 32767, (test case could be performed) 3.1- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR_ON_DESELECT) } 3.2- Applet calls method getTheAdminFileView() with buffer[20] = {ADF1,…} bOffset= 0 bLength= 16 with event: JCSystem.CLEAR_ON_DESELECT 4- Reset	1- Applet1 is triggered 2- No Exception shall be thrown 3.1- No Exception shall be thrown 3.2- SystemException. NO_TRANSIENT_SPACE is thrown	
7	**ILLEGAL_VALUE SystemException** 1- Invoke the method getTheAdminFileView() with every event codes except: 0, 1, 2	1- SystemException.ILLEGAL_VALUE is thrown	
8	**NullPointerException** Invoke the method getTheAdminFileView() with buffer[20] = null bOffset= 0 bLength= 16 with event: 1 - JCSystem.CLEAR_ON_RESET	1- Shall be thrown java.lang.NullPointerException	

Id	Description	API Expectation	APDU Expectation
9	**ArrayIndexOutOfBoundsException** 1-Envelope menu selection is sent to the UICC Invoke the method getTheAdminFileView() with buffer[20] = {ADF1,…} bOffset= 5 bLength= 16 event =JCSystem. CLEAR_ON_RESET Invoke the method getTheAdminFileView() with buffer[20] = {ADF1,…} bOffset= -1 bLength= 16 event =JCSystem. CLEAR_ON_RESET	1- Applet1 is triggered Shall be thrown ArrayIndexOutOfBoundsException Shall be thrown ArrayIndexOutOfBoundsException	
10	**SystemException.ILLEGAL_VALUE** 1-Envelope menu selection is sent to the UICC Invoke the method getTheAdminFileView() with buffer[20] = {ADF1,…} bOffset= 0 bLength= 4 event =JCSystem. CLEAR_ON_RESET Invoke the method getTheAdminFileView() with buffer[20] = {ADF1,…} bOffset= 0 bLength= 17 event =JCSystem. CLEAR_ON_RESET	1- Applet1 is triggered SystemException.ILLEGAL_VALUE shall be thrown SystemException.ILLEGAL_VALUE shall be thrown	

5.3.3 Class AdminException

5.3.3.1 Constructor

Test Area Reference: Api_4_Aex_Coor.

5.3.3.1.1 Conformance requirement

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

```
public AdminException(short reason)
```

5.3.3.1.1.1 Normal execution

- CRRN1: Constructs an AdminException with the specified reason.

5.3.3.1.1.2 Parameter errors

No requirements.

5.3.3.1.1.3 Context errors

No requirements.

5.3.3.1.2 Test area files

Test Source: Test_Api_4_Aex_Coor.java.

Test Applet: Api_4_Aex_Coor_1.java.

Cap File: Api_4_Aex_Coor.cap.

5.3.3.1.3 Test coverage

CRR number	Test case number
N1	1

5.3.3.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**AdminException with the specified reason** (The reason shall set with setReason and compare the Exception with getReason)	Reason (specified)	

5.3.3.2 Method throwIt

Test Area Reference: Api_4_Aex_Thit.

5.3.3.2.1 Conformance requirement

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

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

5.3.3.2.1.1 Normal execution

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

5.3.3.2.1.2 Parameter errors

No requirements.

5.3.3.2.1.3 Context errors

No requirements.

5.3.3.2.2 Test area files

Test Source: Test_Api_4_Aex_Thit.java.

Test Applet: Api_4_Aex_Thit_1.java.

Cap File: Api_4_Aex_Thit.cap.

5.3.3.2.3 Test coverage

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

5.3.3.2.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	Throws the JCRE instance of AdminException with the specified reason	Reason = 0	
2	Throws the JCRE instance of AdminException with the specified reason	Reason = 1	
3	Throws the JCRE instance of AdminException with the specified reason	Reason = 0xA55A	
4	AdminException extends javacard.framework.CardRuntimeException	Reason = 0	
5	AdminException extends javacard.framework.CardRuntimeException	Reason = 1	
6	AdminException extends javacard.framework.CardRuntimeException	Reason = 0xA55A	

5.3.3.3 Reason Codes

Test Area Reference: Api_4_Aex_Cons.

5.3.3.3.1 Conformance Requirement:

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

5.3.3.3.1.1 Normal execution

- CRRN1: The Constants of the class AdminException shall all have the same name and value defined in the ETSI TS 102 241 [9].
- CRRN2: Constructs AdminException an Exception with the specified reason.

5.3.3.3.1.2 Parameter errors

No requirements.

5.3.3.3.1.3 Context errors

No requirements.

5.3.3.3.2 Test area files

None.

5.3.3.3.3 Test Coverage

CRR number	Test case number
N1 & N2	The constants in Java are resolved at compilation time, therefore a runtime test is not useful. No test of constants will be performed

5.3.3.3.4 Test Procedure

None.

5.4 Package uicc.system

5.4.1 Class HandlerBuilder

5.4.1.1 Method buildTLVHandler(byte type, short capacity)

Test Area Reference: Api_3_Hdb_Bthdbs.

5.4.1.1.1 Conformance requirement

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

```
public static ViewHandler buildTLVHandler(byte type,
                                          short capacity)
                                   throws java.lang.ArrayIndexOutOfBoundsException,
                                          javacard.framework.SystemException
```

5.4.1.1.1.1 Normal execution

- CRRN1: Allocates a TLVHandler with an internal buffer of length capacity.

5.4.1.1.1.2 Parameter errors

- CRRP1: If the type parameter does not match with the predefined values, a javacard.framework.SystemException is thrown with ILLEGAL_VALUE reason code.
- CRRP2: If capacity is negative, a javacard.framework.SystemException is thrown with ILLEGAL_VALUE reason code.

5.4.1.1.1.3 Context errors

- CRRC1: If there are not enough resources in the card to allocate the handler, a javacard.framework.SystemException is thrown with NO_RESOURCE reason code.

5.4.1.1.2 Test area files

Test Source: Test_Api_3_Hdb_Bthdbs.java.

Test Applet: Api_3_Hdb_Bthdbs_1.java.

Cap File: Api_3_hdb_bthdbs.cap.

5.4.1.1.3 Test coverage

CRR number	Test case number
N1	1, 2
P1	4, 5
P2	3
C1	Not testable

5.4.1.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation

1	**Call buildTLVHandler() method with EDIT_HANDLER type**
	
	`Type = EDIT_HANDLER`
	`Capacity = (short)10`
	`Check the created object is not null`
		No exception shall be thrown	

2	**Call buildTLVHandler() method with BER_EDIT_HANDLER type**
	
	`Type = BER_EDIT_HANDLER`
	`Capacity = (short)10`
	`Check the created object is not null`
		No exception shall be thrown	

3	**Negative capacity**
	
	`Type = EDIT_HANDLER`
	`Capacity = (short)-10`
		A javacard.framework.SystemException is thrown with ILLEGAL_VALUE reason code..	

4	**Type does not match with predefined values**
	
	`Type = (byte)3`
	`Capacity = (short)10`
		javacard.framework.SystemException shall be thrown with ILLEGAL_VALUE reason code.	

5	**Type does not match with predefined values**
	
	`Type = (byte)0`
	`Capacity = (short)10`
		javacard.framework.SystemException shall be thrown with ILLEGAL_VALUE reason code.	

5.4.1.2 Method buildTLVHandler(byte type, short capacity, byte[] buffer , short offset, short length)

Test Area Reference: Api_3_Hdb_Bthdbs_Bss.

5.4.1.2.1 Conformance requirement

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

```
public static ViewHandler buildTLVHandler(byte type,
                                          short capacity,
                                          byte[] buffer,
                                          short offset,
                                          short length)
                                   throws java.lang.ArrayIndexOutOfBoundsException,
                                          javacard.framework.SystemException,
                                          java.lang.NullPointerException
```

5.4.1.2.1.1 Normal execution

- CRRN1: Allocates a TLVHandler with an internal buffer of length capacity.
- CRRN2: Copies the buffer content to an internal buffer of the TLVHandler starting at bOffset.
- CRRN3: The internal buffer shall be at least bLength long.

5.4.1.2.1.2 Parameter errors

- CRRP1: If the type does not match with the predefined values, a javacard.framework.SystemException is thrown with ILLEGAL_VALUE reason code.
- CRRP2: If buffer is null, a java.lang.NullPointerException is thrown.

- CRRP3: If bOffset would cause access outside array bounds, an java.lang.ArrayIndexOutOfBoundsException is thrown.
- CRRP4: if bLength is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.
- CRRP5: If capacity is negative, a javacard.framework.SystemException is thrown with ILLEGAL_VALUE reason code.
- CRRP6: If bOffset+bLength is greater than the length of the buffer, a java.lang.ArrayIndexOutOfBoundsException is thrown.

5.4.1.2.1.3 Context errors

- CRRC1: If there are not enough resources in the card to allocate the handler, a javacard.framework.SystemException is thrown with NO_RESOURCE reason code.

5.4.1.2.2 Test area files

Test Source: Test_Api_3_Hdb_Bthdbs_Bss.java.

Test Applet: Api_3_Hdb_Bthdbs_Bss_1.java.

Cap File: Api_3_hdb_bthdbs_bss.cap.

5.4.1.2.3 Test coverage

CRR number	Test case number
N1	1, 2
N2	6
N3	1, 2, 6
P1	4, 5
P2	7
P3	8, 9
P4	10
P5	3
P6	11
C1	Not testable

5.4.1.2.4 Test procedure

Id	Description	API Expectation	APDU Expectation

1	**Call buildTLVHandler() method with EDIT_HANDLER type**
	
	`Type = EDIT_HANDLER`
	`Capacity = (short)10`
	`Buffer[10]`
	`Offset = (short)0`
	`Length = (short)0`
	`Check the created object is not null`
		No exception shall be thrown	

2	**Call buildTLVHandler() method with BER_EDIT_HANDLER type**
	
	`Type = BER_EDIT_HANDLER`
	`Capacity = (short)10`
	`Buffer[10]`
	`Offset = (short)0`
	`Length = (short)0`
	`Check the created object is not null`
		No exception shall be thrown	

3	**Negative capacity**
	
	`Type = EDIT_HANDLER`
	`Capacity = (short)-10`
	`Buffer[10]`
	`Offset = (short)0`
	`Length = (short)5`
		A javacard.framework.SystemException is thrown with ILLEGAL_VALUE reason code..	

Id	Description	API Expectation	APDU Expectation
4	**Type does not match with predefined values** Type = (byte)0 Capacity = (short)10 Buffer[10] Offset = (short)0 Length = (short)5	A javacard.framework.SystemException shall be thrown with ILLEGAL_VALUE reason code	
5	**Type does not match with predefined values** Type = (byte)3 Capacity = (short)10 Buffer[10] Offset = (short)0 Length = (short)5	A javacard.framework.SystemException shall be thrown with ILLEGAL_VALUE reason code	
6	**Internal Buffer starts at bOffset** Type = EDIT_HANDLER Capacity = (short)10 Buffer[10] Offset = (short)4 Length = (short)5 Check the internal buffer of the TLVHandler starts with bOffset data.	No exception shall be thrown	
7	**Buffer is null** Type = EDIT_HANDLER Capacity = (short)10 Buffer[] = null Offset = (short)0 Length = (short)5	A java.lang.NullPointerException shall be thrown.	
8	**bOffset > Buffer Length** Type = EDIT_HANDLER Capacity = (short)10 Buffer[10] Offset = (short)11 Length = (short)0	A java.lang.ArrayIndexOutOfBoundsException shall be thrown	
9	**bOffset < 0** Type = EDIT_HANDLER Capacity: (short)10 Buffer[10] Offset = (short)-1 Length = (short)0	A java.lang.ArrayIndexOutOfBoundsException shall be thrown	
10	**bLength < 0** Type = EDIT_HANDLER Capacity = (short)10 Buffer[10] Offset = (short)0 Length = (short)-1	A java.lang.ArrayIndexOutOfBoundsException shall be thrown	
11	**bOffset+bLength > buffer length** Type = EDIT_HANDLER Capacity = (short)10 Buffer[10] Offset = (short)7 Length = (short)8	A java.lang.ArrayIndexOutOfBoundsException shall be thrown	

5.4.2 Interface UICCPlatform

5.4.2.1 Method getTheVolatileByteArray

Test Area Reference: Api_3_Upf_Gvba.

5.4.2.1.1 Conformance requirement

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

```
public static byte[] getTheVolatileByteArray()
```

5.4.2.1.1.1 Normal execution

- CRRN1: Returns the instance of the volatile byte array designated by the JCRE as global array. The byte array length shall be at least equal to 256 bytes.

5.4.2.1.1.2 Parameter errors

No requirement.

5.4.2.1.1.3 Context errors

- CRRC1: If the method is invoked from a context which is not the currently selected applet or the currently triggered applet i.e. the context of the applet that treats the current APDU or the context of the applet that has been triggered by the current APDU, a java.lang.SecurityException is thrown.
- CRRC2: A reference to this byte array cannot be stored in class variables or instance variables or array components.

5.4.2.1.2 Test area files

Test Source: Test_Api_3_Upf_Gvba.java.

Test Applet: Api_3_Upf_Gvba_1.java.

Api_ShareableInterface.java.

Api_GetShareableClientApplet.java.

Cap File: Api_3_upf_gvba.cap.

Api_3_upf_gvba2.cap.

5.4.2.1.3 Test coverage

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

5.4.2.1.4 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Call getTheVolatileByteArray() method and store it in a local variable** 1- Trigger the applet and check the returned byte array length is at least equal to 256 bytes. 2- Select the applet and check the returned byte array length is at least equal to 256 bytes.	1- No exception shall be thrown. 2- No exception shall be thrown.	
2	**Method invoked from a different context** By the way of the Shareable Interface, call the getTheVolatileByteArray() method through another applet.	A java.lang.SecurityException shall be thrown.	
3	**Store the instance in a class variable**	A java.lang.SecurityException shall be thrown.	
4	**Store the instance in an instance variable**	A java.lang.SecurityException shall be thrown.	
5	**Store the instance in an array component**	A java.lang.SecurityException shall be thrown.	

5.5 CAT Runtime Environment

5.5.1 Minimum Handler Availability

This test area tests the rules that define the minimum requirements for the availability of the system handlers.

5.5.1.1 ProactiveHandler

Test Area Reference: Cre_Mha_Pahd.

5.5.1.1.1 Conformance requirement

5.5.1.1.1.1 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_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_BY_NAA

  EVENT_PROFILE_DOWNLOAD

  EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

  EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS

  EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE

  EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED

  EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION

  EVENT_PROACTIVE_HANDLER_AVAILABLE

  EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE

  EVENT_EVENT_BROWSING_STATUS

EVENT_EXTERNAL_FILE_UPDATE

- CRRN2: A ProactiveHandler is considered available when no HANDLER_NOT_AVAILABLE *ToolkitException* is thrown when the corresponding *getTheHandler()* method is called or a method of the handler is called.
- CRRN3: When available the *ProactiveHandler* shall remain available until the termination of the *processToolkit()* method.
- CRRN4: If a proactive command is pending the *ProactiveHandler* may not be available.

5.5.1.1.1.2 Parameter errors

No requirements.

5.5.1.1.1.3 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_ATR

  EVENT_APPLICATION_DESELECT

- CRRC2: The ProactiveHandler shall not be available if the Terminal Profile command has not yet been processed by the CAT Runtime Environment
- CRRC3: The ProactiveHandler shall not be available if the *getTheHandler()* method is not called, directly or indirectly, from the applet's *processToolkit*() method.

5.5.1.1.2 Test area files

Test Source: Test_Cre_Mha_Pahd.java.

Test Applet: Cre_Mha_Pahd_1.java.

Cre_Mha_Pahd_2.java.

Cre_Mha_Pahd_3.java.

Cap File: Cre_Mha_Pahd.cap.

5.5.1.1.3 Test coverage

CRR Number	Test Case Number
CRRN1	2 to 23, 45, 46
CRRN2	1 to 22,45,46
CRRN3	2 to 22, 45, 46
CRRN4	Not testable
CRRC1	1, 24
CRRC2	25 to 44 or also tested in TestCases 25 to 44 in Cre_Mha_Prhd
CRRC3	47

5.5.1.1.4 Test procedure

Id	Description	API /Framework Expectation	APDU Expectation
1	**Applets registration to all events and Proactive Handler availability with EVENT_FIRST_COMMAND_AFTER_ATR** Applet1 is registered to all events defined in TS 102 241 [9] except to EVENT_PROACTIVE_HANDLER_AVAILABLE. Using the methods initMenuEntry() for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION, allocateServiceIdentifier() for EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION and setEventList() for the rest of the events. Applet1 is registered on EVENT_EXTERNAL_FILE_UPDATE on update of EF_{TARU} of the UICC file system Applet2 is registered to all events defined in TS 102 241 [9], except to EVENT_PROACTIVE_HANDLER_AVAILABLE and EVENT_CALL_CONTROL_BY_NAA. 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 on EVENT_EXTERNAL_FILE_UPDATE on update of EF_{TARU} of the UICC file system 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_ATR. 3- Applet2 gets the Proactive Handler Applet2 is deregistered to EVENT_FIRST_COMMAND_AFTER_ATR.	1- Applet1 is triggered by EVENT_FIRST_COMMAND_AFTER_ATR 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet1 finalizes Applet2 is triggered by EVENT_FIRST_COMMAND_AFTER_ATR 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes	
2	**Proactive Handler availability with EVENT_PROFILE_DOWNLOAD** 1- Terminal Profile command is sent to the UICC without the facility of SET UP 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 Applet2 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
3	**Proactive Handler availability with EVENT_MENU_SELECTION_HELP_REQUEST** Perform UICC initialization with all the facilities supported, without facility SET_UP_EVENT_LIST 1- Envelope menu selection with help request is sent to the UICC 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 UICC 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes	
5	**Proactive Handler availability with EVENT_TIMER_EXPIRATION** 1- Timer Id =1 Envelope Timer Expiration is sent to the UICC 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes	
6	**Proactive Handler availability with EVENT_CALL_CONTROL_BY_NAA** 1- Envelope call control by NAA is sent to the UICC 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes	
7	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_MT_CALL** 1- Envelope event download mt call is sent to the UICC 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 Applet2 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
8	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_CALL_CONNECTED** 1- Envelope event download call connected is sent to the UICC 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 Applet2 finalizes	
9	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED** 1- Envelope event download call disconnected is sent to the UICC 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 Applet2 finalizes	
10	**Applets triggering with EVENT_EVENT_DOWNLOAD_LOCATION_STATUS** 1- Envelope event download location status is sent to the UICC 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 Applet2 finalizes	
11	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_USER_ACTIVITY** 1- Envelope event download user activity is sent to the UICC 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 Applet2 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
12	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE** 1- Envelope event download idle screen available is sent to the UICC 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 Applet2 finalizes	
13	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS** 1- Envelope event download card reader status is sent to the UICC 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 Applet2 finalizes	
14	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION** 1- Envelope event download language selection is sent to the UICC 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 Applet2 finalizes	
15	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION** 1- Envelope event download browser termination is sent to the UICC 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 Applet2 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
16	**Proactive Handler availability with EVENT_STATUS_COMMAND** 1- Status command is sent to the UICC 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 Applet2 finalizes	
17	**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 UICC, 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
18	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS** 1- An Envelope Event Download Channel Status is sent to the UICC, with ChannelId=01 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes	
19	**Proactive Handler availability with UNRECOGNIZED_ENVELOPE** 1- An unrecognized Envelope (BER TLV Tag unrecognized) is sent to the UICC 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 Applet2 finalizes	
20	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE** 1- An envelope event download access technology change is sent to the UICC 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 Applet2 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
21	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED**		
	1- An envelope event download display parameter changed is sent to the UICC	1- Applet1 is triggered	
	2- Applet1 gets the Proactive Handler	2- No exception is thrown	
		3- Applet1 sends a DECLARE SERVICE (add) proactive command with its service identifier	3- DECLARE SERVICE (add) proactive command is fetched
		Applet1 finalizes Applet2 is triggered	Successful TERMINAL RESPONSE is issued
	4-Applet2 gets the Proactive Handler	4- No exception is thrown	
		Applet2 finalizes	
22	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION**		
	1- An envelope event download local connection is sent to the UICC, with the allocated service identifier	1- Applet1 is triggered	
	2- Applet1 gets the Proactive Handler	2- No exception is thrown	
		Applet1 finalizes	
23	**Proactive Handler availability with EVENT_PROACTIVE_HANDLER_AVAILABLE**		
	1- Envelope event download call connected is sent to the UICC	1- Applet1 is triggered and registers to EVENT_PROACTIVE_HANDLER_AVAILABLE	
		Applet1 finalizes Applet2 is triggered, registers to EVENT_PROACTIVE_HANDLER_AVAILABLE and sends a Display Test proactive command	
	2-The display Text proactive command is fetch and the terminal response is sent	2- Applet1 is triggered by EVENT_PROACTIVE_HANDLER_AVAILABLE	
	3-Applet1 gets the Proactive Handler	3- No exception is thrown	
		Applet1 finalizes Applet2 is triggered by EVENT_PROACTIVE_HANDLER_AVAILABLE	
	4-Applet2 gets the Proactive Handler	4- No exception is thrown	
		Applet2 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
24	**Proactive Handler availability with EVENT_APPLICATION_DESELECT** 1- Select for activation ADF1 2- Select for termination ADF1 3- Applet1 gets the Proactive Handler Applet1 deregisters to EVENT_APPLICATION_DESELECT. 4-Applet2 gets the Proactive Handler Applet2 deregisters to EVENT_APPLICATION_DESELECT	2- Applet1 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
25	**The ProactiveHandler is not available before the Terminal Profile with EVENT_MENU SELECTION_HELP_REQUEST** 1- Reset the card without sending the Terminal Profile 2- Envelope menu selection with help request is sent to the UICC 3- Applet1 gets the Proactive Handler	2- Applet1 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
26	**The ProactiveHandler is not available before the Terminal Profile with EVENT_MENU SELECTION** 1- Envelope menu selection is sent to the UICC 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
27	**The ProactiveHandler is not available before the Terminal Profile with EVENT_TIMER_EXPIRATION** 1- Timer Id =1 Envelope Timer Expiration is sent to the UICC 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
28	**The ProactiveHandler is not available before the Terminal Profile with EVENT_CALL_CONTROL_BY_NAA** 1- Envelope call control by NAA is sent to the UICC 2- Applet1 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet1 finalizes	
29	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_MT_CALL** 1- Envelope event download mt call is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
30	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_CALL_CONNECTED** 1- Envelope event download call connected is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
31	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED** 1- Envelope event download call disconnected is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
32	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_LOCATION_STATUS** 1- Envelope event download location status is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
33	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_USER_ACTIVITY** 1- Envelope event download user activity is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
34	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE** 1- Envelope event download idle screen available is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
35	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS** 1- Envelope event download card reader status is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
36	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION** 1- Envelope event download language selection is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
37	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION** 1- Envelope event download browser termination is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
38	**The ProactiveHandler is not available before the Terminal Profile with EVENT_STATUS_COMMAND** 1- Status command is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
39	**The ProactiveHandler is not available before the Terminal Profile with UNRECOGNIZED_ENVELOPE** 1- An unrecognized Envelope (BER TLV Tag unrecognized) is sent to the UICC 2- Applet1 gets the Proactive Handler 3-Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
40	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE** 1- An envelope event download access technology change is sent to the UICC 2- Applet1 gets the Proactive Handler 3-Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
41	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED** 1- An envelope event download display parameter changed is sent to the UICC 2- Applet1 gets the Proactive Handler 3-Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 registers to EVENT_PROACTIVE_HANDLER_AVAILABLE then finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 registers to EVENT_PROACTIVE_HANDLER_AVAILABLE then finalizes	
42	**The ProactiveHandler is not available before the Terminal Profile with EVENT_PROACTIVE_HANDLER_AVAILABLE**	Applet1 and Applet2 are not triggered by EVENT_PROACTIVE_HANDLER_AVAILABLE	

Id	Description	API /Framework Expectation	APDU Expectation
43	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE** 1- Envelope event download network search mode change is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 deregisters from EVENT_PROACTIVE_HANDLER_AVAILABLE Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 deregisters from EVENT_PROACTIVE_HANDLER_AVAILABLE Applet2 finalizes	
44	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_BROWSING STATUS** 1- Envelope event download browsing status is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	

Id	Description	API /Framework Expectation	APDU Expectation
45	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE** 0- Reset card and sendTerminal Profile without facility SETUP_EVENT_LIST 1- Envelope event download network search mode change is sent to the UICC 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 Applet2 finalizes	
46	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_BROWSING STATUS** 1- Envelope event download browsing status is sent to the UICC 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 Applet2 finalizes	
47	**The ProactiveHandler is not available outside the processToolkit() method** 1- Install Applet3. In its install method, Applet3 gets the ProactiveHandler in a Try/Catch session 2- Select Applet3 3- Applet3 gets the ProactiveHandler	1- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 2- Applet3 is triggered by its process() method 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown	

5.5.1.2 ProactiveResponseHandler

Test Area Reference: Cre_Mha_Prhd.

5.5.1.2.1 Conformance requirement

5.5.1.2.1.1 Normal execution

- CRRN1: The ProactiveResponseHandler is available as soon as the ProactiveHandler is available and remains available untill the termination of the processToolkit method for the following events:

  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_BY_NAA

EVENT_PROFILE_DOWNLOAD

EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS

EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE

EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED

EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION

EVENT_PROACTIVE_HANDLER_AVAILABLE

EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE

EVENT_EVENT_BROWSING_STATUS

EVENT_EXTERNAL_FILE_UPDATE

- CRRN2: A ProactiveResponseHandler is considered available when no HANDLER_NOT_AVAILABLE ToolkitException is thrown when the corresponding getTheHandler() method is called or a method of the handler is called.

5.5.1.2.1.2 Parameter errors

No requirements.

5.5.1.2.1.3 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_ATR

  EVENT_APPLICATION_DESELECT

- CRRC2: The ProactiveResponseHandler shall not be available if the ProactiveHandler is not available.

- CRRC3: The ProactiveResponseHandler shall not be available if the *getTheHandler()* method is not called, directly or indirectly, from the applet's *processToolkit*() method.

5.5.1.2.2 Test area files

Test Source: Test_Cre_Mha_Prhd.java.

Test Applet: Cre_Mha_Prhd_1.java.

Cre_Mha_Prhd_2.java.

Cre_Mha_Prhd_3.java.

Cap File: Cre_Mha_Prhd.cap.

5.5.1.2.3 Test coverage

CRR Number	Test Case Number
CRRN1	2 to 23, 45, 46
CRRN2	1 to 22,45,46
CRRC1	1, 24
CRRC2	25 to 44
CRRC3	47

5.5.1.2.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applets registration to all events and Proactive Handler availability with EVENT_FIRST_COMMAND_AFTER_ATR** Applet1 is registered to all events defined in TS 102 241 [9] except to EVENT_PROACTIVE_HANDLER_AVAILABLE. Using the methods initMenuEntry() for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION, allocateServiceIdentifier() for EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION and setEventList() for the rest of the events. Applet1 is registered on EVENT_EXTERNAL_FILE_UPDATE on update of EFTARU of the UICC file system Applet2 is registered to all events defined in TS 102 241 [9], except to EVENT_PROACTIVE_HANDLER_AVAILABLE and EVENT_CALL_CONTROL_BY_NAA. 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 on EVENT_EXTERNAL_FILE_UPDATE on update of EFTARU of the UICC file system The priority of Applet1 is higher than priority of Applet2		
	1- Select MF	1- Applet1 is triggered by EVENT_FIRST_COMMAND_AFTER_ATR	
	2- Applet1 gets the Proactive Handler.	2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown.	
	3- Applet2 gets the Proactive Response Handler	3- A ToolkitException HANDLER_NOT_AVAILABLE is	

Id	Description	API/CAT RE Expectation	APDU Expectation
	Applet1 is deregistered from EVENT_FIRST_COMMAND_AFTER_ATR. 4- Applet2 get the Proactive Handler 5- Applet2 gets the Proactive Response Handler Applet2 is deregistered to EVENT_FIRST_COMMAND_AFTER_ATR.	thrown Applet1 finalizes Applet2 is triggered by EVENT_FIRST_COMMAND_AFTER_ATR 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. 5- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes	
2	**Proactive Handler availability with EVENT_PROFILE_DOWNLOAD** 1- Terminal Profile command is sent to the UICC without the facility of SET UP EVENT LIST, POLL INTERVAL,SET UP IDLE MODE TEXT and SET UP MENU. 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler Applet1 is deregistered to EVENT_PROFILE_DOWNLOAD 4- Applet2 gets the Proactive Handler 5-Applet2 gets the Proactive Respones Handler Applet2 is deregistered to EVENT_PROFILE_DOWNLOAD	1- Applet1 is triggered by EVENT_PROFILE_DOWNLOAD 2- No exception is thrown 3- No exception is thrown Applet1 finalizes. Applet2 is triggered by EVENT_PROFILE_DOWNLOAD 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	
3	**Proactive Handler availability with EVENT_MENU_SELECTION_HELP_REQUEST** Perform UICC initialization with all the facilities supported, without facility SET_UP_EVENT_LIST 1- Envelope menu selection with help request is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes	
4	**Proactive Handler availability with EVENT_MENU_SELECTION** 1- Envelope menu selection is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
5	**Proactive Handler availability with EVENT_TIMER_EXPIRATION** 1- Timer Id =1 Envelope Timer Expiration is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes	
6	**Proactive Handler availability with EVENT_CALL_CONTROL_BY_NAA** 1- Envelope call control by NAA is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes	
7	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_MT_CALL** 1- Envelope event download mt call is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4-No exception is thrown 5-No exception is thrown Applet2 finalizes	
8	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_CALL_CONNECTED** 1- Envelope event download call connected is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown 5. No exception is thrown Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
9	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED** 1- Envelope event download call disconnected is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	
10	**Applets triggering with EVENT_EVENT_DOWNLOAD_LOCATION_STATUS** 1- Envelope event download location status is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	
11	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_USER_ACTIVITY** 1- Envelope event download user activity is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
12	**Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS** 1-Envelope event download card reader status is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	
13	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS** 1- Envelope event download card reader status is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	
14	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION** 1- Envelope event download language selection is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
15	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION** 1- Envelope event download browser termination is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	 1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	
16	**Proactive Handler availability with EVENT_STATUS_COMMAND** 1- Status command is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	 1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	
17	**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 UICC, with channelId=01. 3- Applet1 gets the Proactive Handler 4- Applet1 gets the Proactive Response Handler	 2- Applet1 is triggered 3- No exception is thrown 4- No exception is thrown Applet1 finalizes	1- OPEN CHANNEL proactive command is fetched TERMINAL RESPONSE is issued with Channel Id = 01
18	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS** 1- An Envelope Event Download Channel Status is sent to the UICC, with ChannelId=01 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler	 1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
19	**Proactive Handler availability with UNRECOGNIZED_ENVELOPE** 1- An unrecognized Envelope (BER TLV Tag unrecognized) is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	
20	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE** 1- An envelope event download access technology change is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	
21	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED** 1- An envelope event download display parameter changed is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 5- Applet2 gets the Proactive Handler 6- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown 4- Applet1 sends a DECLARE SERVICE (add) proactive command with its service identifier Applet1 finalizes Applet2 is triggered 5- No exception is thrown 6- No exception is thrown Applet2 finalizes	4- DECLARE SERVICE (add) proactive command is fetched Successful TERMINAL RESPONSE is issued

Id	Description	API/CAT RE Expectation	APDU Expectation
22	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION** 1- An envelope event download local connection is sent to the UICC, with the allocated service identifier 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes	
23	**Proactive Handler availability with EVENT_PROACTIVE_HANDLER_AVAILABLE** 1- Envelope event download call connected is sent to the UICC 2- The display Text proactive command is fetch and the terminal response is sent 3- Applet1 gets the Proactive Handler 4- Applet1 gets the Proactive Response Handler 5- Applet2 gets the Proactive Handler 6- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered and registers to EVENT_PROACTIVE_HANDLER_AVAILABLE Applet1 finalizes Applet2 is triggered, registers to EVENT_PROACTIVE_HANDLER_AVAILABLE and sends a Display Test proactive command 2- Applet1 is triggered by EVENT_PROACTIVE_HANDLER_AVAILABLE 3- No exception is thrown 4- No exception is thrown Applet1 finalizes Applet2 is triggered by EVENT_PROACTIVE_HANDLER_AVAILABLE 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
24	**Proactive Handler availability with EVENT_APPLICATION_DESELECT** 1- Select for activation ADF1 2- Select for termination ADF1 3- Applet1 gets the Proactive Handler 4- Applet1 gets the Proactive Response Handler Applet1 deregisters to EVENT_APPLICATION_DESELECT. 5-Applet2 gets the Proactive Handler 6-Applet2 gets the Proactive Response Handler Applet2 deregisters to EVENT_APPLICATION_DESELECT.	2- Applet1 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 5- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 6- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
25	**The ProactiveHandler is not available before the Terminal Profile with EVENT_MENU SELECTION_HELP_REQUEST** 1- Reset the card without sending the Terminal Profile 2- Envelope menu selection with help request is sent to the UICC 3- Applet1 gets the Proactive Handler 4- Applet1 gets the Proactive Response Handler	2- Applet1 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
26	**The ProactiveHandler is not available before the Terminal Profile with EVENT_MENU SELECTION** 1- Envelope menu selection is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
27	**The ProactiveHandler is not available before the Terminal Profile with EVENT_TIMER_EXPIRATION** 1- Timer Id =1 Envelope Timer Expiration is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
28	**The ProactiveHandler is not available before the Terminal Profile with EVENT_CALL_CONTROL_BY_NAA** 1- Envelope call control by NAA is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
29	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_MT_CALL** 1- Envelope event download mt call is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5-A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
30	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_CALL_CONNECTED** 1- Envelope event download call connected is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5-A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
31	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED** 1- Envelope event download call disconnected is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5-A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
32	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_LOCATION_STATUS** 1- Envelope event download location status is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
33	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_USER_ACTIVITY** 1- Envelope event download user activity is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5-A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
34	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE** 1- Envelope event download idle screen available is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes	
35	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS** 1- Envelope event download card reader status is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3-A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
36	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION** 1- Envelope event download language selection is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3--A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5-A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes	
37	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION** 1- Envelope event download browser termination is sent to the UICC 2- Applet1 gets the Proactive Handler **3- Applet2 gets the Proactive Handler**	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
38	**The ProactiveHandler is not available before the Terminal Profile with EVENT_STATUS_COMMAND** 1- Status command is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3-A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes	
39	**The ProactiveHandler is not available before the Terminal Profile with UNRECOGNIZED_ENVELOPE** 1- An unrecognized Envelope (BER TLV Tag unrecognized) is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
40	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE** 1- An envelope event download access technology change is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3-A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5-A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes	
41	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED** 1- An envelope event download display parameter changed is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3-A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet1 registers to EVENT_PROACTIVE_HANDLER_AVAILABLE then finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet2 registers to EVENT_PROACTIVE_HANDLER_AVAILABLE then finalizes	
42	**The ProactiveHandler is not available before the Terminal Profile with EVENT_PROACTIVE_HANDLER_AVAILABLE**	Applet1 and Applet2 are not triggered by EVENT_PROACTIVE_HANDLER_AVAILABLE	

Id	Description	API/CAT RE Expectation	APDU Expectation
43	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE** 1- Envelope event download network search mode change is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet1 deregisters from EVENT_PROACTIVE_HANDLER_AVAILABLE Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet2 deregisters from EVENT_PROACTIVE_HANDLER_AVAILABLE Applet2 finalizes	
44	**The ProactiveHandler is not available before the Terminal Profile with EVENT_EVENT_DOWNLOAD_BROWSING STATUS** 1- Envelope event download browsing status is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 5- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
45	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE** 0- Reset card and sendTerminal Profile without facility SETUP_EVENT_LIST 1- Envelope event download network search mode change is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	 1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	
46	**Proactive Handler availability with EVENT_EVENT_DOWNLOAD_BROWSING STATUS** 1- Envelope event download browsing status is sent to the UICC 2- Applet1 gets the Proactive Handler 3- Applet1 gets the Proactive Response Handler 4- Applet2 gets the Proactive Handler 5- Applet2 gets the Proactive Response Handler	 1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown 5- No exception is thrown Applet2 finalizes	
47	**The ProactiveHandler is not available outside the processToolkit() method** 1- Install Applet3. In its install method, Applet3 gets the Proactive Handler and the Proactive Response Handler in a Try/Catch session 2- Select Applet3 3- Applet3 gets the ProactiveHandler and the Proactive Response Handler.	 1- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 2- Applet3 is triggered by its process() method 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown	

5.5.1.3 EnvelopeHandler

Test Area Reference: Cre_Mha_Enhd.

5.5.1.3.1 Conformance requirement

5.5.1.3.1.1 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_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_BY_NAA

  EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

  EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS

  EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE

  EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED

  EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION

  EVENT_APPLICATION_DESELECT

  EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE

  EVENT_EVENT_BROWSING_STATUS

  EVENT_EXTERNAL_FILE_UPDATE

- CRRN2: An EnvelopeHandler is considered available when no HANDLER_NOT_AVAILABLE ToolkitException is thrown when the corresponding getTheHandler() method is called or a method of the handler is called.

5.5.1.3.1.2 Parameter errors

No requirements.

5.5.1.3.1.3 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_ATR

  EVENT_PROACTIVE_HANDLER_AVAILABLE

- CRRC2: The EnvelopeHandler shall not be available if the *getTheHandler()* method is not called, directly or indirectly, from the applet's *processToolkit*() method.

5.5.1.3.2 Test area files

Test Source: Test_Cre_Mha_Enhd.java.

Test Applet: Cre_Mha_Enhd_1.java.

Cre_Mha_Enhd_2.java.

Cre_Mha_Enhd_3.java.

Cap File: Cre_Mha_Enhd.cap.

5.5.1.3.3 Test coverage

CRR Number	Test Case Number
CRRN1	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18, 19, 20, 22, 24, 26, 27, 28, 29
CRRN2	1 to 24, 26, 27, 28, 29
CRRC1	1, 2, 16, 23
CRRC2	25

5.5.1.3.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet1 and Applet2 registration and Envelope Handler availability with EVENT_FIRST_COMMAND_AFTER_ATR** 1- Applet1 is registered to all events defined TS 102 241 [9] except to EVENT_PROACTIVE_HANDLER_AVAILABLE. The registration is done using the methods initMenuEntry() for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION, allocateServiceIdentifier() for EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION and setEventList() for the rest of the events. Applet1 is registered on EVENT_EXTERNAL_FILE_UPDATE on update of EF_{TARU} of the UICC file system Applet2 is registered to all events defined TS 102 241 [9] except to EVENT_PROACTIVE_HANDLER_AVAILABLE and EVENT_CALL_CONTROL_BY_NAA . 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 on EVENT_EXTERNAL_FILE_UPDATE on update of EF_{TARU} of the UICC file system 2- Select MF. 3- EnvelopeHandler.getTheHandler() method is called by Applet1 Applet1 is deregistered from EVENT_FIRST_COMMAND_AFTER_ATR. 4- EnvelopeHandler.getTheHandler() method is called by Applet2 Applet2 is deregistered to EVENT_FIRST_COMMAND_AFTER_ATR.	1- No exception is thrown 2- Applet1 is triggered by EVENT_FIRST_COMMAND_AFTER_ATR 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 4- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
2	**Handler availability with EVENT_PROFILE_DOWNLOAD** 1- Terminal Profile command is sent to the UICC 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 ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered by EVENT_PROFILE_DOWNLOAD 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
3	**Envelope Handler availability with EVENT_MENU_SELECTION_HELP_REQUEST** Perform UICC initialization with all the facilities supported 1- Envelope menu selection with help request is sent to the UICC 2- EnvelopeHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes	
4	**Envelope Handler availability with EVENT_MENU_SELECTION** 1- Envelope menu selection is sent to the UICC 2- EnvelopeHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes	
5	**Envelope Handler availability with EVENT_TIMER_EXPIRATION** Timer id=1 1- Envelope Timer Expiration is sent to the UICC 2- EnvelopeHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes	
6	**Envelope Handler availability with EVENT_CALL_CONTROL_BY_NAA** 1- Envelope call control by NAA is sent to the UICC 2- EnvelopeHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes	
7	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_MT_CALL** 1- Envelope event download mt call is sent to the UICC 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 Applet2 finalizes	
8	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_CALL_CONNECTED** 1- Envelope event download call connected is sent to the UICC 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 Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
9	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_CALL_DISCONECTTED** 1- Envelope event download call disconnected is sent to the UICC 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 Applet2 finalizes	
10	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_LOCATION_STATUS** 1- Envelope event download location status is sent to the UICC 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 Applet2 finalizes	
11	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_USER_ACTIVITY** 1- Envelope event download user activity is sent to the UICC 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 Applet2 finalizes	
12	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE** 1- Envelope event download idle screen available is sent to the UICC 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 Applet2 finalizes	
13	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS** 1- Envelope event download card reader status is sent to the UICC 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 Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
14	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_LANGUAGE_ SELECTION** 1- Envelope event download language selection is sent to the UICC 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 Applet2 finalizes	
15	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_BROWSER_ TERMINATION** 1- Envelope event download browser termination is sent to the UICC 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 Applet2 finalizes	
16	**Envelope Handler availability with EVENT_STATUS_COMMAND** 1- Status command is sent to the UICC 2- EnvelopeHandler.getTheHandler() method is called by Applet1 3- EnvelopeHandler.getTheHandler() method is called by Applet2	 1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
17	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_DATA_AVAILAB LE** 1- Applet1 builds a proactive command OPEN CHANNEL. proactiveHandler.send() method is called 2- Envelope event download data available is sent to the UICC with ChannelId=01. 3- EnvelopeHandler.getTheHandler() method is called by Applet1	 1- Applet1 is registered to EVENT_EVENT_DOWNLOAD_DA TA_AVAILABLE and EVENT_EVENT_DOWNLOAD_CH ANNEL_STATUS 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

Id	Description	API/CAT RE Expectation	APDU Expectation
18	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS** 1- Envelope event download channel status is sent to the UICC with ChannelId=01. 2- EnvelopeHandler.getTheHandler() method is called by Applet1	 1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes	
19	**Envelope Handler availability with EVENT_UNRECOGNIZED_ENVELOPE** 1- An unrecognized Envelope is sent to the UICC 2- EnvelopeHandler.getTheHandler() method is called by Applet1 4- EnvelopeHandler.getTheHandler() method is called by Applet2	 1- Applet1 is triggered 2- No exception is thrown 3- Applet1 sends a DECLARE SERVICE (add) proactive command with its service identifier Applet1 finalizes Applet2 is triggered 4- No exception is thrown Applet2 finalizes	 3- DECLARE SERVICE (add) proactive command is fetched Successful TERMINAL RESPONSE is issued
20	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION** 1- Envelope event download local connection is sent to the UICC with the allocated service Id of Applet1 2- EnvelopeHandler.getTheHandler() method is called by Applet1	 1- Applet1 is triggered 2- No exception is thrown Applet1 finalizes	
21	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE** 1- Envelope event download access technology change is sent to the UICC 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 Applet2 finalizes	
22	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED** 1- Envelope event display parameter changed is sent to the UICC 2- EnvelopeHandler.getTheHandler() method is called by Applet1 3- EnvelopeHandler.getTheHandler() method is called by Applet2	 1- Applet1 is triggered and registers to EVENT_PROACTIVE_HANDLER_AVAILABLE 2- No exception is thrown Applet1 finalizes Applet2 is triggered and registers to EVENT_PROACTIVE_HANDLER_AVAILABLE 3- No exception is thrown Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
23	**Envelope Handler availability with EVENT_PROACTIVE_HANDLER_AVAILABLE** 2- EnvelopeHandler.getTheHandler() method is called by Applet1 3- EnvelopeHandler.getTheHandler() method is called by Applet2	1- Applet1 is triggered by EVENT_PROACTIVE_HANDLER_AVAILABLE 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes Applet2 is triggered by EVENT_PROACTIVE_HANDLER_AVAILABLE 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet2 finalizes	
24	**Envelope Handler availability with EVENT_APPLICATION_DESELECT** 1- Select for activation ADF1 2- Select for termination ADF1 3- EnvelopeHandler.getTheHandler() method is called by Applet1. Applet1 deregisters to EVENT_APPLICATION_DESELECT. 4- EnvelopeHandler.getTheHandler() method is called by Applet2. Applet1 deregisters to EVENT_APPLICATION_DESELECT.	2- Applet1 is triggered 3- No exception is thrown Applet1 finalizes Applet2 is triggered 4- No exception is thrown Applet2 finalizes	
25	**Envelope Handler availability with EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE** 1- Envelope event download network search mode change is sent to the UICC 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 Applet2 finalizes	
26	**Envelope Handler availability with EVENT_EVENT_BROWSING_STATUS** 1- Envelope event download browsing status is sent to the UICC 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 Applet2 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
27	**The EnvelopeHandler is not available outside the processToolkit() method** 1- Install Applet3. In its install method, Applet3 gets the EnvelopeHandler in a Try/Catch session 2- Select Applet3 3- Applet3 gets the EnvelopeHandler	1- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 2- Applet3 is triggered by its process() method 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown	

5.5.1.4 EnvelopeResponseHandler

Test Area Reference: Cre_Mha_Erhd

5.5.1.4.1 Conformance requirement

5.5.1.4.1.1 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_CALL_CONTROL_BY_NAA

  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.
- CRRN4: An EnvelopeResponseHandler is considered available when no HANDLER_NOT_AVAILABLE ToolkitException is thrown when the corresponding getTheHandler() method is called or a method of the handler is called.

5.5.1.4.1.2 Parameter errors

No requirements.

5.5.1.4.1.3 Context errors

- CRRC1: The handler is not available for the following events:

  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_ATR

EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS

EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE

EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED

EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION

EVENT_APPLICATION_DESELECT

EVENT_PROACTIVE_HANDLER_AVAILABLE

EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE

EVENT_EVENT_BROWSING_STATUS

EVENT_EXTERNAL_FILE_UPDATE

- CRRC2: The EnvelopeResponseHandler shall not be available if the *getTheHandler()* method is not called, directly or indirectly, from the applet's *processToolkit*() method.

5.5.1.4.2 Test area files

Test Source: Test_Cre_Mha_Erhd.java.

Test Applet: Cre_Mha_Erhd_1.java.

Cre_Mha_Erhd_2.java.

Cre_Mha_Erhd_3.java.

Cap File: Cre_Mha_Erhd.cap.

5.5.1.4.3 Test coverage

CRR Number	Test Case Number
CRRN1	18, 19, 20, 21
CRRN2	18, 19
CRRN3	18, 19
CRRN4	1 to 29
CRRC1	1 to 17 and 22 to 25, 26, 27, 28
CRRC2	29

5.5.1.4.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Toolkit Applet1 and Toolkit Applet2 registration and Envelope Response Handler availability with EVENT_FIRST_COMMAND_AFTER_ATR** 1- Applet1 is registered to all events defined in TS 102 241 [9] except EVENT_PROACTIVE_HANDLER_AVAILABLE. Using the methods initMenuEntry() for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION, allocateServiceIdentifier() for EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION and setEventList() for the rest of the events. Applet1 is registered on EVENT_EXTERNAL_FILE_UPDATE on update of EF_{TARU} of the UICC file system Applet2 is registered to EVENT_UNRECOGNIZED_ENVELOPE. 2- Select MF. 3- EnvelopeResponseHandler.getTheHandler() method is called by Applet1 Applet1 is deregistered to EVENT_FIRST_COMMAND_AFTER_ATR.	1- No exception is thrown 2- Applet1 is triggered by EVENT_FIRST_COMMAND_AFTER_ATR 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
2	**Handler availability with EVENT_PROFILE_DOWNLOAD** 1- Terminal Profile command is sent to the UICC 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 ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
3	**Envelope Response Handler availability with EVENT_MENU_SELECTION_HELP_REQUEST** Perform UICC initialization with all the facilities supported, except facility SET UP EVENT LIST 1- Envelope menu selection with help request is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
4	**Envelope Response Handler availability with EVENT_MENU_SELECTION** 1- A envelope menu selection is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
5	**Envelope Response Handler availability with EVENT_TIMER_EXPIRATION** 1- Envelope Timer Expiration is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown	
6	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_MT_CALL** 1- Envelope event download mt call is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
7	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_CALL_CONNECTED** 1- Envelope event download call connected is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
8	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED** 1- Envelope event download call disconnected is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
9	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_LOCATION_STATUS** 1- Envelope event download location status is sent to the UICC 2- Applet1 obtains the Envelope Response Handler	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
10	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_USER_ACTIVITY** 1- Envelope event download user activity is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	

Id	Description	API/CAT RE Expectation	APDU Expectation
11	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE** 1- Envelope event download idle screen available is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
12	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS** 1- Envelope event download card reader status is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
13	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION** 1- Envelope event download language selection is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
14	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION** 1- Envelope event download browser termination is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
15	**Envelope Response Handler availability with EVENT_STATUS_COMMAND** 1- Status command is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
16	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE** 1- Applet1 initializes a proactive command OPEN CHANNEL and calls the send() method. 2- Envelope event download data avalaible is sent to the UICC with channelId=01 3- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	2- Applet1 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	1- The OPEN CHANNEL command is fetched TERMINAL RESPONSE IS SENT TO THE UICC with channel Id=01

Id	Description	API/CAT RE Expectation	APDU Expectation
17	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS** 1- Envelope event download channel status is sent to the UICC with channelId=01 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
18	**Envelope Response Handler availability with EVENT_CALL_CONTROL_BY_NAA** 1- Envelope call control by NAA is sent to the UICC 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 NAA is sent to the UICC 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- ToolkitException HANDLER_NOT_AVAILABLE is thrown for each method Applet1 finalizes 5- Applet1 is triggered 6- No Exception is thrown 8- A ToolkitException HANDLER_NOT_AVAILABLE is thrown for each method Applet1 finalizes	3- The envelope response is sent 7- The proactive command is fetched and the Terminal response is issued

Id	Description	API/CAT RE Expectation	APDU Expectation
19	**Envelope Response Handler availability with EVENT_UNRECOGNIZED_ENVELOPE** 1- An unrecognized Envelope is sent to the UICC 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 UICC 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 ToolkitException HANDLER_NOT_AVAILABLE is thrown for each method Applet1 finalizes Applet2 is triggered 5- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes 6- Applet1 is triggered 7- No exception is thrown 9- A ToolkitException HANDLER_NOT_AVAILABLE is thrown for each method Applet1 finalizes Applet2 is triggered 10- A ToolkitException HANDLER_NOT_AVAILABLE is thrown. Applet2 finalizes	3- The envelope response is sent 9- The proactive command is fetched and the Terminal response is issued
20	**The envelope response is sent when a proactive session is ongoing** 1- An unrecognized envelope is sent to the UICC. 2- Proactive command DISPLAY TEXT is built and it calls the send() method. 3- A call control by NAA envelope is sent to the UICC. 4- EnvelopeResponseHandler.getTheHandler() method is called by Applet1 5- Applet1 builds the envelope response and it calls the postAsBERTLV()	1- Applet1 is triggered. 3- Applet1 is triggered 4- No exception is thrown	2- 91 XX 5- The envelope response is checked Expected SW = 91 XX Fetch DISPLAY TEXT Terminal Response DISPLAY TEXT

Id	Description	API/CAT RE Expectation	APDU Expectation
21	**Envelope Response Handler availability with EVENT_UNRECOGNIZED_ENVELOPE in case of multi-triggering** 1- An unrecognized Envelope is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1 3- EnvelopeResponseHandler.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 Applet2 finalizes	
22	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE** 1- Envelope event download access technology change is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
23	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED** 1- Envelope event download display parameter changed is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered and registers to EVENT_PROACTIVE_HANDLER_AVAILABLE 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
24	**Envelope Response Handler availability with EVENT_PROACTIVE_HANDLER_AVAILABLE** 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered by EVENT_PROACTIVE_HANDLER_AVAILABLE 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 3- Applet1 sends a DECLARE SERVICE (add) proactive command with its service identifier Applet1 finalizes	3- DECLARE SERVICE (add) proactive command is fetched Successful TERMINAL RESPONSE is issued

Id	Description	API/CAT RE Expectation	APDU Expectation
25	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_LOCAL_CONNACTION** 1- Envelope event download local connection is sent to the UICC with the allocated service Id of Applet1 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
26	**Envelope Response Handler availability with EVENT_APPLICATION_DESELECT** 1- Select for activation ADF1 2- Select for termination ADF1 3- EnvelopeResponseHandler.getTheHandler() method is called by Applet1 Applet1 deregisters to EVENT_APPLICATION_DESELECT	2- Applet1 is triggered 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
27	**Envelope Response Handler availability with EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE** 1- Envelope event download network search mode change is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
28	**Envelope Response Handler availability with EVENT_EVENT_BROWSING_STATUS** 1- Envelope event download browsing status is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	1- Applet1 is triggered 2- A ToolkitException HANDLER_NOT_AVAILABLE is thrown Applet1 finalizes	
29	**The EnvelopeResponseHandler is not available outside the processToolkit() method** 1- Install Applet3. In its install method, Applet3 gets the EnvelopeResponseHandler in a Try/Catch session 2- Select Applet3 3- Applet3 gets the EnvelopeResponseHandler	1- A ToolkitException HANDLER_NOT_AVAILABLE is thrown 2- Applet3 is triggered by its process() method 3- A ToolkitException HANDLER_NOT_AVAILABLE is thrown	

5.5.2 Handler Integrity

5.5.2.1 ProactiveHandler

Test Area Reference: Cre_Hin_Pahd.

5.5.2.1.1 Conformance requirement

5.5.2.1.1.1 Normal execution

- CRRN1: At the processToolkit invocation the TLV-List is cleared.
- CRRN2: After a call to *ProactiveHandler.send()* method the content of the handler shall not be modified by the CAT Runtime Environment.
- CRRN3: At the call of its init method the content is cleared and then initialized.

5.5.2.1.1.2 Parameter errors

No requirements.

5.5.2.1.1.3 Context errors

No requirements.

5.5.2.1.2 Test area files

Test Source: Test_Cre_Hin_Pahd.java.

Test Applet: Cre_Hin_Pahd_1.java.

Cre_Hin_Pahd_2.java.

Cap File: Cre_Hin_Pahd.cap.

5.5.2.1.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3, 4
CRRN2	3
CRRN3	5

5.5.2.1.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**At the processToolkit invocation the TLV-List is cleared** Applet1 and Applet2 are registered to EVENT_EVENT_DOWNLOAD_USER_ACTIVITY. 1-An envelope containing an event download user activity is sent to the UICC 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 is checked that the content is the same than before the calling to send method using ProactiveHandler.copy and Util.arrayCompare methods	2- The return value is 9, and its contents is the same than before the calling to send method Applet1 finalizes	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	Applet2 is triggered 1- The return value is 0 2- ToolkitException UNAVAILABLE_ELEMENT is thrown	

Id	Description	API/CAT RE Expectation	APDU Expectation
5	**At the call of its init method the content is cleared and then initialized** 1- proactiveHandler.init() method is called by Applet2 2- proactiveHandler.initCloseChannel() method is called by Applet2 3- proactiveHandler.initDisplayText() method is called by Applet2 4- proactiveHandler.initGetInkey() method is called by Applet2 5- proactiveHandler.initGetInput() method is called by Applet2 6- ProactiveHandler.initMoreTime() method is called by Applet2 7- proactiveHandler.init() method is called by Applet2	1- Check that the content of the proactive handler corresponds to the command defined by the parameters of the init() method 2- Check that the content of the proactive handler corresponds to the close channel command 3- Check that the content of the proactive handler corresponds to the display text command 4- Check that the content of the proactive handler corresponds to the get inkey command 5- Check that the content of the proactive handler corresponds to the get input command 6- Check that the content of the proactive handler corresponds to the More Time command 7- Check that the content of the proactive handler corresponds to the command defined by the parameters of the init() method	

5.5.2.2 ProactiveResponseHandler

Test Area Reference: Cre_Hin_Prhd.

5.5.2.2.1 Conformance requirement

5.5.2.2.1.1 Normal execution

- CRRN1: The *ProactiveResponseHandler* content shall be updated after each successful call to *ProactiveHandler.send()* method and shall remain unchanged until the next successful call to the *ProactiveHandler.send()* method.
- CRRN2: The ProactiveResponseHandler TLV list is filled with the comprehension TLV data objects of the last TERMINAL RESPONSE APDU command.
- CRRN3: The comprehension TLV data objects shall be provided in the order given in the TERMINAL RESPONSE command data.
- CRRN4: The ProactiveResponseHandler TLV list shall be empty before the first call to the ProactiveHandler.send() method.

5.5.2.2.1.2 Parameter errors

No requirements.

5.5.2.2.1.3 Context errors

No requirements.

5.5.2.2.2 Test area files

Test Source: Test_Cre_Hin_Prhd.java.

Test Applet: Cre_Hin_Prhd_1.java.

Cap File: Cre_Hin_Prhd.cap.

5.5.2.2.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2
CRRN3	2
CRRN4	1

5.5.2.2.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration and ProactiveResponseHandler obtaining** 1- Applet is registered to all events defined in ETSI TS 102 241[13] except to EVENT_FIRST_COMMAND_AFTER_ATR, EVENT_PROACTIVE_HANDLER_AVAILABLE and EVENT_APPLICATION_DESELECT. Using the methods initMenuEntry for EVENT_MENU_SELECTION and EVENT_MENU_SELECTION_HELP_REQUEST requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION, allocateServiceIdentifier() for EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION and setEventList() for the rest of the events. Applet is registered on EVENT_EXTERNAL_FILE_UPDATE on update of EF_{TARU} of the UICC file system Terminal Profile command is sent to the UICC without the facilities of SETUP_EVENT_LIST, SETUP_IDLE_MODE_TEXT, SETUP_MENU and POLL_INTERVAL. 2- For each event/triggering: 3- ProactiveResponseHandler.getTheHandler() is called 4- ProactiveResponseHandler.getLength() is called	1- No exception is thrown 2- Applet is triggered. For the first triggering, the applet registers to EVENT_PROACTIVE_HANDLER_AVAILABLE 3- No exception is thrown 4- The return value is 0	

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**The ProactiveResponseHandler remains unchanged after send() method invocation until next send() method invocation**		
	1- Applet builds a proactive command then the ProactiveHandler.send() method is called		1- A proactive command is fetched The terminal response is sent with length 12
	2- proactiveResponseHandler.getTheHandler() method is called	2- The ProactiveResponseHandler contains the terminal response. Comprehension TLV are provided in the order of the Terminal Response APDU	
	3-ProactiveResponseHandler.getLength() method is called	3- The return value is 12	
	4-ProactiveHandler.init() method is called	4- No exception is thrown and the Proactive Response Handler remains unchanged	
	5-ProactiveHandler.send() method is called		5- A proactive command is fetched The terminal response is sent with length 15
	6- proactiveResponseHandler.getTheHandler() method is called	6- The ProactiveResponseHandler contains the terminal response of the second proactive command. Comprehension TLV are provided in the order of the Terminal Response APDU	
	7-ProactiveResponseHandler.getLength() method is called	7- The return value is 15	

5.5.2.3 EnvelopeHandler

Test Area Reference: Cre_Hin_Enhd.

5.5.2.3.1 Conformance requirement

5.5.2.3.1.1 Normal execution

- CRRN1: When available, the *EnvelopeHandler* shall remain available and its content shall remain unchanged from the invocation to the termination of the *processToolkit()* method.
- CRRN2: The EnvelopeHandler TLV list is filled with the Comprehension TLV data objects of the ENVELOPE APDU command. The Comprehension TLV data objects shall be provided in the order given in the ENVELOPE command data.
- CRRN3: When an applet is triggered by the *EVENT_EXTERNAL_FILE_UPDATE* event, the system Envelope Handler shall be available
- CRRN4: When an applet is triggered by the *EVENT_EXTERNAL_FILE_UPDATE* event, the Envelope Handler shall contains the following COMPREHENSION TLVs (the order of the TLVs given in the system EnvelopeHandler is not specified):
  - Device Identity with source set to terminal and destination set to UICC, as defined in ETSI TS 102 223 [6];
  - File List, as defined in ETSI TS 102 223 [6]. The number of files shall be set to one. If a SFI referencing is used in the APDU Command, it shall be converted to its File Identifier;

- AID of the ADF, as defined in ETSI TS 102 223 [6], if the updated file belongs to an ADF. In this case, the path '3F007FFF' given in the File List indicates the ADF of the UICC application given through the AID. If the updated file belongs to the UICC shared file system, the AID TLV object is not present;

- File Update Information object.

Byte(s)	Description	Length
1	File Update Information tag	1
2	Length = 4	1
3 to 4	Position	2
5 to 6	Number of bytes updated	2

Position depends on the file type:

In case of transparent file, Position = Offset

In case of record file, Position = Absolute Record number

For the INCREASE APDU, the number of bytes updated is the record length.

- CRRN5: The value returned upon a *getTag()* method invocation is equal to the BER-TLV tag for intra-UICC communication, as defined in ETSI TS 101 220 [4].

5.5.2.3.1.2 Parameter errors

No requirements.

5.5.2.3.1.3 Context errors

No requirements.

5.5.2.3.2 Test area files

Test Source: Test_Cre_Hin_Enhd.java.

Test Applet: Cre_Hin_Enhd_1.java.

Cap File: Cre_Hin_Enhd.cap.

5.5.2.3.3 Test coverage

CRR Number	Test Case Number
CRRN1	1 to 21
CRRN2	1 to 21
CRRN3	22, 23
CRRN4	22, 23
CRRN5	1

5.5.2.3.4 Test procedure

Id	Description	API/CAT RE 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 102 241 [9] except EVENT_PROFILE_DOWNLOAD, EVENT_PROACTIVE_HANDLER_AVAILABLE and EVENT_STATUS_COMMAND. Using the methods initMenuEntry() for EVENT_MENU_SELECTION and EVENT_MENU_SELECTION_HELP_REQUEST, allocateTimer()for EVENT_TIMER_EXPIRATION, allocateServiceIdentifier() for EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION and setEventList() for the rest of the events. Perform UICC initialization with all the facilities supported Applet is registered on EVENT_EXTERNAL_FILE_UPDATE on update of EF_{TARU} of the UICC file system and on update of EF_{LARU} of the ADF file system 2- Envelope menu selection with help request is sent to the UICC 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- EnvelopeHandler.getTag() method is called 6- A proactive command DISPLAY TEXT is sent 7- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called 8- It is 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. 9- Check that the TAG_HELP_REQUEST is the TLV selected 10- 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 5- 0xD3 is returned 7- Applet is triggered 8- No exception is thrown and the handler contains the envelope call control by NAA 10- The contents of the envelope handler shall be the same as stored in buffer1	6- 91 xx. A proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**Envelope Handler integrity checks with EVENT_MENU_SELECTION** 1- An envelope menu selection is sent to UICC 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_ITEM_IDENTIFIER 4- A proactive command DISPLAY TEXT is sent 5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called 6- It is checked 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 7- It is checked that the TAG_ITEM_IDENTIFIER is the TLV selected 8- 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 NAA 8- The contents of the envelope handler shall be the same as stored in buffer1	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC

Id	Description	API/CAT RE Expectation	APDU Expectation
3	**Envelope Handler integrity checks with EVENT_TIMER_EXPIRATION**		
	1- A timer expiration envelope is sent to UICC	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() The EnvelopeHandler.findTLV() method is called with TAG_TIMER_ID	3- No exception is thrown.	
	4- A proactive command DISPLAY TEXT is sent		4- 91 XX
	5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called	5- Applet is triggered	
	6- It is 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	6- No exception is thrown and the handler contains the envelope call control by NAA	Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC
	7- It is checked that the TAG_TIMER_ID is the TLV selected 8- 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 buffer1	

Id	Description	API/CAT RE Expectation	APDU Expectation
4	**Envelope Handler integrity checks with EVENT_CALL_CONTROL_BY_NAA** 1- A call control envelope is sent to UICC 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 NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called 6- It is 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 7- It is checked that the TAG_ADDRESS is the TLV selected 8- 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 NAA 8- The contents of the envelope handler shall be the same as stored in buffer1	 4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC

Id	Description	API/CAT RE Expectation	APDU Expectation
5	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_MT_CALL** 1- A event download mt call envelope is sent to UICC 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_EVENT_LIST 4- A proactive command DISPLAY TEXT is sent 5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called 6- It is 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 7- It is checked that the TAG_EVENT_LIST is the TLV selected 8- 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 NAA 8- 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 UICC

Id	Description	API/CAT RE Expectation	APDU Expectation
6	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_CALL_CONNECTED**		
	1- A event download call connected envelope is sent to UICC	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() The EnvelopeHandler.findTLV() method is called with TAG_EVENT_LIST	3- No exception is thrown.	
	4- A proactive command DISPLAY TEXT is sent		4- 91 XX
	5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called	5- Applet is triggered	
	6- It is 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	6- No exception is thrown and the handler contains the envelope call control by NAA	
			Proactive command Display Text is fetched
	7- It is checked that the TAG_EVENT_LIST is the TLV selected		The terminal Response of DISPLAY TEXT is sent to the UICC
	8- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	8- The contents of the envelope handler shall be the same as stored in buffer1	

Id	Description	API/CAT RE Expectation	APDU Expectation
7	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED** 1- A event download call disconnected envelope is sent to UICC 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_EVENT_LIST 4- A proactive command DISPLAY TEXT is sent 5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called 6- It is 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 7- It is checked that the TAG_EVENT_LIST is the TLV selected 8- 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 NAA 8- The contents of the envelope handler shall be the same as stored in buffer1	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC

Id	Description	API/CAT RE Expectation	APDU Expectation
8	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_LOCATION_STATUS** 1- A event download location status envelope is sent to UICC 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_LOCATION_STATUS 4- A proactive command DISPLAY TEXT is sent 5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called 6- It is 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 7- It is checked that the TAG_LOCATION_STATUS is the TLV selected 8- 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 NAA 8- The contents of the envelope handler shall be the same as stored in buffer1	4-91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC

Id	Description	API/CAT RE Expectation	APDU Expectation
9	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_USER_ACTIVITY**		
	1- A event download user activity envelope is sent to UICC	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() The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES	3- No exception is thrown.	
	4- A proactive command DISPLAY TEXT is sent		4- 91 XX
	5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called	5- Applet is triggered	
	6- It is 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	6- No exception is thrown and the handler contains the envelope call control by NAA	
			Proactive command Display Text is fetched
	7-It is checked that the TAG_DEVICE_IDENTITIES is the TLV selected		The terminal Response of DISPLAY TEXT is sent to the UICC
	8- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	8- The contents of the envelope handler shall be the same as stored in buffer1	

Id	Description	API/CAT RE Expectation	APDU Expectation
10	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE**		
	1- A event download idle screen available envelope is sent to UICC	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() The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES	3- No exception is thrown.	
	4- A proactive command DISPLAY TEXT is sent		4- 91 XX
	5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called	5- Applet is triggered	
	6- It is 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	6- No exception is thrown and the handler contains the envelope call control by NAA	
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC
	7- It is checked that the TAG_DEVICE_IDENTITIES is the TLV selected		
	8- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	8- The contents of the envelope handler shall be the same as stored in buffer1	

Id	Description	API/CAT RE Expectation	APDU Expectation
11	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS**		
	1- A event download card reader status envelope is sent to UICC	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() The EnvelopeHandler.findTLV() method is called with TAG_CARD_READER_STATUS	3- No exception is thrown.	
	4- A proactive command DISPLAY TEXT is sent		4- 91 XX
	5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called	5- Applet is triggered	
	6- It is 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	6- No exception is thrown and the handler contains the envelope call control by NAA	
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC
	7- It is checked that the TAG_CARD_READER_STATUS is the TLV selected		
	8- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	8- The contents of the envelope handler shall be the same as stored in buffer1	

Id	Description	API/CAT RE Expectation	APDU Expectation
12	**Envelope Handler integrity checks with UNRECOGNIZED_ENVELOPE** 1- A unrecognized envelope is sent to UICC 2- EnvelopeHandler.getTheHandler() method is called 3- Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() 4- A proactive command DISPLAY TEXT is sent 5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called The EnvelopeHandler.getValueLength() is called 6- It is 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 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 NAA 7- The contents of the envelope handler shall be the same as stored in buffer1	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC

Id	Description	API/CAT RE Expectation	APDU Expectation
13	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION**		
	1- A event download language selection envelope is sent to UICC	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() The EnvelopeHandler.findTLV() method is called with TAG_EVENT_LIST	3-No exception is thrown.	
	4- A proactive command DISPLAY TEXT is sent		4-91 XX
	5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called	5- Applet is triggered	
	6- It is 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	6- No exception is thrown and the handler contains the envelope call control by NAA	
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC
	7- It is checked that the TAG_EVENT_LIST is the TLV selected		
	8- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	8- The contents of the envelope handler shall be the same as stored in buffer1	

Id	Description	API/CAT RE Expectation	APDU Expectation
14	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION**		
	1- A event download browser termination envelope is sent to UICC	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() The EnvelopeHandler.findTLV() method is called with TAG_EVENT_LIST	3-No exception is thrown.	
	4- A proactive command DISPLAY TEXT is sent		4-91 XX
	5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called	5- Applet is triggered	
	6- It is 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	6- No exception is thrown and the handler contains the envelope call control by NAA	Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC
	7- It is checked that the TAG_EVENT_LIST is the TLV selected		
	8- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	8- The contents of the envelope handler shall be the same as stored in buffer1	

Id	Description	API/CAT RE Expectation	APDU Expectation
15	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE**		
	1- The applet builds a proactive command OPEN CHANNEL. proactiveHandler.send() method is called	1- The applet is registered to EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE and EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS	1- OPEN CHANNEL proactive command is fetched TERMINAL RESPONSE is issued with Channel Id = 01
	2- A event download data available envelope is sent to UICC	2- Applet is triggered	
	3- EnvelopeHandler.getTheHandler() method is called	3-No exception is thrown.	
	4- Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_CHANNEL_STATUS	4-No exception is thrown.	
	5- A proactive command DISPLAY TEXT is sent		
	6- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called	6- Applet is triggered	5-91 XX
	7- It is 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	7- No exception is thrown and the handler contains the envelope call control by NAA	
	8- It is checked that the TAG_CHANNEL_STATUS is the TLV selected		Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC
	9- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	9- The contents of the envelope handler shall be the same as stored in buffer1	

Id	Description	API/CAT RE Expectation	APDU Expectation
16	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS**		
	1- A event download channel status envelope is sent to UICC	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() The EnvelopeHandler.findTLV() method is called with TAG_CHANNEL_STATUS	3-No exception is thrown.	
	4- A proactive command DISPLAY TEXT is sent		4-91 XX
	5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called	5- Applet is triggered	
	6- It is 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	6- No exception is thrown and the handler contains the envelope call control by NAA	
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC
	7- It is checked that the TAG_CHANNEL_STATUS is the TLV selected		
	8- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	8- The contents of the envelope handler shall be the same as stored in buffer1	

Id	Description	API/CAT RE Expectation	APDU Expectation
17	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE**		
	1- A event download access technology change envelope is sent to UICC	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() The EnvelopeHandler.findTLV() method is called with TAG_ACCESS_TECHNOLOGY	3- No exception is thrown.	
	4- A proactive command DISPLAY TEXT is sent		4- 91 XX
	5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called	5- Applet is triggered	
	6- It is 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	6- No exception is thrown and the handler contains the envelope call control by NAA	
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC
	7- It is checked that the TAG_ACCESS_TECHNOLOGY is the TLV selected		
	8- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	8- The contents of the envelope handler shall be the same as stored in buffer1	

Id	Description	API/CAT RE Expectation	APDU Expectation
18	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETER_CHANGED**		
	1- A event download display parameter changed envelope is sent to UICC	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() The EnvelopeHandler.findTLV() method is called with TAG_DISPLAY_PARAMETER	3- No exception is thrown.	
	4- A proactive command DISPLAY TEXT is sent		4- 91 XX
	5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called	5- Applet is triggered	
	6- It is 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	6- No exception is thrown and the handler contains the envelope call control by NAA	
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC
	7- It is checked that the TAG_DISPLAY_PARAMETER is the TLV selected		
	8- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	8- The contents of the envelope handler shall be the same as stored in buffer1	
		9- Applet sends a DECLARE SERVICE (add) proactive command with its service identifier	9- DECLARE SERVICE (add) proactive command is fetched Successful TERMINAL RESPONSE is issued

Id	Description	API/CAT RE Expectation	APDU Expectation
19	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION** 1- A event download local connection envelope is sent to UICC with the allocated service Id of Applet 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_SERVICE_RECORD 4- A proactive command DISPLAY TEXT is sent 5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called 6- It is 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 7- It is checked that the TAG_SERVICE_RECORD is the TLV selected 8- 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 NAA 8- The contents of the envelope handler shall be the same as stored in buffer1	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC

Id	Description	API/CAT RE Expectation	APDU Expectation
20	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE**		
	1- A event download network search mode change envelope is sent to UICC	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() The EnvelopeHandler.findTLV() method is called with TAG_NETWORK_SEARCH_MODE	3- No exception is thrown.	
	4- A proactive command DISPLAY TEXT is sent		4- 91 XX
	5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called	5- Applet is triggered	
	6- It is 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	6- No exception is thrown and the handler contains the envelope call control by NAA	Proactive command Display Text is fetched
	7-It is checked that the TAG_NETWORK_SEARCH_MODE is the TLV selected		The terminal Response of DISPLAY TEXT is sent to the UICC
	8- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	8- The contents of the envelope handler shall be the same as stored in buffer1	

Id	Description	API/CAT RE Expectation	APDU Expectation
21	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_BROWSING STATUS** 1- A event download browsing status envelope is sent to UICC 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_BROWSING_STATUS 4- A proactive command DISPLAY TEXT is sent 5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called 6- It is 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 7- It is checked that the TAG_BROWSING_STATUS is the TLV selected 8- 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 NAA 8- The contents of the envelope handler shall be the same as stored in buffer1	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC

Id	Description	API/CAT RE Expectation	APDU Expectation
22	**Envelope Handler integrity checks with EVENT_EXTERNAL_FILE_UPDATE under MF** 1- EF_{TARU} of the UICC file system is updated 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_FILE_LIST 4- A proactive command DISPLAY TEXT is sent 5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called 6- It is 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 7- It is checked that the TAG_FILE_LIST is the TLV selected 8- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare() 9- It is checked that the EnvelopeHandler contains the comprehension TLVs: Device Identity, File List and File Update Information with the correct value. The EnvelopeHandler does not contain the comprehension TLV AID	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 NAA 8- The contents of the envelope handler shall be the same as stored in buffer1	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC

Id	Description	API/CAT RE Expectation	APDU Expectation
23	**Envelope Handler integrity checks with EVENT_EXTERNAL_FILE_UPDATE under ADF** 1- EF_{LARU} of the ADF file system is updated 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_FILE_LIST 4- A proactive command DISPLAY TEXT is sent 5- Envelope call control by NAA is sent to UICC EnvelopeHandler.getTheHandler() method is called 6- It is 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 7- It is checked that the TAG_FILE_LIST is the TLV selected 8- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare() 9- It is checked that the EnvelopeHandler contains the comprehension TLVs: Device Identity, File List, AID and File Update Information with the correct value.	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 NAA 8- The contents of the envelope handler shall be the same as stored in buffer1	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the UICC

5.5.2.4 EnvelopeResponseHandler

Test Area Reference: Cre_Hin_Erhd.

5.5.2.4.1 Conformance requirement

5.5.2.4.1.1 Normal execution

- CRRN1: At the processToolkit invocation the TLV-List is cleared.

5.5.2.4.1.2 Parameter errors

No requirements.

5.5.2.4.1.3 Context errors

No requirements.

5.5.2.4.2 Test area files

Test Source: Test_Cre_Hin_Erhd.java.

Test Applet: Cre_Hin_Erhd_1.java.

Cap_File: Cre_Hin_Erhd.cap.

5.5.2.4.3 Test coverage

CRR Number	Test Case Number
CRRN1	1

5.5.2.4.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**At the processToolkit invocation the TLV-List is cleared** Applet1 is registered to EVENT_UNRECOGNIZED_ENVELOPE. 1-An unrecognized envelope is sent to the UICC 2- EnvelopeResponseHandler.getTheHandler()is called by the Applet1. 3- EnvelopeResponseHandler.getLength() method is called by Applet1	1- Applet1 is triggered 3- The return value shall be 0	

5.5.3 Applet Triggering

5.5.3.1 General behaviour

Test Area Reference: Cre_Apt_Genb.

5.5.3.1.1 Conformance requirement

5.5.3.1.1.1 Normal execution

- CRRN1: When a first level application is the selected application and when a Toolkit Applet is triggered the *select*() method of the Toolkit Applet shall not be launched since the Toolkit Applet itself is not selected.
- CRRN2: The CAT Runtime Environment shall only trigger a Toolkit Applet if it is in the selectable state as defined in ETSI TS 102 226 [8].
- CRRN3: When the CAT Runtime Environment has to trigger several applets on the same event, the next applet is triggered on the return of the *processToolkit()* method of the previous Toolkit Applet.

5.5.3.1.1.2 Parameter errors

No requirements.

5.5.3.1.1.3 Context errors

No requirements.

5.5.3.1.2 Test area files

Test Source: Test_Cre_Apt_Genb.java.

Test Applet: Cre_Apt_Genb_1.java.

Cap File: cre_apt_genb.cap.

5.5.3.1.3 Test coverage

CR Number	Test Case Number
CRRN1	Covered by all other tests of this clause (5.5.3)
CRRN2	1, 2, 3, 4
CRRN3	Covered by all other tests of this clause (5.5.3)

5.5.3.1.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Triggering an applet in the installed state** When installed, the applet is registered to EVENT_EVENT_DOWNLOAD_USER_ACTIVITY 1- Install the applet without making it selectable 2- Select the Applet by AID 3- An envelope event download user activity is sent to the UICC	2- Applet is not triggered	1- SW = 90 00 2- SW ≠ 90 00
2	**Triggering an applet in the make selectable state** 1- Install the applet to make it selectable 2- Select the Applet by AID3- An envelope event download user activity is sent to the UICC	2- Applet is triggered	1- SW = 90 00 2- SW = 90 00
3	**Triggering an applet in the lock state** 1- Set the applet in the lock state 2- Select the Applet by AID 3- An envelope event download user activity is sent to the UICC	2- Applet is not triggered	1- SW = 90 00 2- SW ≠ 90 00
4	**Triggering an applet in the make selectable state** 1- Set the applet in the make selectable state 2- Select the Applet by AID 3- An envelope event download user activity is sent to the UICC	2- Applet is triggered	1- SW = 90 00 2- SW = 90 00

5.5.3.2 EVENT_PROFILE_DOWNLOAD

Test Area Reference: Cre_Apt_Epdw.

5.5.3.2.1 Conformance requirement

5.5.3.2.1.1 Normal execution

- CRRN1: Upon reception of a TERMINAL PROFILE APDU command the CAT Runtime Environment shall store the terminal profile and trigger all the Toolkit Applet(s) registered to this event.
- CRRN2: The applet is not triggered by the EVENT_PROFILE_DOWNLOAD once it has deregistered from this event.
- CRRN3: The CAT Runtime Environment shall not reply busy to a Terminal Profile command.

5.5.3.2.1.2 Parameter errors

No requirements.

5.5.3.2.1.3 Context errors

No requirements.

5.5.3.2.2 Test area files

Test Source: Test_Cre_Apt_Epdw.java.

Test Applet: Cre_Apt_Epdw_1.java.

Cre_Apt_Epdw_2.java.

Cre_Apt_Epdw_3.java.

Cap_File: Cre_apt_epdw.cap.

5.5.3.2.3 Test coverage

CRR Number	Test Case Number
CRRN1	1
CRRN2	3
CRRN3	2

5.5.3.2.4 Test procedure

Id	Description	API/CAT RE 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_MENU_SELECTION. 1-Terminal Profile command is sent to UICC	1- Applet1 is triggered Applet1 finalizes 2- Applet2 is triggered Applet2 finalizes 3- Applet3 is not triggered	

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**The CAT Runtime Environment shall not reply busy to a Terminal Profile command** 1-Envelope menu selection is sent to the UICC Applet3 builds a REFRESH proactive command in UICC initialization mode 2-ProactiveHandler.send() method is called by applet3 3-Terminal Profile command is sent to the UICC Applet1 calls ToolkitRegistry.clearEvent(EVENT_PROFILE_DOWNLOAD) 4-Applet2 calls ToolkitRegistry.clearEvent(EVENT_PROFILE_DOWNLOAD) 6-Applet3 calls ToolkitRegistry.setEvent(EVENT_PROFILE_DOWNLOAD)	 1- Applet3 is triggered by the EVENT_MENU_SELECTION Applet3 is suspended until the terminal response 3- Applet1 is triggered by EVENT_PROFILE_DOWNLOAD Applet1 finalizes Applet2 is triggered by EVENT_PROFILE_DOWNLOAD Applet2 finalizes Applet3 finalizes	 2- A proactive command is sent 5- The terminal response of the proactive command is sent
3	**Deregistered applets are not triggered** Terminal Profile command is sent to the UICC	 Applet3 is triggered (Applet1 and Applet2 are not triggered)	

5.5.3.3 EVENT_MENU_SELECTION

Test Area Reference: Cre_Apt_Emse.

5.5.3.3.1 Conformance requirement

5.5.3.3.1.1 Normal execution

- CRRN1: Upon reception of an ENVELOPE (MENU SELECTION) APDU command, the CAT Runtime Environment shall only trigger the Toolkit Applet registered to the corresponding event with the associated menu identifier.

5.5.3.3.1.2 Parameter errors

No requirements.

5.5.3.3.1.3 Context errors

No requirements.

5.5.3.3.2 Test area files

Test Source: Test_Cre_Apt_Emse.java.

Test Applet: Cre_Apt_Emse_1.java.

Cre_Apt_Emse_2.java.

Cap File: Cre_apt_emse.cap.

5.5.3.3.3 Test coverage

CRR Number	Test Case Number
CRRN1	1

5.5.3.3.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration to EVENT_MENU_SELECTION and triggering** ToolkitRegistry.initMenuEntry() method is called at the installation 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=true IconQualifier=0 IconIdentifier=0 event= EVENT_MENU_SELECTION 1-ToolkitRegistry.isEventSet() is called at installation. Perform UICC initialization with the facility SET UP MENU and without the facilities SET EVENT LIST and POLL INTERVAL features 2-Item Identifier = 1 Event Menu Selection envelope is sent to the UICC with the item identifier of a menu entry of Applet1 3-Item Identifier = 2 Event Menu Selection envelope is sent to the UICC with the item identifier of a menu entry of Applet2	1- The method shall return true 2- Applet1 is triggered and Applet2 is not triggered Applet1 finalizes 3- Applet2 is triggered and Applet1 is not triggered	

5.5.3.4 EVENT_MENU_SELECTION_HELP_REQUEST

Test Area Reference: Cre_Apt_Emsh.

5.5.3.4.1 Conformance requirement

5.5.3.4.1.1 Normal execution

- CRRN1: If an ENVELOPE (MENU_SELECTION_HELP_SUPPORTED) command is received for one entry supporting help, then CAT Runtime Environment 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 CAT Runtime Environment 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 CAT Runtime Environment shall indicate to the ME that help information is available unless all the menus entries that support help are disabled.

5.5.3.4.1.2 Parameter errors

No requirements.

5.5.3.4.1.3 Context errors

No requirements.

5.5.3.4.2 Test area files

Test Source: Test_Cre_Apt_Emsh.java.

Test Applet: Cre_Apt_Emsh_1.java.

Cre_Apt_Emsh_2.java.

Cre_Apt_Emsh_3.java.

Cap File: Cre_apt_emsh.cap.

5.5.3.4.3 Test coverage

CRR Number	Test Case Number
CRRN1	1
CRRN2	1
CRRN3	2

5.5.3.4.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration to EVENT_MENU_SELECTION_HELP_REQUEST and triggering** Applet1 and Applet2 are installed ToolkitRegistry.InitMenuEntry() method is called at the installation 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 at the installation 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 at installation Perform UICC 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 UICC with item identifier 1 belonging to Applet1 4-Item identifier = 2 Menu Selection Help Request envelope is sent to the UICC with item identifier 2 belonging to Applet1 5-Item identifier = 3 Menu Selection Help Request envelope is sent to the UICC with item identifier 3 belonging to Applet2 6-Item identifier = 4 Menu Selection Help Request envelope is sent to the UICC with item identifier 4 belonging to Applet2	1- The command shall return true 2- The command shall return true 3- Applet1 is triggered and Applet2 is not triggered 4 Applet1 and Applet2 are not triggered 5- Applet2 is triggered and Applet1 is not triggered 6- Applet2 and Applet1 are not triggered	

Id	Description	API/CAT RE 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 at the installation 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 UICC 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 UICC 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 UICC 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 UICC 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 UICC shall issue a SET UP MENU proactive command with Menu Entry ID entry '06' and '07', and Help supported set to true. 5- The UICC shall issue a SET UP MENU proactive command with Menu Entry ID entry '07', and Help supported set to false

5.5.3.5 EVENT_CALL_CONTROL_BY_NAA

Test Area Reference: Cre_Apt_Eccn.

5.5.3.5.1 Conformance requirement

5.5.3.5.1.1 Normal execution

- CRRN1: The applet is triggered by the EVENT_CALL_CONTROL_BY_NAA once it has registered to this event and an Envelope Call Control by NAA is received.
- CRRN2: The applet is not triggered by the EVENT_CALL_CONTROL_BY_NAA once it has deregistered from this event.
- CRRN3: Regardless of the Toolkit Applet state the CAT Runtime Environment shall not allow more than one Toolkit Applet to be registered to this event at a time.

5.5.3.5.1.2 Parameter errors

No requirements.

5.5.3.5.1.3 Context errors

No requirements.

5.5.3.5.2 Test area files

Test Source: Test_Cre_APT_ECCN.java.

Test Applet: Cre_Apt_Eccn_1.java.

Cre_Apt_Eccn_2.java.

Cap File: Cre_apt_eccn.cap.

5.5.3.5.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3
CRRN2	3
CRRN3	See API_2_TKR_SEVTB

5.5.3.5.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applets registration to EVENT_CALL_CONTROL_BY_NAA and triggering** Applet1 is registered to EVENT_CALL_CONTROL_BY_NAA. Applet2 is registered to EVENT_MENU_SELECTION 1- An Envelope Call control by NAA is sent to the UICC	1- Applet1 is triggered	
2	**Applet deregistration and registration of the second applet to EVENT_CALL_CONTROL_BY_NAA.** 1- An Envelope menu selection is sent to the UICC Applet2 contructs a Display Text proactive command. 2- ProactiveHandler.send() method is called 3- An Envelope Call control by NAA envelope is sent to the UICC ToolkitRegistry.clearEvent() is called for EVENT_CALL_CONTROL_BY_NAA 4- TERMINAL RESPONSE of Display Text is sent to the UICC ToolkitRegistry.setEvent() method is called for EVENT_CALL_CONTROL_BY_NAA	1- Applet2 is triggered by EVENT_MENU_SELECTION. 3- Applet1 is triggered Applet1 finalizes. 4- Applet2 is resumed Applet2 finalizes	2- A proactive command Display Text is sent and applet is suspended until the terminal response
3	**Applet triggering** An Envelope Call control by NAA envelope is sent to the UICC	Applet2 is triggered. (Applet1 is not triggered)	

5.5.3.6 EVENT_TIMER_EXPIRATION

Test Area Reference: Cre_Apt_Etex.

5.5.3.6.1 Conformance requirement

5.5.3.6.1.1 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.

5.5.3.6.1.2 Parameter errors

No requirements.

5.5.3.6.1.3 Context errors

No requirements.

5.5.3.6.2 Test area files

Test Source: Test_Cre_Apt_Etex.java.

Test Applet: Cre_Apt_Etex_1.java.

Cap File: Cre_apt_etex.cap.

5.5.3.6.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.6.4 Test procedure

Id	Description	API/CAT RE 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_MENU_SELECTION. event= EVENT_TIMER_EXPIRATION 1- ToolkitRegistry.isEventSet() method is called. 2- An Envelope TIMER_EXPIRATION is sent to the UICC.	1- The method returns true 2- Applet is triggered.	
2	**Applet deregistration** Timer id=1 ToolkitRegistry.ReleaseTimer() method is called 1- An Envelope timer expiration is sent to the UICC An Envelope Menu selection is sent to the UICC ToolkitRegistry.AllocateTimer() method is called 2- An Envelope TIMER_EXPIRATION is sent to the UICC	1- Applet is not triggered 2- Applet is triggered	

5.5.3.7 EVENT_EVENT_DOWNLOAD_MT_CALL

Test Area Reference: Cre_Apt_Edmc.

5.5.3.7.1 Conformance requirement

5.5.3.7.1.1 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.

5.5.3.7.1.2 Parameter errors

No requirements.

5.5.3.7.1.3 Context errors

No requirements.

5.5.3.7.2 Test area files

Test Source: Test_Cre_Apt_Edmc.java.

Test Applet: Cre_Apt_Edmc_1.java.

Cap File: cre_apt_edmc.cap.

5.5.3.7.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.7.4 Test procedure

Id	Description	API/CAT RE 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_MENU_SELECTION event= EVENT_EVENT_DOWNLOAD_MT_CALL 1-ToolkitRegistry.isEventSet() method is called 2-An Envelope EVENT_DOWNLOAD_MT_CALL is sent to the UICC	1- The method returns true 2- Applet is triggered	
2	**Applet deregistration** event= EVENT_EVENT_DOWNLOAD_MT_CALL ToolkitRegistry.clearEvent()method is called 1-An Envelope EVENT_DOWNLOAD_MT_CALL is sent to the UICC An Envelope menu selestion is sent to the UICC event= EVENT_EVENT_DOWNLOAD_MT_CALL ToolkitRegistry.setEvent() method is called 2-An Envelope EVENT_DOWNLOAD_MT_CALL is sent to the UICC	1- Applet is not triggered 2- Applet is triggered	

5.5.3.8 EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

Test Area Reference: Cre_Apt_Edcc.

5.5.3.8.1 Conformance requirement

5.5.3.8.1.1 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.

5.5.3.8.1.2 Parameter errors

No requirements.

5.5.3.8.1.3 Context errors

No requirements.

5.5.3.8.2 Test area files

Test Source: Test_Cre_Apt_Edcc.java.

Test Applet: Cre_Apt_Edcc_1.java.

Cap File: Cre_apt_edcc.cap.

5.5.3.8.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.8.4 Test procedure

Id	Description	API/CAT RE 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_MENU_SELECTION event= EVENT_EVENT_DOWNLOAD_CALL_CONNECTED 1-ToolkitRegistry.isEventSet() method is called 2-An Envelope EVENT_DOWNLOAD_CALL_CONNECTED is sent to the UICC	1- Method returns true 2- Applet is triggered	
2	**Applet deregistration** event=EVENT_EVENT_DOWNLOAD_CALL_CONNECTED ToolkitRegistry.clearEvent()method is called 1-A call connected event dowload is sent to the UICC An Envelope menu selection is sent to the UICC Event= EVENT_EVENT_DOWNLOAD_CALL_CONNECTED ToolkitRegistry.setEvent() method is called 2-An Envelope EVENT_DOWNLOAD_CALL_CONNECTED is sent to the UICC	1- Applet is not triggered 2- Applet is triggered	

5.5.3.9 EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

Test Area Reference: Cre_Apt_Edcd.

5.5.3.9.1 Conformance requirement

5.5.3.9.1.1 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.

5.5.3.9.1.2 Parameter errors

No requirements.

5.5.3.9.1.3 Context errors

No requirements.

5.5.3.9.2 Test area files

Test Source: Test_Cre_Apt_Edcd.java.

Test Applet: Cre_Apt_Edcd_1.java.

Cap File: Cre_apt_edcd.cap.

5.5.3.9.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.9.4 Test procedure

Id	Description	API/CAT RE 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_MENU_SELECTION Event=EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED 1-ToolkitRegistry.isEventSet() method is called 2-An Envelope EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED is sent to the UICC	1- Method returns true 2- Applet is triggered	
2	**Applet deregistration** Event= EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED ToolkitRegistry.clearEvent()method is called 1-An Envelope EVENT_DOWNLOAD_CALL_DISCONNECTED is sent to the UICC An envelope menu selection is sent to the UICC Event= EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED ToolkitRegistry.setEvent() method is called 2-An Envelope EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED is sent to the UICC	1- Applet is not triggered 2- Applet is triggered	

5.5.3.10 EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

Test Area Reference: Cre_Apt_Edls.

5.5.3.10.1 Conformance requirement

5.5.3.10.1.1 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.

5.5.3.10.1.2 Parameter errors

No requirements.

5.5.3.10.1.3 Context errors

No requirements.

5.5.3.10.2 Test area files

Test Source: Test_Cre_Apt_Edls.java.

Test Applet: Cre_Apt_Edls_1.java.

Cap File: Cre_apt_edls.cap.

5.5.3.10.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.10.4 Test procedure

Id	Description	API/CAT RE 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_MENU_SELECTION Event=EVENT_EVENT_DOWNLOAD_LOCATION_STATUS 1-ToolkitRegistry.isEventSet() method is called 2-An Envelope EVENT_EVENT_DOWNLOAD_LOCATION_STATUS is sent to the UICC	1- Method returns true 2- Applet is triggered	

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**Applet deregistration** Event=EVENT_EVENT_DOWNLOAD_LOCATION_STATUS ToolkitRegistry.clearEvent()method is called 1-An Envelope EVENT_DOWNLOAD_LOCATION_STATUS is sent to the UICC An Envelope menu selection is sent to the UICC Event= EVENT_EVENT_DOWNLOAD_LOCATION_STATUS ToolkitRegistry.setEvent() method is called 2-An Envelope EVENT_DOWNLOAD_LOCATION_STATUS is sent to the UICC	1- Applet is not triggered 2- Applet is triggered	

5.5.3.11 EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

Test Area Reference: Cre_Apt_Edua

5.5.3.11.1 Conformance requirement

5.5.3.11.1.1 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.

5.5.3.11.1.2 Parameter errors

No requirements.

5.5.3.11.1.3 Context errors

No requirements.

5.5.3.11.2 Test area files

Test Source: Test_Cre_Apt_Edua.java.

Test Applet: Cre_Apt_Edua_1.java.

Cap File: Cre_apt_edua.cap.

5.5.3.11.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.11.4 Test procedure

Id	Description	API/CAT RE 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_MENU_SELECTION Event= EVENT_EVENT_DOWNLOAD_USER_ACTIVITY 1-ToolkitRegistry.isEventSet() method is called 2-An Envelope EVENT_DOWNLOAD_USER_ACTIVITY is sent to the UICC	1- Method returns true 2- Applet is triggered	
2	**Applet deregistration** Event= EVENT_EVENT_DOWNLOAD_USER_ACTIVITY ToolkitRegistry.clearEvent()method is called 1-An Envelope EVENT_DOWNLOAD_USER_ACTIVITY is sent to the UICC An Envelope menu selection is sent to the UICC Event= EVENT_EVENT_DOWNLOAD_USER_ACTIVITY ToolkitRegistry.setEvent() method is called 2-An Envelope EVENT_DOWNLOAD_USER_ACTIVITY is sent to the UICC	1- Applet is not triggered 2- Applet is triggered	

5.5.3.12 EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE

Test Area Reference: Cre_Apt_Edis.

5.5.3.12.1 Conformance requirement

5.5.3.12.1.1 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.

5.5.3.12.1.2 Parameter errors

No requirements.

5.5.3.12.1.3 Context errors

No requirements.

5.5.3.12.2 Test area files

Test Source: Test_Cre_Apt_Edis.java.

Test Applet: Cre_Apt_Edis_1.java.

Cap File: Cre_apt_edis.cap.

5.5.3.12.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.12.4 Test procedure

Id	Description	API/CAT RE 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_MENU_SELECTION Event= EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE 1-ToolkitRegistry.isEventSet() method is called 2- An Envelope EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE is sent to the UICC	1- Method returns true 2- Applet is triggered	
2	**Applet deregistration** Event=EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE ToolkitRegistry.clearEvent()method is called 1- An Envelope EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE is sent to the UICC An Envelope menu selection is sent to the UICC Event= EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE ToolkitRegistry.setEvent() method is called 2-An Envelope EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE is sent to the UICC	1- Applet is not triggered 2- Applet is triggered	

5.5.3.13 EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

Test Area Reference: Cre_Apt_Edcr.

5.5.3.13.1 Conformance requirement

5.5.3.13.1.1 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.

5.5.3.13.1.2 Parameter errors

No requirements.

5.5.3.13.1.3 Context errors

No requirements.

5.5.3.13.2 Test area files

Test Source: Test_Cre_Apt_Edcr.java.

Test Applet: Cre_Apt_Edcr_1.java.

Cap File: Cre_apt_edcr.cap.

5.5.3.13.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.13.4 Test procedure

Id	Description	API/CAT RE 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_MENU_SELECTION Event=EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS 1-ToolkitRegistry.isEventSet() method is called. 2- An Envelope EVENT_DOWNLOAD_CARD_READER_STATUS is sent to the UICC	1- Method returns true 2- Applet is triggered	
2	**Applet deregistration** Event= EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS ToolkitRegistry.clearEvent()method is called 1-An Envelope EVENT_DOWNLOAD_CARD_READER_STATUS is sent to the UICC An Envelope menu selection is sent to the UICC Event= EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS ToolkitRegistry.setEvent() method is called 2-An Envelope EVENT_DOWNLOAD_CARD_READER_STATUS is sent to the UICC	1- Applet is not triggered 2- Applet is triggered	

5.5.3.14 EVENT_UNRECOGNIZED_ENVELOPE

Test Area Reference: Cre_Apt_Euev.

5.5.3.14.1 Conformance requirement

5.5.3.14.1.1 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.

5.5.3.14.1.2 Parameter errors

No requirements.

5.5.3.14.1.3 Context errors

No requirements.

5.5.3.14.2 Test area files

Test Source: Test_Cre_Apt_Euen.java.

Test Applet: Cre_Apt_Euen_1.java.

Cap File: Cre_apt_euen.cap.

5.5.3.14.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.14.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration to EVENT_UNRECOGNIZED_ENVELOPE and triggering** Applet is registered to the EVENT_UNRECOGNIZED_ENVELOPE Event= EVENT_UNRECOGNIZED_ENVELOPE 1-ToolkitRegistry.isEventSet() method is called 2-An Envelope UNRECOGNIZED_ENVELOPE is sent to the UICC	 1- Method returns true 2- Applet is triggered	

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**Applet deregistration** Event= EVENT_UNRECOGNIZED_ENVELOPE ToolkitRegistry.clearEvent()method is called 1-An Envelope UNRECOGNIZED_ENVELOPE is sent to the UICC An Envelope menu selection is sent to the UICC Event= EVENT_UNRECOGNIZED_ENVELOPE ToolkitRegistry.setEvent() method is called 2-An Envelope UNRECOGNIZED_ENVELOPE is sent to the UICC	1- Applet is not triggered 2- Applet is triggered	

5.5.3.15 EVENT_STATUS_COMMAND

Test Area Reference: Cre_Apt_Estc.

5.5.3.15.1 Conformance requirement

5.5.3.15.1.1 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.

5.5.3.15.1.2 Parameter errors

No requirements.

5.5.3.15.1.3 Context errors

No requirements.

5.5.3.15.2 Test area files

Test Source: Test_Cre_Apt_Estc.java.

Test Applet: Cre_Apt_Estc_1.java.

Cre_Apt_Estc_2.java.

Cre_Apt_Estc_3.java.

Cap File: Cre_apt_estc.cap.

5.5.3.15.3 Test coverage

CR Number	Test Case Number
CRRN1	1, 2, 3
CRRN2	3

5.5.3.15.4 Test procedure

Id	Description	API/CAT RE 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_MENU_SELECTION. 1-A status command is sent to UICC	1- Applet1 is triggered. Applet1 finalizes 2- Applet2 is triggered. Applet2 finalizes 3- Applet3 is not triggered	
2	**Applet deregistration and registration of the third applet to EVENT_STATUS_COMMAND.** 1- An Envelope menu selection is sent to UICC Applet3 builds a Display Text. 2- ProactiveHandler.send() is called 3- A status command is sent to UICC 4- requestPollInterval() method with POLL_NO_DURATION is called 5- requestPollInterval() method with POLL_NO_DURATION is called 6- TERMINAL RESPONSE of Display Text is sent to the UICC 7- requestPollInterval() method is called	1- Applet3 is triggered. 3- Applet1 is triggered. 4- Applet1 is deregistered to EVENT_STATUS_COMMAND Applet1 finalizes Applet2 is triggered. 5- Applet2 is deregistered to EVENT_STATUS_COMMAND Applet2 finalizes 6- Applet3 is resumed 7- Applet3 is registered to EVENT_STATUS_COMMAND Applet3 finalizes	2- A proactive command Display Text is sent and applet is suspended until the terminal response
3	**Applet3 triggering** Status command is sent to UICC	Applet3 is triggered. (Applet1 and Applet2 are not triggered)	

5.5.3.16 EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION

Test Area Reference: Cre_Apt_Edlg.

5.5.3.16.1 Conformance requirement

5.5.3.16.1.1 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.

5.5.3.16.1.2 Parameter errors

No requirements.

5.5.3.16.1.3 Context errors

No requirements.

5.5.3.16.2 Test area files

Test Source: Test_Cre_Apt_Edlg.java.

Test Applet: Cre_Apt_Edlg_1.java.

Cap File: Cre_apt_edlg.cap.

5.5.3.16.3 Test coverage

CR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.16.4 Test procedure

Id	Description	API/CAT RE 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_MENU_SELECTION. Event= EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION 1-ToolkitRegistry.isEventSet() method is called 2-An Envelope EVENT_DOWNLOAD_LANGUAGE_SELECTION is sent to the UICC	1-Method returns true 2- Applet is triggered	

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**Applet deregistration** Event= EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION ToolkitRegistry.clearEvent()method is called 1-An Envelope EVENT_DOWNLOAD_LANGUAGE_SELECTION is sent to the UICC An Envelope menu selection is sent to the UICC Event= EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION ToolkitRegistry.setEvent() method is called 2-An Envelope EVENT_DOWNLOAD_LANGUAGE_SELECTION is sent to the UICC	1- Applet is not triggered 2- Applet is triggered	

5.5.3.17 EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION

Test Area Reference: Cre_Apt_Edbt.

5.5.3.17.1 Conformance requirement

5.5.3.17.1.1 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.

5.5.3.17.1.2 Parameter errors

No requirements.

5.5.3.17.1.3 Context errors

No requirements.

5.5.3.17.2 Test area files

Test Source: Test_Cre_Apt_Edbt.java.

Test Applet: Cre_Apt_Edbt_1.java.

Cap File: Cre_apt_edbt.cap.

5.5.3.17.3 Test coverage

CR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.17.4 Test procedure

Id	Description	API/CAT RE 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_MENU_SELECTION Event= EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION 1-ToolkitRegistry.isEventSet() method is called 2-An Envelope EVENT_DOWNLOAD_BROWSER_TERMINATION is sent to the UICC	1-Method returns true 2- Applet is triggered	
2	**Applet deregistration** Event= EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION ToolkitRegistry.clearEvent()method is called 1-An Envelope EVENT_DOWNLOAD_BROWSER_TERMINATION is sent to the UICC An Envelope menu selection is sent to the UICC Event= EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION ToolkitRegistry.setEvent() method is called 2-An Envelope EVENT_DOWNLOAD_BROWSER_TERMINATION is sent to the UICC	1- Applet is not triggered 2- Applet is triggered	

5.5.3.18 EVENT_FIRST_COMMAND_AFTER_ATR

Test Area Reference: Cre_Apt_Efca.

5.5.3.18.1 Conformance requirement

5.5.3.18.1.1 Normal execution

- CRRN1: The applet is triggered by the EVENT_FIRST_COMMAND_AFTER_ATR once it has registered to this event and upon the reception of the first APDU after the ATR, before the Status Word of the processed command has been sent back by the UICC.
- CRRN2: The applet is not triggered by the EVENT_FIRST_COMMAND_AFTER_ATR once it has deregistered from this event.
- CRRN3: If the first APDU received is a toolkit applet triggering APDU (e.g. TERMINAL PROFILE), the toolkit applets registered to the EVENT_FIRST_COMMAND_AFTER_ATR event shall be triggered first.

5.5.3.18.1.2 Parameter errors

No requirements.

5.5.3.18.1.3 Context errors

No requirements.

5.5.3.18.2 Test area files

Test Source: Test_Cre_Apt_Efca.java.

Test Applet: Cre_Apt_Efca_1.java.

Cre_Apt_Efca_2.java.

Cre_Apt_Efca_3.java.

Cre_Apt_Efca_4.java.

Cre_Apt_Efca_5.java.

Cap File: Cre_apt_efca.cap.

5.5.3.18.3 Test coverage

CR Number	Test Case Number
CRRN1	1, 2, 3, 4
CRRN2	3
CRRN3	1, 4

5.5.3.18.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applets registration to EVENT_FIRST_COMMAND_AFTER_ATR and triggering** Applet1 is registered to the EVENT_FIRST_COMMAND_AFTER_ATR Applet2 is registered to the EVENT_PROFILE_DOWNLOAD Applet3 is registered to EVENT_MENU_SELECTION 1- Terminal Profile command is sent to the UICC 2- Applet1 deregisters from EVENT_FIRST_COMMAND_AFTER_ATR 3- Applet2 deregisters from EVENT_PROFILE_DOWNLOAD 4- An Envelope menu selection is sent to the UICC 5- Applet3 calls setEvent() on event EVENT_FIRST_COMMAND_AFTER_ATR	1- Applet1 is triggered by EVENT_FIRST_COMMAND_AFTER_ATR Applet1 finalizes Applet2 is triggered by EVENT_PROFILE_DOWNLOAD Applet2 finalizes Applet3 is not triggered 4- Applet3 is triggered.	

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**Deregistered applets are not triggered** 1-Reset then Terminal Profile command is sent to the UICC 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_ATR and EVENT_PROFILE_DOWNLOAD** Applet4 is installed, with the same priority level as Applet3 1-Reset then Terminal Profile command is sent to the UICC 2- Delete all applets	 1- Applet4 is triggered by EVENT_FIRST_COMMAND_AFTER_ATR Applet4 finalizes Applet3 is triggered by EVENT_FIRST_COMMAND_AFTER_ATR Applet3 finalizes Applet4 is triggered by EVENT_PROFILE DOWNLOAD Applet4 finalizes Applet3 is triggered by EVENT_PROFILE_DOWNLOAD Applet3 finalizes	
4	**Check that the applet is triggered before the first status word is sent** 1- Install Applet5 Applet5 is registered with two entries in the menu entries list Applet5 is also registered to EVENT_FIRST_COMMAND_AFTER_ATR 2- Reset then Terminal Profile command is sent to the UICC 3- Applet5 disables a menu entry	 2- Applet5 is triggered	 3-The SETUP MENU proactive command is fetched. There is only one item for Applet5

5.5.3.19 EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

Test Area Reference: Cre_Apt_Edda.

5.5.3.19.1 Conformance requirement

5.5.3.19.1.1 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 with General Result ="0x0X", 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 with General Result ="0x0X", the framework shall release the channel identifier contained in the command.

5.5.3.19.1.2 Parameter errors

No requirements.

5.5.3.19.1.3 Context errors

No requirements.

5.5.3.19.2 Test area files

Test Applet: Test_Cre_Apt_Edda.java.

Test Applet: Cre_Apt_Edda_1.java.

Cap File: Cre_apt_edda.cap.

5.5.3.19.3 Test coverage

CR Number	Test Case Number
CRRN1	2
CRRN2	1, 4, 5
CRRN3	1
CRRN4	3

5.5.3.19.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE**		
	Applet1 is registered to Event Menu selection. 1- An Envelope menu selection is sent to the UICC	1- Applet1 is triggered by the envelope menu selection	
	2- Applet calls setEvent() with the event EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE	2- Applet1 finalizes	
	3- An envelope Event Download Data Available is sent to the UICC Channel Status = 81 00	3- Applet1 is not triggered	
	4- An Envelope menu selection is sent to the UICC	4- Applet1 is triggered by the envelope menu selection	
	5- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init() method		
	6- send() method is called to register to this event		6- OPEN CHANNEL proactive command is fetched Unsuccessful TERMINAL RESPONSE of OPEN CHANNEL is sent to the UICC with General Result = 0x10
		7- Applet1 finalizes	
	8- An envelope Event Download Data Available is sent to the UICC with Channel Status = 01 00	8- Applet1 is not triggered	
	9- An Envelope menu selection is sent to the UICC	9- Applet1 is triggered by the envelope menu selection	
	10- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init() method		
	11- send() method is called to register to this event		11- OPEN CHANNEL proactive command is fetched. Successful TERMINAL RESPONSE of OPEN CHANNEL is sent to the UICC with Channel Id = 01 with General Result = 0x00
		12- Applet1 finalizes	
2	**Applet triggering to EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE**		
	1- An envelope Event Download Data Available is sent to the UICC Channel Status = 81 00	1- Applet1 is triggered	

Id	Description	API/CAT RE Expectation	APDU Expectation
3	**Applet deregistration to EVENT_EVENT_ DOWNLOAD_DATA_ AVAILABLE** 1- An Envelope menu selection is sent to the UICC 2- Applet1 initializes and sends an OPEN CHANNEL proactive command 3- Applet1 builds a CLOSE CHANNEL Proactive Command calling ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods 4- An envelope Event Download Data Available is sent to the UICC Channel Status = 82 00 5- Applet1 builds a CLOSE CHANNEL Proactive Command calling ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods.	1- Applet1 is triggered 4- Applet1 is triggered 6- Applet1 finalize	2- OPEN CHANNEL proactive command is fetched Successful terminal response is sent, with channelId=02 with General Result = 0x01 3- CLOSE CHANNEL proactive command is fetched Unsuccessful TERMINAL RESPONSE of CLOSE CHANNEL is sent to the UICC with General Result = 0X20 5- CLOSE CHANNEL proactive command is fetched Successful TERMINAL RESPONSE of CLOSE CHANNEL is sent to the UICC with Channel Id = 02 with General Result = 0X02
4	**Applet triggering to EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE** 1- An envelope Event Download Data Available is sent to the UICC Channel Status = 82 00	1- Applet1 is not triggered	
5	**Applet1 not triggered after a reset** 1- Applet1 is triggered by an envelope menu selection 2- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init() method 3- send() method is called to register to this event 4- isEventSet() method is called 5- Reset the card 6- An envelope Event Download Data Available is sent to the UICC Channel Status = 82 00	4- returns true 6- Applet1 is not triggered	3- OPEN CHANNEL proactive command is fetched Successful TERMINAL RESPONSE of OPEN CHANNEL is sent to the UICC with Channel Id = 02 with General Result = 0X00

5.5.3.20 EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS

Test Area Reference: Cre_Apt_Edcs.

5.5.3.20.1 Conformance requirement

5.5.3.20.1.1 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').

5.5.3.20.1.2 Parameter errors

No requirements.

5.5.3.20.1.3 Context errors

No requirements.

5.5.3.20.2 Test area files

Test Source: Test_Cre_Apt_Edcs.java.

Test Applet: Cre_Apt_Edcs_1.java.

Cap File: Cre_apt_edcs.cap.

5.5.3.20.3 Test coverage

CR Number	Test Case Number
CRRN1	2
CRRN2	1, 4, 5
CRRN3	1
CRRN4	3

5.5.3.20.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS** Applet1 is registered to Event Menu Selection 1- An Envelope menu selection is sent to the UICC 2-The applet calls setEvent() with EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS 3- An envelope Event Download Channel Status is sent to the UICC Channel Status = 81 00 4- An Envelope menu selection is sent to the UICC 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 UICC with Channel Status = 01 00 9- An Envelope menu selection is sent to the UICC 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 the envelope menu selection Applet1 finalizes 3- Applet1 is not triggered 4- Applet1 is triggered by the envelope menu selection 7- Applet finalizes 8- Applet1 is not triggered 9- Applet1 is triggered by the envelope menu selection 12- Applet1 finalizes	6- OPEN CHANNEL proactive command is fetched Unsuccessful TERMINAL RESPONSE of OPEN CHANNEL is sent to the UICC with General Result = 0x10 11- OPEN CHANNEL proactive command is fetched Successful TERMINAL RESPONSE of OPEN CHANNEL is sent to the UICC with Channel Id = 01 with General Result = 0x00
2	**Applet triggering to EVENT_EVENT_DOWNLOAD_CHANNEL STATUS** 1- An envelope Event Download Channel Status is sent to the UICC Channel Status = 81 00	1- Applet1 is triggered	

Id	Description	API/CAT RE Expectation	APDU Expectation
3	**Applet deregistration to EVENT_EVENT_ DOWNLOAD_CHANNEL STATUS** 1- An Envelope menu selection is sent to the UICC 2-Applet1 initializes and sends an OPEN CHANNEL proactive command 3- Applet1 builds a CLOSE CHANNEL Proactive Command calling ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods 4-An envelope Event Download Channel Status is sent to the UICC Channel Status = 82 00 5- Applet1 builds a Close Channel Proactive Command calling ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods	1- Applet1 is triggered 4- The applet is triggered 6- Applet1 finalizes	2- OPEN CHANNEL proactive command is fetched Successful terminal response is sent, with channel Id=02 with General Result = 0x01 3-CLOSE CHANNEL proactive command is fetched Unsuccessful TERMINAL RESPONSE of CLOSE CHANNEL is sent to the UICC with General Result = 0X20 5- CLOSE CHANNEL proactive command is fetched Successful TERMINAL RESPONSE of CLOSE CHANNEL is sent to the UICC with Channel Id = 02 with General Result = 0X02
4	**Applet triggering to EVENT_EVENT_DOWNLOAD_CHANNEL STATUS** 1- An envelope Event Download Channel Status is sent to the UICC Channel Status = 82 00	1- Applet1 is not triggered	
5	**Applet1 not triggered after a reset** 1- Applet1 is triggered by an envelope menu selection 2- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init() method 3- send() method is called to register to this event 4- isEventSet() method is called 5- Reset the card 6- An envelope Event Download Channel Status is sent to the UICC Channel Status = 82 00	4- returns true 6- Applet1 is not triggered	3- OPEN CHANNEL proactive command is fetched Successful TERMINAL RESPONSE of OPEN CHANNEL is sent to the UICC with Channel Id = 02 with General Result = 0X00

5.5.3.21 EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE

Test Area Reference: Cre_Apt_Edat.

5.5.3.21.1 Conformance requirement

5.5.3.21.1.1 Normal execution

- CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE once it has registered to this event and an Envelope Event DownLoad Access Technology Change is received.
- CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE once it has deregistered from this event.

5.5.3.21.1.2 Parameter errors

No requirements.

5.5.3.21.1.3 Context errors

No requirements.

5.5.3.21.2 Test area files

Test Source: Test_Cre_Apt_Edat.java.

Test Applet: Cre_Apt_Edat_1.java.

Cap File: Cre_apt_edat.cap.

5.5.3.21.3 Test coverage

CR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.21.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD ACCESS_TECHNOLOGY_CHANGE and triggering** Applet is registered to the EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE and to EVENT_MENU_SELECTION Event= EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE 1- ToolkitRegistry.isEventSet() method is called 2- An Envelope EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE is sent to the UICC	1- Method returns true 2- Applet is triggered	

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**Applet deregistration** Event= EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE ToolkitRegistry.clearEvent()method is called 1- An Envelope EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE is sent to the UICC 2- An Envelope menu selection is sent to the UICC Event= EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE ToolkitRegistry.setEvent() method is called 3- An Envelope EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE is sent to the UICC	1- Applet is not triggered 2- Applet is triggered 3- Applet is triggered	

5.5.3.22 EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED

Test Area Reference: Cre_Apt_Eddp.

5.5.3.22.1 Conformance requirement

5.5.3.22.1.1 Normal execution

- CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED once it has registered to this event and an Envelope Event DownLoad Display Parameters Changed is received.
- CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED once it has deregistered from this event.

5.5.3.22.1.2 Parameter errors

No requirements.

5.5.3.22.1.3 Context errors

No requirements.

5.5.3.22.2 Test area files

Test Source: Test_Cre_Apt_Eddp.java.

Test Applet: Cre_Apt_Eddp_1.java.

Cap File: Cre_apt_eddp.cap.

5.5.3.22.3 Test coverage

CR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.22.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD DISPLAY_PARAMETERS_CHANGED and triggering** Applet is registered to the EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED and to EVENT_MENU_SELECTION Event= EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED 1- ToolkitRegistry.isEventSet() method is called 2- An Envelope EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED is sent to the UICC	1- Method returns true 2- Applet is triggered	
2	**Applet deregistration** Event= EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED ToolkitRegistry.clearEvent()method is called 1- An Envelope EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED is sent to the UICC 2- An Envelope menu selection is sent to the UICC Event= EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED ToolkitRegistry.setEvent() method is called 3- An Envelope EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETERS_CHANGED is sent to the UICC	1- Applet is not triggered 2- Applet is triggered 3- Applet is triggered	

5.5.3.23 EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION

Test Area Reference: Cre_Apt_Edlc.

5.5.3.23.1 Conformance requirement

5.5.3.23.1.1 Normal execution

- CRRN1: For EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION, the framework shall only trigger the applet registered to this event with the associated service identifier.
- CRRN2: The registration to the EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION is effective once the toolkit applet has issued a successful DECLARE SERVICE (add) proactive command, and valid till the first successful DECLARE SERVICE (delete) with the corresponding service identifier or the end of the card session.

5.5.3.23.1.2 Parameter errors

No requirements.

5.5.3.23.1.3 Context errors

No requirements.

5.5.3.23.2 Test area files

Test Source: Test_Cre_Apt_Edlc.java.

Test Applet: Cre_Apt_Edlc_1.java.

Cap File: Cre_apt_edlc.cap.

5.5.3.23.3 Test coverage

CR Number	Test Case Number
CRRN1	1, 2
CRRN2	3, 4, 5

5.5.3.23.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD LOCAL_CONNECTION** Applet is registered to the EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION using the allocateServiceIdentifier() method and to EVENT_MENU_SELECTION using the initMenuEntry() method. 1- An envelope menu selection is sent to the UICC Event= EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION 2- ToolkitRegistry.isEventSet() method is called 3- An Envelope EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION is sent to the UICC with the service identifier of Applet 4- An envelope menu selection is sent to the UICC 5- Applet builds and sends a DECLARE SERVICE (add) proactive command with its service ID to register to this event. 6- An Envelope EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION is sent to the UICC with the service identifier of Applet 7- An envelope menu selection is sent to the UICC 8- Applet builds and sends a DECLARE SERVICE (add) proactive command with its service ID to register to this event.	1- Applet is triggered 2- Method returns true 3- Applet is not triggered 4- Applet is triggered by the envelope menu selection 5- Applet finalizes 6- Applet is not triggered 7- Applet is triggered by the envelope menu selection 9- Applet finalizes	5- DECLARE SERVICE (add) proactive command is fetched Unsuccessful TERMINAL RESPONSE is sent to the UICC with General Result = 0x20 8- DECLARE SERVICE proactive command is fetched Successful TERMINAL RESPONSE is sent to the UICC with General Result = 0x00
2	**Applet triggering to EVENT_EVENT_DOWNLOAD LOCAL_CONNECTION** 1- An envelope Event Download local connection is sent to the UICC with the service identifier of the Applet	1- Applet is triggered	

Id	Description	API/CAT RE Expectation	APDU Expectation
3	**Applet deregistration to EVENT_EVENT_DOWNLOAD LOCAL_CONNECTION with proactive command**		
	1- An Envelope menu selection is sent to the UICC	1- Applet1 is triggered	
	2- Applet initializes and sends a DECLARE SERVICE (delete) proactive command with the service identifier of Applet		2- DECLARE SERVICE (delete) proactive command is fetched Unsuccessful TERMINAL RESPONSE is sent to the UICC with General Result = 0X20
		Applet finalizes	
	3- An envelope Event Download local connection is sent to the UICC with the service identifier of the Applet	3- Applet is triggered	
	4-Applet initializes and sends a DECLARE SERVICE (delete) proactive command with the service identifier of Applet		4- DECLARE SERVICE (delete) proactive command is fetched Successful TERMINAL RESPONSE is sent to the UICC with General Result = 0X00
		5- Applet finalizes	
4	**Applet triggering to EVENT_EVENT_DOWNLOAD LOCAL_CONNECTION**		
	1- An envelope Event Download local connection is sent to the UICC with the service identifier of the Applet	1- Applet is not triggered	
5	**Applet1 not triggered after a reset**		
	1- Applet1 is triggered by an envelope menu selection		
	2- Applet builds and sends a DECLARE SERVICE (add) proactive command with its service ID to register to this event		2- DECLARE SERVICE (add) proactive command is fetched Successful TERMINAL RESPONSE is sent to the UICC with General Result = 0x00
	3- An envelope Event Download local connection is sent to the UICC with the service identifier of the Applet	3- Applet is triggered	
	4- Reset the card		
	5- An envelope Event Download local connection is sent to the UICC with the service identifier of the Applet	5- Applet is not triggered	

5.5.3.24 EVENT_APPLICATION_DESELECT

Test Area Reference: Cre_Apt_Eade.

5.5.3.24.1 Conformance requirement

5.5.3.24.1.1 Normal execution

- CRRN1: The applet is triggered by the EVENT_APPLICATION_DESELECT once it has registered to this event and once an application session is terminated (as described in ETSI TS 102 221 [5]).
- CRRN2: The applet is not triggered by the EVENT_APPLICATION_DESELECT once it has deregistered from this event.
- CRRN3: The AID of the deselected application is available to the Toolkit Applet in the *EnvelopeHandler*, as an AID Comprehension TLV data object as defined in the ETSI TS 102 223 [6].

5.5.3.24.1.2 Parameter errors

No requirements.

5.5.3.24.1.3 Context errors

No requirements.

5.5.3.24.2 Test area files

Test Source: Test_Cre_Apt_Eade.java.

Test Applet: Cre_Apt_Eade_1.java.

Cap File: Cre_apt_eade.cap.

5.5.3.24.3 Test coverage

CR Number	Test Case Number
CRRN1	1, 2
CRRN2	2
CRRN3	1, 2

5.5.3.24.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration to EVENT_APPLICATION_DESELECT and triggering** Applet is registered to the EVENT_APPLICATION_DESELECT and to EVENT_MENU_SELECTION Event= EVENT_APPLICATION_DESELECT 1- ToolkitRegistry.isEventSet() method is called 2- Select for activation ADF1 3- Select for activation ADF2 4- Select for termination ADF2	 1- Method returns true 2- Applet is not triggered 3- Applet is triggered The envelope handler contains the AID of ADF1 4- Applet is triggered The envelope handler contains the AID of ADF2	
2	**Applet deregistration** Event= EVENT_APPLICATION_DESELECT ToolkitRegistry.clearEvent() method is called Perform UICC initialization 1- Select for activation ADF1 2- Select for activation ADF2 3- Select for termination ADF2 4- An Envelope menu selection is sent to the UICC Event= EVENT_APPLICATION_DESELECT ToolkitRegistry.setEvent() method is called Perform UICC initialization 5- Select for activation ADF1 6- Select for activation ADF2 7- Select for termination ADF2	 2- Applet is not triggered 3- Applet is not triggered 4- Applet is triggered 5- Applet is not triggered 6- Applet is triggered The envelope handler contains the AID of ADF1 7- Applet is triggered The envelope handler contains the AID of ADF2	

5.5.3.25 EVENT_PROACTIVE_HANDLER_AVAILABLE

Test Area Reference: Cre_Apt_Epha.

5.5.3.25.1 Conformance requirement

5.5.3.25.1.1 Normal execution

- CRRN1: The applet is triggered by the EVENT_PROACTIVE_HANDLER_AVAILABLE once it has registered to this event and once the *ProactiveHandler* is available and all the Toolkit Applets registered to the previous event have been triggered and have returned from the *processToolkit()* invocation.
- CRRN2: The applet is not triggered by the EVENT_PROACTIVE_HANDLER_AVAILABLE once it has deregistered from this event.
- CRRN3: When the Toolkit Applet is triggered it is automatically deregistered by the CAT Runtime Environment.
- CRRN4: If a CAT session ends prior to an Applet triggering, the Applet will be triggered at the next CAT session.

5.5.3.25.1.2 Parameter errors

No requirements.

5.5.3.25.1.3 Context errors

No requirements.

5.5.3.25.2 Test area files

Test Source: Test_Cre_Apt_Epha.java.

Test Applet: Cre_Apt_Epha_1.java.

Cre_Apt_Epha_2.java.

Cap File: Cre_apt_epha.cap.

5.5.3.25.3 Test coverage

CR Number	Test Case Number
CRRN1	1
CRRN2	1
CRRN3	1
CRRN4	2

5.5.3.25.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration to EVENT_PROACTIVE_HANDLER_AVAILABLE, triggering and automatic deregistration** Applet1 is registered to EVENT_MENU_SELECTION Applet2 is registered to EVENT_MENU_SELECTION and EVENT_EVENT_DOWNLOAD_USER_ACTIVITY 1- Applet1 is triggered by an envelope menu selection 1.1- ToolkitRegistry.setEvent() method is called with Event = EVENT_PROACTIVE_HANDLER_AVAILABLE 1.2- ToolkitRegistry.setEvent() method is called with Event = EVENT_EVENT_DOWNLOAD_USER_ACTIVITY 1.3- ToolkitRegistry.isEventSet() method is called with Event = EVENT_PROACTIVE_HANDLER_AVAILABLE Event= EVENT_PROACTIVE_HANDLER_AVAILABLE 3- ToolkitRegistry.isEventSet() method is called by Applet1 4- An envelope event download user activity is sent to the UICC Event= EVENT_PROACTIVE_HANDLER_AVAILABLE 5- ToolkitRegistry.setEvent() method is called by Applet1 6- ToolkitRegistry.isEventSet() method is called by Applet1 Event= EVENT_PROACTIVE_HANDLER_AVAILABLE 7- ToolkitRegistry.setEvent() method is called by Applet2 8- ToolkitRegistry.isEventSet() method is called by Applet2 Event= EVENT_PROACTIVE_HANDLER_AVAILABLE 9- ToolkitRegistry.isEventSet() method is called by Applet1 Event= EVENT_PROACTIVE_HANDLER_AVAILABLE 10- ToolkitRegistry.isEventSet() method is called by Applet1	1- Applet1 is triggered 1.1- No exception is thrown 1.2- No exception is thrown 1.3- Method returns TRUE 1.4- 2- Applet1 is triggered by EVENT_PROACTIVE_HANDLER_AVAILABLE 3- Method returns FALSE 4- Applet1 is triggered by the envelope 5- No exception is thrown 6- Method returns TRUE Applet1 finalizes Applet2 is triggered by the envelope 7- No exception is thrown 8- Method returns TRUE Applet2 finalizes Applet1 is triggered by EVENT_PROACTIVE_HANDLER_AVAILABLE 9- Method returns FALSE Applet1 finalizes Applet2 is triggered by EVENT_PROACTIVE_HANDLER_AVAILABLE 10- Method returns FALSE Applet2 finalizes	2- SW = 9000 is returned to the envelope

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**Applet triggering between 2 CAT sessions** 1- Applet1 is triggered by an envelope menu selection 1.1- Applet1 prepares and sends a Display Text proactive command 2- Fetch the proactive command 3- Applet2 is triggered by an envelope menu selection 3.1 Applet2 registers to EVENT_PROACTIVE_HANDLER_AVAILABLE then finalizes 4- Reset the card and send the profile download command	1- Applet1 is triggered 1.1 Applet1 is suspended 3- Applet2 is triggered 3.1- No exception is thrown 4- Applet2 is triggered by event EVENT_PROACTIVE_HANDLER_AVAILABLE	

5.5.3.26 EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE

Test Area Reference: Cre_Apt_Edns.

5.5.3.26.1 Conformance requirement

5.5.3.26.1.1 Normal execution

- CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE once it has registered to this event and an Envelope Event Event Download Network Search Mode Change is received.
- CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE once it has deregistered from this event.

5.5.3.26.1.2 Parameter errors

No requirements.

5.5.3.26.1.3 Context errors

No requirements.

5.5.3.26.2 Test area files

Test Source: Test_Cre_Apt_Edns.java.

Test Applet: Cre_Apt_Edns_1.java.

Cap File: Cre_apt_edns.cap.

5.5.3.26.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.26.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE and triggering** Applet is registered to the EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE and to EVENT_MENU_SELECTION event= EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE 1-ToolkitRegistry.isEventSet() method is called 2-An Envelope EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE is sent to the UICC	1- Method returns true 2- Applet is triggered	
2	**Applet deregistration** event= EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE ToolkitRegistry.clearEvent()method is called 1-A network search mode change event dowload is sent to the UICC 2-An Envelope menu selection is sent to the UICC Event= EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE ToolkitRegistry.setEvent() method is called 3-An Envelope EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE is sent to the UICC	1- Applet is not triggered 2- Applet is triggered 3- Applet is triggered	

5.5.3.27 EVENT_EVENT_DOWNLOAD_BROWSING_STATUS

Test Area Reference: Cre_Apt_Edbs.

5.5.3.27.1 Conformance requirement

5.5.3.27.1.1 Normal execution

- CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_BROWSING_STATUS once it has registered to this event and an Envelope Event Event Download Browsing Status is received.
- CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_BROWSING_STATUS once it has deregistered from this event.

5.5.3.27.1.2 Parameter errors

No requirements.

5.5.3.27.1.3 Context errors

No requirements.

5.5.3.27.2 Test area files

Test Source: Test_Cre_Apt_Edbs.java.

Test Applet: Cre_Apt_Edbs_1.java.

Cap File: Cre_apt_edbs.cap.

5.5.3.27.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	2

5.5.3.27.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration to EVENT_EVENT_DOWNLOAD_BROWSING_STATUS and triggering** Applet is registered to the EVENT_EVENT_DOWNLOAD_BROWSING_STATUS and to EVENT_MENU_SELECTION event= EVENT_EVENT_DOWNLOAD_BROWSING_STATUS 1-ToolkitRegistry.isEventSet() method is called 2-An Envelope EVENT_EVENT_DOWNLOAD_BROWSING_STATUS is sent to the UICC	1- Method returns true 2- Applet is triggered	
2	**Applet deregistration** event= EVENT_EVENT_DOWNLOAD_BROWSING_STATUS ToolkitRegistry.clearEvent()method is called 1-A browsing status event dowload is sent to the UICC An Envelope menu selection is sent to the UICC Event= EVENT_EVENT_DOWNLOAD_BROWSING_STATUS ToolkitRegistry.setEvent() method is called 2-An Envelope EVENT_EVENT_DOWNLOAD_BROWSING_STATUS is sent to the UICC	1 – Applet is not triggered 2- Applet is triggered	

5.5.3.28 EVENT_EXTERNAL_FILE_UPDATE

Test Area Reference: Cre_Apt_Eefu

5.5.3.28.1 Conformance requirement

5.5.3.28.1.1 Normal execution

- CRRN1: The applet is triggered by the EVENT_EXTERNAL_FILE_UPDATE once it has registered to this event and a successful execution of an UPDATE BINARY or UPDATE RECORD or INCREASE APDU command (sent by the Terminal and received by the UICC on the I/O line) as defined in TS 102 221 [5] is performed on the associated updated file.
- CRRN2: An applet shall only be triggered once per command.
- CRRN3: The registration to this event is effective once the applet has successfully called a method registerFileEvent(…).
- CRRN4: The applet is not triggered by the EVENT_EXTERNAL_FILE_UPDATE once it has deregistered from this event.
- CRRN5: The deregistration for a particular file to this event is effective once the Applet has successfully called the method deregisterFileEvent(…).
- CRRN6: A call to the method *clearEvent*(EVENT_EXTERNAL_FILE_UPDATE) clears the event EVENT_EXTERNAL_FILE_UPDATE from the ToolkitRegistry of the Applet i.e. the Applet is no longer triggered when a file is updated.

5.5.3.28.1.2 Parameter errors

No requirements.

5.5.3.28.1.3 Context errors

- CRRC1: The applet shall not be triggered if the UPDATE BINARY or UPDATE RECORD or INCREASE APDU command are not sent by the Terminal and received by the UICC on the I/O line.

5.5.3.28.2 Test area files

Test Source: Test_Cre_Apt_Eefu.java.

Test Applet: Cre_Apt_Eefu_1.java.

Cap File: Cre_apt_eefu.cap.

5.5.3.28.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	1, 2
CRRN3	1, 2
CRRN4	2
CRRN5	2
CRRN6	3
CRRC1	See ETSI TS 131 213 [12]

5.5.3.28.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet registration to EVENT_EXTERNAL_FILE_UPDATE and triggering** Applet is registered to the EVENT_MENU_SELECTION 1- Send an envelope menu selection Event=EVENT_EXTERNAL_FILE_UPDATE 2-ToolkitRegistry.isEventSet() method is called 3- ToolkitRegistry.registerFileEvent() method with fileview is called to register to EF_{TARU} 4- ToolkitRegistry.isEventSet() method is called 5- ToolkitRegistry.registerFileEvent() method with paths of EF_{LARU} and EF_{CARU} is called 6- Update EF_{TARU} 7- Update EF_{LARU} 8- Increase EF_{CARU}	1- Applet is triggered 2- Method returns false 3- No exception is thrown 4- Method returns true 5- No exception is thrown Applet finalizes 6- Applet is triggered 7- Applet is triggered 8- Applet is triggered	
2	**Applet deregistration – case 1** 1-Update EF_{TARU} 2- ToolkitRegistry.deregisterFileEvent() method with fileview is called to deregister EF_{CARU} Event=EVENT_EXTERNAL_FILE_UPDATE 3- ToolkitRegistry.isEventSet() method is called 4- Increase EF_{CARU} 5- Update EF_{TARU} 6- ToolkitRegistry.deregisterFileEvent() method with path of EF_{LARU} is called 7- ToolkitRegistry.isEventSet() method is called 8- Update EF_{LARU} 9- Update EF_{TARU} 10- ToolkitRegistry.deregisterFileEvent() method with path of EF_{TARU} is called 11- ToolkitRegistry.isEventSet() method is called 12- Update EF_{TARU}	1- Applet is triggered 2- No exception is thrown 3- Method returns true 4- Applet is not triggered 5- Applet is triggered 6- No exception is thrown 7- Method returns true 8- Applet is not triggered 9- Applet is triggered 10- No exception is thrown 11- Method returns false 12- Applet is not triggered	

Id	Description	API/CAT RE Expectation	APDU Expectation
3	**Applet deregistration – case 2** 1- Send an envelope menu selection 2- ToolkitRegistry.registerFileEvent() method with fileview is called to register to EF_{TARU} 3- ToolkitRegistry.registerFileEvent() method with paths EF_{LARU} and EF_{CARU} is called 4- Update EF_{TARU} 5- Update EF_{LARU} 6- Increase EF_{CARU} Event=EVENT_EXTERNAL_FILE_UPDATE 7- ToolkitRegistry.clearEvent() method is called 8- ToolkitRegistry.isEventSet() method is called 9- Update EF_{TARU} 10- Update EF_{LARU} 11- Increase EF_{CARU} 12- Restore EF_{CARU}, EF_{TARU} and EF_{LARU}	1- Applet is triggered 2- No exception is thrown 3- No exception is thrown 4- Applet is triggered 5- Applet is triggered 6- Applet is triggered 7- No exception is thrown 8- Method returns false 9- Applet is not triggered 10- Applet is not triggered 11- Applet is not triggered	

5.5.4 Proactive Command Sending by the CAT Runtime Environment

5.5.4.1 System Proactive Commands

Test Area Reference: Cre_Pcs_Spco.

5.5.4.1.1 Conformance requirement

5.5.4.1.1.1 Normal execution

- CRRN1: During a CAT session the CAT Runtime Environment shall send a SET UP MENU system proactive command whenever a menu entry is modified, added or removed.
- CRRN2: The CAT Runtime Environment shall use the data of the EFsume file when issuing the SET UP MENU proactive command.
- CRRN3: During a CAT session the CAT Runtime Environment shall send a SET UP MENU system proactive command whenever the EF_{SUME} file under the $DF_{TELECOM}$ file is updated as defined in TS 102 222 [7].
- CRRN4: At the beginning of a CAT session, the CAT Runtime Environment shall send a SET UP MENU system proactive command, if at least one menu entry is registered and enabled by a selectable Toolkit Applet.
- CRRN5: At the beginning of a CAT session, the CAT Runtime Environment shall send a SET UP EVENT LIST system proactive command, if at least one event is registered by a selectable Toolkit Applet.
- CRRN6: During a CAT session the CAT Runtime Environment shall send a SET UP EVENT LIST system proactive command whenever the registered event list is changed.
- CRRN7: At the beginning of a CAT session, the CAT Runtime Environment shall send a POLL INTERVAL system proactive command, if at least one Toolkit Applet has requested a poll interval duration.
- CRRN8: During a CAT session the CAT Runtime Environment shall send a POLL INTERVAL or POLLING OFF system proactive command whenever the system poll interval duration is changed.
- CRRN9: The CAT Runtime Environment shall send its system proactive command(s) 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.

- CRRN10: The system proactive command shall only contain information from Toolkit Applets that are in the selectable state.
- CRRN11: If help is available for at least one Menu Entry inserted in the SET UP MENU system proactive command the CAT Runtime Environment shall indicate to the terminal that help information is available.
- CRRN12: If help is not available for all Menu Entries inserted in the SET UP MENU system proactive command the CAT Runtime Environment shall not indicate to the terminal that help information is available.
- CRRN13: If a text attribute different from the default format is provided for at least one Menu Entry, the SET UP MENU system proactive command shall contain the item text attribute list Comprehension TLV. The default format as defined in ETSI TS 123 040 [10] is '00 00 03 90'.

5.5.4.1.1.2 Parameter errors

No requirements.

5.5.4.1.1.3 Context errors

No requirements.

5.5.4.1.2 Test area files

Test Source: Test_Cre_Pcs_Spco.java.

Test Applet: Cre_Pcs_Spco_1.java.

Cre_Pcs_Spco_2.java.

Cre_Pcs_Spco_3.java.

Cap File: Cre_pcs_spco.cap.

5.5.4.1.3 Test coverage

CRR number	Test case number
CRRN1	see: Api_2_Tkr_Cmet, CRRN1, Api_2_Tkr_Dmet, CRRN3, Api_2_Tkr_Emet, CRRN3, Api_2_Tkr_Imet, CRRN1 Api_2_Tkr_Smta, CRRN3
CRRN2	1
CRRN3	1
CRRN4	2, 3, 4
CRRN5	5, 6, 7
CRRN6	8
CRRN7	9, 10, 11
CRRN8	12
CRRN9	13
CRRN10	2, 4, 6, 7, 10, 11
CRRN11	See Api_2_Tkr_Cmet, CRRN6
CRRN12	See Api_2_Tkr_Cmet, CRRN6
CRRN13	14

5.5.4.1.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Set Up Menu at the beginning of a CAT session** Install Applet2 Applet2 registers to EVENT_UNRECOGNIZED_ENVELOPE and have its access conditions set to "ALWAYS" 1- Perform UICC initialization with set up menu facilities supported 2- Select EF_{SUME} under $DF_{TELECOM}$ with a select by path command 3- Update EF_{SUME} with the text "TEST MENU" and Text Attribute "00 00 13 90" (Bold) 4- An unrecognized envelope is sent to trigger Applet2 5- An unrecognized envelope is sent to trigger Applet2 6- The EF_{SUME} under $DF_{TELECOM}$ is updated with the text "UICC TEST" and restore the initial Text Attribute	4- Applet2 selects EF_{SUME} and updates its content with the text "TEST UICC" and no Text Attribute 5- Applet2 selects EF_{SUME} and updates its content with the text "TEST UICC" and Text Attribute "00 00 13 90"	1- SET UP MENU with main menu "UICC TEST" 2- SW = 9000 3- SET UP MENU with main menu "TEST MENU" and Text Attribute "00001390" 4- SET UP MENU with main menu "TEST UICC" and no Text Attribute 5- SET UP MENU with main menu "TEST UICC" and Text Attribute "00001390" 6- SET UP MENU with main menu "UICC TEST"
2	**Set Up Menu at the beginning of a CAT session** 1- Install Applet1 Applet1 registers to EVENT_MENU_SELECTION using the initMenuEntry(), to EVENT_STATUS_COMMAND using the requestPollIntervall() and to EVENT_EVENT_DOWNLOAD_MT_CALL, EVENT_EVENT_DOWNLOAD_LOCATION_STATUS		1- SET UP MENU with the menu of Applet1
3	**Set Up Menu with applet in LOCK state** 1- Lock Applet1		1- SET UP MENU with no menu of Applet1
4	**Set Up Menu with applet in SELECTABLE state** 1- Make selectable Applet1 2- An envelope menu selection is sent to trigger Applet1 3- Applet1 disables its menu 4- An envelope event download MT call is sent to trigger Applet1 5- Applet1 enables its menu	2- Applet1 is triggered 4- Applet1 is triggered	1- SET UP MENU with the menu 3- SET UP MENU with no menu 5- SET UP MENU with the menu
5	**Set Up Event List at the beginning of a CAT session** 1- Perform UICC initialization with EVENT DOWNLOAD and set up event list facilities supported		1- SET UP EVENT LIST proactive command [Event list]= '19020003' or '99020003'

Id	Description	API/CAT RE Expectation	APDU Expectation
6	**Setup Event List with applet in LOCK state** 1- Lock Applet1		1- SET UP EVENT LIST Proactive command [CommandQualifier]= 00h
7	**Setup Event List with applet in SELECTABLE state** 1- Make selectable Applet1		1- SET UP EVENT LIST proactive command [Event list]= '19020003' or '99020003'
8	**Dynamic setup event list on registry modification** 1- An envelope menu selection is sent to trigger Applet1 2- Applet1 deregisters to event EVENT_EVENT_DOWNLOAD_MT_CALL 3- An unrecognized envelope is sent to trigger Applet2 4- Applet registers to event EVENT_EVENT_DOWNLOAD_LOCATION_STATUS and to EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION 5- An envelope event download location status is sent to the UICC 6- Applet1 and Applet2 clear their events download 7- An envelope menu selection is sent to trigger Applet1 8- Applet1 registers to event EVENT_EVENT_DOWNLOAD_MT_CALL 9- Delete Applet1 10- Install Applet1 (same registration as before, plus registration to EVENT_UNRECOGNIZED_ENVELOPE)	1- Applet1 is triggered 3- Applet2 is triggered 5- Applet1 and Applet2 are triggered 7- Applet1 is triggered	2- SET UP EVENT LIST proactive command [Event list]= '190103' or '990103' 4- SET UP EVENT LIST proactive command [Event list]= '19020307' or '99020307' 6- SET UP EVENT LIST proactive command [CommandQualifier]= 00h 8- SET UP EVENT LIST proactive command [Event list]= '190100' or '990100' 9- SET UP EVENT LIST proactive command [CommandQualifier]= 00h 10- SET UP EVENT LIST proactive command [Event list]= '19020003' or '99020003'
9	**Poll Interval at the beginning of a CAT session** 1- Perform UICC initialization with polling facilities supported		1- POLL INTERVAL proactive command
10	**Poll Interval with applet in LOCK state** 1- Lock Applet1		1- POLLING OFF proactive command
11	**Poll Interval with applet in SELECTABLE state** 1- Make selectable Applet1		1- POLL INTERVAL proactive command

Id	Description	API/CAT RE Expectation	APDU Expectation
12	**Dynamic Polling commands on registry modification**		
	1- A status command is sent	1- Applet1 is triggered	
	2- Applet1 calls the method requestPollInterval() with POLL_NO_DURATION		2- POLLING OFF proactive command
	3- An unrecognized envelope is sent	3- Applet1 is triggered	
	4- Applet1 calls the method requestPollInterval() with POLL_SYSTEM_DURATION	Applet1 finalizes Applet2 is triggered	4- POLL INTERVAL proactive command
	5- Delete Applet1		5- POLLING OFF proactive command
	6- Install Applet1 (same registration as before, plus registration to EVENT_UNRECOGNIZED_ENVELOPE)		6- POLL INTERVAL proactive command
13	**System Proactive Commands sending**		
	1- Perform UICC initialization with system proactive commands facilities		
	2- An unrecognized envelope is sent	2- Applet1 is triggered	
	3- Applet1 deregisters to event EVENT_EVENT_DOWNLOAD_MT_CALL and UNRECOGNIZED_ENVELOPE, disables its menu entry, calls method requestPollIntervall() with POLL_NO_DURATION then builds and sends a Display Text Proactive command with text 'Text1'	Applet1 finalizes Applet2 is triggered	3- Display Text with text 'text1' proactive command
	4- Applet2 registers to event EVENT_PROACTIVE_HANDLER_AVAILABLE, disable its menu entry then builds and sends a Display Text Proactive command with text 'Text21'	Applet2 finalizes Applet2 is triggered by event EVENT_PROACTIVE_HANDLER_AVAILABLE	4- Display Text with text 'text21' proactive command
	5- Applet2 builds and sends a Display Text Proactive command with text 'Text22'	Applet2 finalizes	5- Display Text with text 'text22' proactive command
	7- Delete Applet1		6- SET UP MENU proactive command with no menu, SET UP EVENT LIST proactive command [Event list]= '190103' or '990103' and POLLING OFF proactive command

Id	Description	API/CAT RE Expectation	APDU Expectation
14	**Text Attribute management in Set Up Menu**		
	1- Install Applet3, Applet3 calls the initMenuEntry() method, then reinitialize the card.		1- SET UP MENU proactive command with one menu and no Item Text Attribute List or the default Text Attribute List '00000390”
	2- Send an Unrecognized Envelope to trigger Applet2	2- Applet2 is triggered	
	3- Applet2 enable its menu entry	Applet2 finalizes	3- SET UP MENU proactive command two menus and no Item Text Attribute List or the default Text Attribute List '0000039000000390”
	4-Send an envelope Menu Selection with the Item Id of Applet2	4- Applet2 is triggered	
	5- Applet2 calls setMenuEntryTextAttribute() method to set the attribute to “00 00 13 90” (Bold)	Applet2 finalizes	5- SET UP MENU proactive command two menus and the Item Text Attribute List “00001390 00000390”
	6-Send an envelope Menu Selection with the Item Id of Applet2	6- Applet2 is triggered	
	7- Applet2 calls disableMenuEntry()	Applet2 finalizes	7- SET UP MENU proactive command with one menu and no Item Text Attribute List or the default Text Attribute List '00000390”
	8-Send an envelope Menu Selection with the Item Id of Applet3	8- Applet3 is triggered	
	9- Applet3 calls setMenuEntryTextAttribute() method to set the attribute to “00 00 23 90” (Italic)	Applet3 finalizes	9- SET UP MENU proactive command one menu and the Item Text Attribute List “00002390”
	10-Send an envelope Menu Selection with the Item Id of Applet2	10- Applet2 is triggered	
	11- Applet2 calls enableMenuEntry()	Applet2 finalizes	11- SET UP MENU proactive command two menus and the Item Text Attribute List “00001390 00002390”
	12-Send an envelope Menu Selection with the Item Id of Applet2	12- Applet2 is triggered	
	13- Applet2 calls setMenuEntryTextAttribute() method to set the attribute to “00 00 03 90” (default)	Applet2 finalizes	13- SET UP MENU proactive command two menus and the Item Text Attribute List “00000390 00002390”
	14- Lock Applet3		14- SET UP MENU proactive command with one menu and no Item Text Attribute List or the default Text Attribute List '00000390”
	15- unlock Applet3		15- SET UP MENU proactive command two menus and the Item Text Attribute List “00000390 00002390”
	16- Send an envelope Menu Selection with the Item Id of Applet3	16- Applet3 is triggered	

Id	Description	API/CAT RE Expectation	APDU Expectation
	17- Applet3 calls setMenuEntryTextAttribute() method to set the attribute to "00 00 03 90" (default)	Applet3 finalizes	17- SET UP MENU proactive command two menus and no Item Text Attribute List or the default Text Attribute List '0000039000000390"

5.5.4.2 Interaction with GSM commands

Test Area Reference: Cre_Pcs_Igco

5.5.4.2.1 Conformance requirement

5.5.4.2.1.1 Normal execution

- CRRN1: The CAT Runtime Environment shall process a UICC command even when a proactive command is pending (before and after the FETCH command until the terminal response). The CAT Runtime Environment shall answer with the SW1 and SW2 described in ETSI TS 102 221 [5] and ETSI TS 102 223 [6].

5.5.4.2.1.2 Parameter errors

No requirements.

5.5.4.2.1.3 Context errors

No requirements.

5.5.4.2.2 Test area files

Test Source: Test_Cre_Pcs_Igco.java.

Test Applet: Cre_Pcs_Igco_1.java.

Cap File: Cre_pcs_igco.cap.

5.5.4.2.3 Test coverage

CRR number	Test case number
CRRN1	1, 2, 3

5.5.4.2.4 Test procedure

Id	Description	API/CAT RE 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- Failed SELECT File 4- FETCH 5- SELECT MF 6- TERMINAL RESPONSE		1- 91xx 2- Expected data = XX XX XX XX 3F 00 SW = 91XX 3- 6A82 4- Proactive Command: SETUP MENU 5- Expected data = XX XX XX XX 3F 00 SW = 9000 6- 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- Failed SELECT File 3- FETCH 4- SELECT MF 5- TERMINAL RESPONSE		1- Expected data = XX XX XX XX 3F 00 SW = 91XX 2- 6A82 3- Proactive Command: Display Text 4- Expected data = XX XX XX XX 3F 00 SW = 9000 5- 9000

Id	Description	API/CAT RE Expectation	APDU Expectation
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- FETCH 3- SELECT MF 4- Failed SELECT File 5- TERMINAL RESPONSE 6- SELECT MF 7-Failed SELECT File 8-FETCH 9-SELECT MF 10-TERMINAL RESPONSE		1- Expected data = XX XX XX XX 3F 00 SW = 91XX 2- Proactive Command: Display Text 3- Expected data = XX XX XX XX 3F 00 SW = 9000 4- 6A82 5- 91XX 6- Expected data = XX XX XX XX 3F 00 SW = 91XX 7-6A82 8-Proactive Command: Display Text 9- Expected data = XX XX XX XX 3F 00 SW = 9000 10- 9000

5.5.4.3 Proactive Command Control

Test Area Reference: Cre_Pcs_Pcco.

5.5.4.3.1 Conformance requirement

5.5.4.3.1.1 Normal execution

- CRRN1: The CAT Runtime Environment 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 CAT Runtime Environment shall throw an exception.
- CRRN2: The CAT Runtime Environment 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 CAT Runtime Environment shall throw an exception.
- CRRN3: The CAT Runtime Environment 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 CAT Runtime Environment shall throw an exception.
- CRRN4: The CAT Runtime Environment 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 CAT Runtime Environment shall throw an exception.
- CRRN5: The CAT Runtime Environment shall prevent a Toolkit Applet to issue a DECLARE SERVICE (add, delete) proactive command using a service identifier, which is not allocated to it. If an applet attempts to send such a command, the CAT Runtime Environment shall throw an exception.
- CRRN6: All proactive commands shall be sent to the terminal as constructed by the Toolkit Applet without any check by the CAT Runtime Environment.
- CRRN7: The CAT Runtime Environment 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.

5.5.4.3.1.2 Parameter errors

No requirements.

5.5.4.3.1.3 Context errors

No requirements.

5.5.4.3.2 Test area files

Test Source: Test_Cre_Pcs_Pcco.java.

Test Applet : Cre_Pcs_Pcco_1.java.

Cre_Pcs_Pcco_2.java.

Cre_Pcs_Pcco_3.java.

Cap File: Cre_pcs_pcco.cap.

5.5.4.3.3 Test coverage

CRR number	Test case number
CRRN1	1
CRRN2	2
CRRN3	3,4
CRRN4	3,4
CRRN5	5
CRRN6	6
CRRN7	Not testable

5.5.4.3.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
0	**Applets installation** Applet1 is installed with 4 timers maximum, 0 channel maximum, 1 menu and 4 service identifiers maximum Applet2 is installed with 8 timers maximum, 3 channels maximum, 1 menu and 8 service identifiers maximum Applet3 is installed with 1 channel maximum, 1 menu and no service identifier		
1	**STK Proactive Commands** 1- Send envelope menu selection with the item id 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 ToolkitException is thrown 3- COMMAND_NOT_ALLOWED ToolkitException is thrown 4- COMMAND_NOT_ALLOWED ToolkitException is thrown 5- COMMAND_NOT_ALLOWED ToolkitException is thrown	1- 90 00 (no proactive command is sent)

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**TIMER MANAGEMENT Proactive command** 1- Send envelope menu selection with the item id of Applet2 2- Applet2 allocates 8 timers by calling allocateTimer() method and release the 3 timers from id 1 to 3. 3- Send envelope menu selection with the item id of Applet1 4- Applet1 allocates 3 timers (Id 1 to 3) by calling allocateTimer() method 3 times 5- Send envelope menu selection with the item id of Applet2 6- Applet2 releases timers of Id 4 to 7 7- Send envelope menu selection with the item id 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 ToolkitException 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 envelope menu selection with the item id of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4- Applet1 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 ToolkitException is thrown 3- COMMAND_NOT_ALLOWED ToolkitException is thrown 4- COMMAND_NOT_ALLOWED ToolkitException is thrown 5- COMMAND_NOT_ALLOWED ToolkitException is thrown 6- COMMAND_NOT_ALLOWED ToolkitException is thrown	1- 90 00 (no proactive command is sent)

Id	Description	API/CAT RE Expectation	APDU Expectation
4	**4 Channels allowed**		
	1- Send envelope menu selection with the item id of Applet3	1- Applet3 is triggered	
	2- Applet3 builds and sends a CSD OPEN CHANNEL proactive command	2- No exception is thrown	2- 91 1C
	3- Send a Fetch and Terminal Response OK on channel 7		3- OPEN CHANNEL proactive
	4- Send envelope menu selection with the item id 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 ToolkitException is thrown	
	10- For each channel id from 3 to 7, Applet2 builds and sends a RECEIVE DATA proactive command	10- COMMAND_NOT_ALLOWED ToolkitException is thrown	
	11- For each channel id from 3 to 7, Applet2 builds and sends a CLOSE CHANNEL proactive command	11- COMMAND_NOT_ALLOWED ToolkitException 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 ToolkitException is thrown	14- 90 00 expected to the previous Terminal Response (no proactive command is sent)

Id	Description	API/CAT RE Expectation	APDU Expectation
5	**DECLARE SERVICE Proactive command** 1- Send envelope menu selection with the item id of Applet2 2- Applet2 allocates 8 services by calling allocateServiceIdentifier() method and release the 3 services from id 0 to 2 using method releaseServiceIdentifier(). 3- Send envelope menu selection with the item id of Applet1 4- Applet1 allocates 3 services (Id 0 to 2) by calling allocateServiceIdentifier() method 3 times 5- Send envelope menu selection with the item id of Applet2 6- Applet2 releases services of Id 5 to 7 7- Send envelope menu selection with the item id of Applet1 8- For each of the 3 services allocated by Applet1 (Id 0 to 2) DECLARE SERVICE (add) proactive commands are sent 9- For other services (Id 3 to 8), Applet1 builds and sends a DECLARE SERVICE (add) proactive command 10- For each of the 3 services allocated by Applet1 (Id 0 to 2) DECLARE SERVICE (delete) proactive commands are sent 11- For other services (Id 3 to 8), Applet1 builds and sends a DECLARE SERVICE (delete) 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 ToolkitException is thrown 10- No exception is thrown 11- COMMAND_NOT_ALLOWED ToolkitException is thrown	8- 3 DECLARE SERVICE proactive commands are fetched 9- The Status word of the last previous Terminal Response is 91 1C on the Terminal Response 10- 3 DECLARE SERVICE proactive commands are fetched 11- The Status word of the last previous Terminal Response is 90 00 (no more proactive command is sent)
6	**Unknown proactive command** 1- Send an envelope menu selection with the item id of Applet1 2- Applet1 builds an unknown proactive command 3- Fetch and terminal response OK	1- Applet1 is triggered	2- 91 08 3- The unknown proactive command is fetched

5.5.5 Exception Handling

5.5.5.1 General Behaviour

Test Area Reference: Cre_Exh_Genb.

5.5.5.1.1 Conformance requirement

5.5.5.1.1.1 Normal execution

- CRRN1: If more than one Applet shall be triggered by the currently processed event all Exceptions shall be caught by the CAT Runtime Environment and shall not be sent to the terminal. The CAT Runtime Environment shall proceed with the triggering.
- CRRN2: If only one Applet shall be triggered by the currently processed event and an ISOException with the reason code REPLY_BUSY is thrown, it shall be sent to the terminal using the Status Word 0x9300.
- CRRN3: If only one Applet shall be triggered by the currently processed event other Exceptions than an ISOException with the reason code REPLY_BUSY shall not be propagated to the terminal.

5.5.5.1.1.2 Parameter errors

No requirements.

5.5.5.1.1.3 Context errors

No requirements.

5.5.5.1.2 Test area files

Test Source: Test_Cre_Exh_Genb.java.

Test Applet: Cre_Exh_Genb_1.java.

Cre_Exh_Genb_2.java.

Cap File: Cre_exh_genb.cap.

5.5.5.1.3 Test coverage

CRR Number	Test Case Number
CRRN1	1
CRRN2	2
CRRN3	3

5.5.5.1.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
0	Applet1 is installed and registers to EVENT_MENU_SELECTION and EVENT_EVENT_DOWNLOAD_MT_CALL Applet2 is installed and registers to EVENT_EVENT_DOWNLOAD_MT_CALL and EVENT_EVENT_DOWNLOAD_USER_ACTIVITY		
1	**ISOException REPLY_BUSY is not sent to the terminal in multi triggering** 1- Send an envelope Event Download MT Call (multi triggering event, multi registered applets) 4- Send an envelope Event Download MT Call (multi triggering event, multi registered applets)	1- Applet1 is triggered 2- Applet1 sends a ISOException with the reason code REPLY_BUSY then finalizes Applet2 is triggered, does nothing and finalizes 4- Applet1 is triggered, does nothing and finalizes Applet2 is triggered, sends a ISOException with the reason code REPLY_BUSY then finalizes	3- SW = 90 00 5- SW = 90 00

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**ISOException REPLY_BUSY is sent to the terminal in single triggering**		
	1- Send an envelope Menu Selection to trigger Applet1 (single triggering event)	1- Applet1 is triggered, sends a ISOException with the reason code REPLY_BUSY then finalizes	1- SW = 93 00
	2- Send an envelope Event Download User Activity (multi triggering event, single registered applet)	2- Applet2 is triggered, sends a ISOException with the reason code REPLY_BUSY then finalizes	2- SW = 93 00
3	**Other exception than ISOException REPLY_BUSY are not sent to the terminal**		
	1- Send an envelope Menu Selection to trigger Applet1 (single triggering event)	1- Applet1 is triggered, sends a ISOException with reason code different to REPLY_BUSY then finalizes	1- SW = 90 00
	2- Send an envelope Menu Selection to trigger Applet1 (single triggering event)	2- Applet1 is triggered, sends a ToolkitException then finalizes	2- SW = 90 00

5.5.5.2 Interaction with Multiple Triggering

Test Area Reference: Cre_Exh_Imtg.

5.5.5.2.1 Conformance requirement

5.5.5.2.1.1 Normal execution

- CRRN1: An exception thrown by a toolkit applet, will not influence toolkit applets registered to the same event.

5.5.5.2.1.2 Parameter errors

No requirements.

5.5.5.2.1.3 Context errors

No requirements.

5.5.5.2.2 Test area files

Test Source: Test_Cre_Exh_Imtg.java.

Test Applet: Cre_Exh_Imtg_1.java.

Cre_Exh_Imtg_2.java.

Cap File: Cre_exh_imtg.cap.

5.5.5.2.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3, 4

5.5.5.2.4 Test procedure

Id	Description	API/CAT RE 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** Applet1: Priority= 0x01, Applet2: Priority= 0x02, (i.e. Applet1 is triggered before Applet2)		
1	**Profile_Download is sent**	1- Applet1 is triggered 2- NullPointerException is thrown 3- Applet2 is triggered	
2	Status_Command is sent	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	

5.5.6 Envelope Response Posting

5.5.6.1 General Behaviour

Test Area Reference: Cre_Erp_Genb.

5.5.6.1.1 Conformance requirement

5.5.6.1.1.1 Normal execution

- CRRN1: A Toolkit Applet can post a response to some events with the *post()* or the *postAsBERTLV()* methods and can continue its processing after the call to these methods.
- CRRN2: The CAT Runtime Environment shall send the response before the emission of the next proactive command or when all the Toolkit Applets triggered by the event have finished their processing.
- CRRN3: The Boolean parameter passed to the *post()* or *postAsBERTLV()* method shall be mapped by the CAT Runtime Environment to the correct status word. If the value is true it corresponds to a successful ending of the command status word "9000". If the value is false it corresponds to a warning status word "6200".

5.5.6.1.1.2 Parameter errors

No requirements.

5.5.6.1.1.3 Context errors

No requirements.

5.5.6.1.2 Test area files

None.

5.5.6.1.3 Test coverage

CRR Number	Test Case Number
CRRN1	See Api_2_Erh_Postb: CRRN1 See Api_2_Erh_Postbb: CRRN1
CRRN2	See Api_2_Erh_Postb: CRRN3 See Api_2_Erh_Postbb: CRRN3
CRRN3	See Api_2_Erh_Postb: CRRN4 See Api_2_Erh_Postbb: CRRN4

5.5.6.1.4 Test procedure

None.

5.5.6.2 EVENT_CALL_CONTROL_BY_NAA

Test Area Reference: Cre_Erp_Eccn.

5.5.6.2.1 Conformance requirement

5.5.6.2.1.1 Normal execution

- CRRN1: The CAT Runtime Environment can't reply busy when an Envelope(Call Control) is sent to the UICC.

5.5.6.2.1.2 Parameter errors

No requirements.

5.5.6.2.1.3 Context errors

No requirements.

5.5.6.2.2 Test area files

Test Source: Test_Cre_Erp_Eccn.java.

Test Applet: Cre_Erp_Eccn_1.java.

Cre_Erp_Eccn_2.java.

Cre_Erp_Eccn_3.java.

Cap File: Cre_erp_eccn.cap.

5.5.6.2.3 Test coverage

CRR Number	Test Case Number
CRRN1	1, 2

5.5.6.2.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applet1 is registered on the EVENT_CALL_CONTROL_BY_NAA, 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 UICC 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 UICC 5- A Terminal Response command is sent to the UICC 6- Delete Applet1 & Applet2 7- Install Applet3	1- Applet2 is suspended 2- Applet1 is triggered 5- Applet2's execution shall continue	3- The dialling number is retrieved and the status words is 91xx
2	**Applet3 is registered on both the events EVENT_CALL_CONTROL_BY_NAA and EVENT_MENU_SELECTION** 1- Envelope Menu Selection is sent to the UICC 2- Applet3 invokes the method send()and no fetch is performed 3- Envelope(Call Control) is sent to the UICC 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 UICC 6- A Terminal Response command is sent to the UICC	1- Applet3 is triggered on the EVENT_MENU_SELECTION 2- Applet3 is suspended on the send() method 3- Applet3 is triggered on the EVENT_CALL_CONTROL_BY_NAA 6- The Applet3's execution shall continue	4- The dialling number is retrieved and the status words is 91xx

5.5.6.3 EVENT_UNRECOGNIZED_ENVELOPE

Test Area Reference: Cre_Erp_Euen.

5.5.6.3.1 Conformance requirement

5.5.6.3.1.1 Normal execution

- CRRN1: The EnvelopeResponseHandler is available for the EVENT_UNRECOGNIZED_ENVELOPE.

5.5.6.3.1.2 Parameter errors

No requirements.

5.5.6.3.1.3 Context errors

No requirements.

5.5.6.3.2 Test area files

Test Source: Test_Cre_Erp_Euen.java.

Test Applet: Cre_Erp_Euen_1.java.

Cap File: Cre_erp_euen.cap.

5.5.6.3.3 Test coverage

CRR Number	Test Case Number
CRRN1	1

5.5.6.3.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**An applet triggered on the EVENT_UNRECOGNIZED_ENVELOPE calls the EnvelopeResponseHandler.post() method**	The post() method returns no exception	The UICC answers to the Envelope with status words 9000. The data retrieved are the ones posted by the applet.

5.5.7 Toolkit Installation

5.5.7.1 General Behaviour

Test Area Reference: Cre_Tin_Genb

5.5.7.1.1 Conformance requirement

5.5.7.1.1.1 Normal execution

- CRRN1: The UICC Toolkit Application specific parameters (Tag 80h) are mandatory for applications using the *uicc.toolkit.ToolkitInterface* defined in ETSI TS 102 241 [9].
- CRRN2: Any additional parameters of the UICC Toolkit Application specific parameters field (Tag 80h) shall be ignored by the card.
- CRRN3: Some unused byte may be added at the end of the UICC Toolkit Application specific parameters field (Tag 80h).
- CRRN4: The UICC Access Application specific parameters (Tag 81h) are applicable to applications using the *uicc.access.FileView* defined in ETSI TS 102 241 [9].
- CRRN5: The UICC Toolkit Application specific parameters field (Tag 80h) is not required for applications that do not use the *uicc.toolkit.ToolkitInterface* defined in ETSI TS 102 241 [9].
- CRRN6: The UICC Access Application specific parameters field (Tag 81h) is not required for applications that a do not use the *uicc.access.FileView* defined in ETSI TS 102 241 [9].

5.5.7.1.1.2 Parameter errors

No requirements.

5.5.7.1.1.3 Context errors

No requirements.

5.5.7.1.2 Test area files

Test Source: Test_Cre_Tin_Genb.java.

Test Applet: Cre_Tin_Genb_1.java (use *uicc.toolkit.ToolkitInterface*).

Cre_Tin_Genb_2.java (use *uicc.access.FileView*).

Cre_Tin_Genb_3.java (use *uicc.toolkit.ToolkitInterface* and *uicc.access.FileView*).

Cap File: Cre_tin_genb.cap.

5.5.7.1.3 Test coverage

CRR number	Test case number
CRRN1	1, 2
CRRN2	2
CRRN3	2
CRRN4	3
CRRN5	3
CRRN6	1, 2

5.5.7.1.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Good installation with the only tag 80h** 1- Install(install) Applet1 with only the UICC Toolkit Application specific parameters field		1- RAPDU = 00 90 00
2	**Good installation with the only tag 80h** 1- Install(install) Applet1 with only the UICC Toolkit Application specific parameters field which contains some unused bytes		1- RAPDU = 00 90 00
3	**Good installation with the only tag 81h** 1- Install(install) Applet2 with only the UICC Access Application specific parameters field		1- RAPDU = 00 90 00

5.5.7.2 Timers Allocation

Test Area Reference: Cre_Tin_Tmal.

5.5.7.2.1 Conformance requirement

5.5.7.2.1.1 Normal execution

- CRRN1: One toolkit applet can register to several timers, but a timer can only be allocated to one toolkit applet.

5.5.7.2.1.2 Parameter errors

No requirements.

5.5.7.2.1.3 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 ETSI TS 102 223 [6], the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

5.5.7.2.2 Test area files

Test Source: Test_Cre_Tin_Tmal.java.

Test Applet: Cre_Tin_Tmal_1.java.

Cre_Tin_Tmal_2.java.

Cre_Tin_Tmal_3.java.

Cap File: Cre_tin_tmal.cap.

5.5.7.2.3 Test coverage

CRR number	Test case number
CRRN1	2, 3, 8
CRRC1	1, 7
CRRC2	4, 5, 6

5.5.7.2.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**More than 8 timers at the instantiation of Applet1: check that Applet1 is not installed.** 1- Install Applet1 with maximum 9 timers allocated 2- Applet1 is selected		1- The installation failed with the status word 6A80 2- Applet1 is not found, RAPDU != <applet selected data> 90 00
	Reset the card		
2	**Good installation of Applet2** 1- Install Applet2 with maximum 4 timers allocated		
3	**Allocate 4 timers Applet2** 1- An envelope menu selection is send to trigger Applet2 2- Applet2 allocates 4 timers	2- No exception shall be thrown	
4	**Allocate one more timer Applet2** 1- Applet2 allocates one more timer	1- Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE	
5	**Good installation of applet3** 1- Install Applet3 with maximum 8 timers allocated		

Id	Description	API/CAT RE Expectation	APDU Expectation
6	**Allocate 4 timers** **Applet3** 1- an envelope menu selection is send to trigger Applet3 2- Applet3 allocates 4 timers	2-No exception shall be thrown	
7	**Allocate one more timer** **Applet3** 1- Applet3 allocates one more timer	1- 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**		

5.5.7.3 Item Identifier

Test Area Reference: Cre_Tin_Itid

5.5.7.3.1 Conformance requirement

5.5.7.3.1.1 Normal execution

- CRRN1: If the requested item identifier in the range [1 to 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].

5.5.7.3.1.2 Parameter errors

- CRRP1: If the requested item identifier is in the range [128,255], then the card shall reject the install command.

5.5.7.3.1.3 Context errors

- CRRC1: If the requested item identifier in the range [1 to 127] is already allocated, then the card shall reject the install command.

5.5.7.3.2 Test area files

Test Source: Test_Cre_Tin_Itid.java.

Test Applet: Cre_Tin_Itid_1.java.

Cre_Tin_Itid_2.java.

Cre_Tin_Itid_3.java.

Cap File: Cre_tin_itid.cap.

5.5.7.3.3 Test coverage

CRR number	Test case number
CRRN1	2
CRRN2	4, 5, 6
CRRP1	1
CRRC1	3

5.5.7.3.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Bad installation of Applet1** 1- Installation of Applet1 The following parameters item Id equal to 128 2- Applet1 is selected		1- The installation failed with the status word 6A80 2- Applet1 is not found, RAPDU != <applet selected data> 90 00
2	**Good installation of Applet1** 1- Installation of Applet1 Item Id = 1 for the first menu and 127 for the second one 2- A Terminal Profile is sent to the card with only PROFILE_DOWNLOAD, MENU_SELECTION, SET_UP_MENU and COMMAND_RESULT facilities.		2- The UICC answers 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** 1- Installation of Applet2 item Id = 127 2- Applet2 is selected		1- The installation failed with the status word 6A80 2- Applet2 is not found, RAPDU != <applet selected data> 90 00
4	**Good installation of Applet2** 1- Installation of Applet2 item Id = 0		1- The UICC answers 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** 1- Installation of Applet3 item Id = 0		1- The UICC answers 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/CAT RE Expectation	APDU Expectation
6	**Good deletion and installation of Applet2** 1- Delete instance of Applet2 2- Install for install of Applet2 item Id = 0		1- The UICC answers 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 3- The UICC answers 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

5.5.7.4 Item Position

Test Area Reference: Cre_Tin_Itpo.

5.5.7.4.1 Conformance requirement

5.5.7.4.1.1 Normal execution

- CRRN1: If the new Menu Entry has to be inserted at an already occupied position, the entries from the requested position to the last element of the Menu Entries' list are shifted to the next positions.
- CRRN2: If the position indicated is greater than the number of elements in the Menu Entries' list, then the Menu Entry takes the last position in the Menu Entries' list.
- CRRN3: If the position indicated is equal to '00', then the Menu Entry takes the last position in the Menu Entries' list.

5.5.7.4.1.2 Parameter errors

No requirements.

5.5.7.4.1.3 Context errors

No requirements.

5.5.7.4.2 Test area files

Test Source: Test_Cre_Tin_Itpo.java.

Test Applet: Cre_Tin_Itpo_1.java.

Cre_Tin_Itpo_2.java.

Cre_Tin_Itpo_3.java.

Cre_Tin_Itpo_4.java.

Cre_Tin_Itpo_5.java.

Cre_Tin_Itpo_6.java.

Cap File: Cre_tin_itpo.cap.

5.5.7.4.3 Test coverage

CRR number	Test case number
CRRN1	1 to 10
CRRN2	5
CRRN3	4

5.5.7.4.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Installation of Applet1** 1- Install Applet1 Position/ItemId 01/01 02/02 03/03 04/04		1- The UICC answers to the Envelope with status words 91xx to send back to the ME the 4 menus The menus are (position(1)/itemId/text) 01/01/menu11 02/02/menu12 03/03/menu13 04/04/menu14 (1) position is the position in the set up menu proactive command
2	**Installation of Applet2** 1- Install Applet2 Position/ItemId 03/05		1- The UICC 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 03/05/menu21 04/03/menu13 05/04/menu14
3	**Installation of Applet3** 1- Install Applet3 Position/ItemId 02/06 03/07		1- The UICC answers to the Envelope with status words 91xx to send back to the ME the 7 menus The menus are (position/itemId/text) 01/01/menu11 02/06/menu31 03/07/menu32 04/02/menu12 06/05/menu21 07/03/menu13 08/04/menu14

Id	Description	API/CAT RE Expectation	APDU Expectation
4	**Installation of Applet4** 1- Install Applet4 Position/ItemId 00/08		1- The UICC answers to the Envelope with status words 91xx to send back to the ME the 8 menus The menus are (position/itemId/text) 01/01/menu11 02/06/menu31 03/07/menu32 04/02/menu12 06/05/menu21 07/03/menu13 08/04/menu14 09/08/menu41
5	**Installation of Applet5** 1- Install Applet5 Position/ItemId 20/09		1- The UICC answers to the Envelope with status words 91xx to send back to the ME the 9 menus The menus are (position/itemId/text) 01/01/menu11 02/06/menu31 03/07/menu32 04/02/menu12 06/05/menu21 07/03/menu13 08/04/menu14 09/08/menu41 10/09/menu51
6	**Disabling of the first menu of Applet1 and locking of Applet2** 1- An envelope menu selection is sent with Item Id = 02 2- Applet1 disables its first menu (Item Id = 01) 3- Lock the Applet2	1- Applet1 is triggered	2- The UICC answers to the Envelope with status words 91xx to send back to the ME the 08 menus The menus are (position/itemId/text) 01/06/menu31 02/07/menu32 03/02/menu12 05/05/menu21 06/03/menu13 07/04/menu14 08/08/menu41 09/09/menu51 3- The UICC answers to the Envelope with status words 91xx to send back to the ME the 07 menus The menus are (position/itemId/text) 01/06/menu31 02/07/menu32

Id	Description	API/CAT RE Expectation	APDU Expectation
			03/02/menu12 05/03/menu13 06/04/menu14 07/08/menu41 08/09/menu51
7	**Installation of Applet6** 1- Install Applet6 Position/ItemId 01/10 04/11 15/12		1- The UICC answers to the Envelope with status words 91xx to send back to the ME the 10 menus The menus are (position/itemId/text) 01/10/menu61 02/06/menu31 03/11/menu62 04/07/menu32 05/02/menu12 07/03/menu13 08/04/menu14 09/08/menu41 10/09/menu51 11/12/menu63
8	**Enabling of the first menu of Applet1 and unlocking of Applet2** 1- An envelope menu selection is sent with Item Id = 02 2- Applet1 enables its first menu (Item Id = 01) 3- Unlock the Applet2	1- Applet1 is triggered	2- The UICC answers to the Envelope with status words 91xx to send back to the ME the 11 menus The menus are (position/itemId/text) 01/10/menu61 02/01/menu11 03/06/menu31 04/11/menu62 05/07/menu32 06/02/menu12 08/03/menu13 09/04/menu14 10/08/menu41 11/09/menu51 12/12/menu63 3- The UICC answers to the Envelope with status words 91xx to send back to the ME the 12 menus The menus are (position/itemId/text) 01/10/menu61 02/01/menu11 03/06/menu31 04/11/menu62 05/07/menu32 06/02/menu12 08/05/menu21 09/03/menu13 10/04/menu14 11/08/menu41 12/09/menu51 13/12/menu63

Id	Description	API/CAT RE Expectation	APDU Expectation
09	**Deletion of Applet2** 1- Delete Applet2		1- The UICC answers to the Envelope with status words 91xx to send back to the ME the 11 menus The menus are (position/itemId/text) 01/10/menu61 02/01/menu11 03/08/menu31 04/11/menu62 05/07/menu32 06/02/menu12 08/03/menu13 09/04/menu14 10/08/menu61 11/09/menu51 12/12/menu63
10	**Installation of Applet2** 1- Install Applet2 Position/ItemId 03/05		1- The UICC answers to the Envelope with status words 91xx to send back to the ME the 12 menus The menus are (position/itemId/text) 01/10/menu61 02/01/menu11 03/05/menu21 04/06/menu31 05/11/menu62 06/07/menu32 07/02/menu12 09/03/menu13 10/04/menu14 11/08/menu41 12/09/menu51 13/12/menu63

5.5.7.5 Maximum Text Length for a menu entry

Test Area Reference: Cre_Tin_Mlme.

5.5.7.5.1 Conformance requirement

5.5.7.5.1.1 Normal execution

- CRRN1: The maximum length of item text string is defined at the installation of the toolkit applet.

5.5.7.5.1.2 Parameter 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.

5.5.7.5.1.3 Context errors

No requirements.

5.5.7.5.2 Test area files

Test Source: Test_Cre_Tin_Mlme.java.

Test Applet: Cre_Tin_Mlme_1.java.

Cap File: Cre_tin_mlme.cap.

5.5.7.5.3 Test coverage

CRR number	Test case number
CRRN1	1, 3, 4
CRRP1	2
CRRP2	5

5.5.7.5.4 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 3 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 (3 items) is issued with TLV item length equal to 11 (Identifier + Text string of item)
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 (3 items) is issued with TLV item length equal to 11 (Identifier + Text string of item)

Id	Description	API / Framework Expectation	APDU Expectation
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	SW = 90 00

5.5.7.6 Maximum number of menu entries

Test Area Reference: Cre_Tin_Nbme.

5.5.7.6.1 Conformance requirement

5.5.7.6.1.1 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.

5.5.7.6.1.2 Parameter errors

- CRRP1: If the menu entry cannot be initialized (e.g. no more item data in applet loading parameter), a ToolkitException with the REGISTRY_ERROR reason code is thrown.

5.5.7.6.1.3 Context errors

No requirements.

5.5.7.6.2 Test area files

Test Source: Test_Cre_Tin_Nbme.java.

Test Applet: Cre_Tin_Nbme_1.java.

Cre_Tin_Nbme_2.java.

Cap File: Cre_tin_nbme.cap.

5.5.7.6.3 Test coverage

CRR number	Test case number
CRRN1	1
CRRP1	2, 3

5.5.7.6.4 Test procedure

Id	Description	API/CAT RE 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: “menu1”, “menu2”, “menu3”)
3	**Installation of 2nd applet with 0 menu** 1- Install (install) another applet, with max. number of menu entry is '0', defined at the install (install) command. initMenuEntry once MenuEntry = "menu5" Offset = 0 Length = 5 NextAction = '00' HelpSupported = false IconQualifier = '00' IconIdentifier = 0 2- Perform a RESET and a Terminal Profile with the facilities of PROFILE_DOWNLOAD, MENU_SELECTION, COMMAND_RESULT and SET_UP_MENU	ToolkitException REGISTRY_ERROR is thrown	2- SET UP MENU (3 items: “menu1”, “menu2”, “menu3”)

5.5.7.7 Access Domain

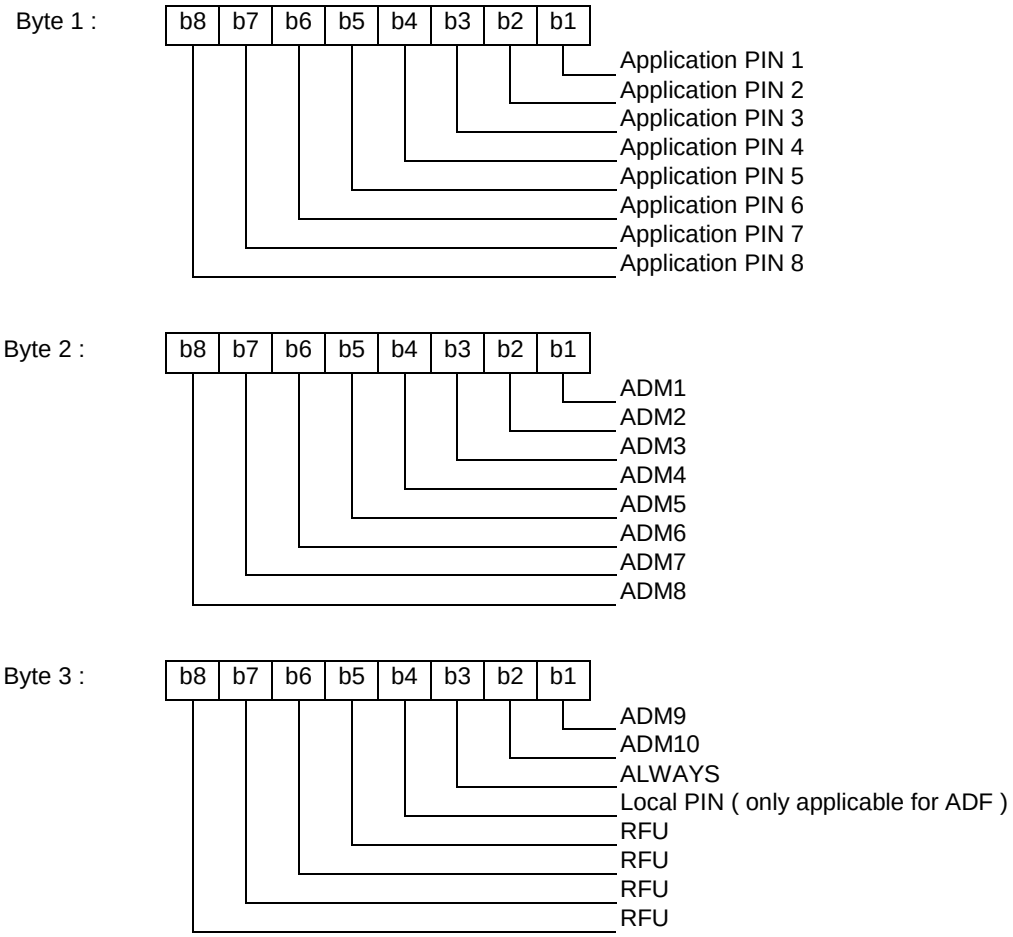
Test Area Reference: Cre_Tin_Acdo.

5.5.7.7.1 Conformance requirement

5.5.7.7.1.1 Normal execution

- CRRN1: The UICC access specific parameters (Tag 80h) indicate the mechanism used to control the application instance access to the File System ('00' means full access to the File System, '02' means UICC access mechanism and 'FF' means no access to the File System).
- CRRN2: The UICC access specific parameters are applicable to applications using the *uicc.access.FileView* defined in ETSI TS 102 241 [9].
- CRRN3: The UICC administrative access parameter (Tag 83h) indicate the mechanism used to control the application instance access to the File System ('00' means full access to the File System, '02' means UICC access mechanism and 'FF' means no access to the File System).
- CRRN4: The UICC administrative access parameters are applicable to applications using the *uicc.access.fileadministration.AdminFileView* defined in ETSI TS 102 241 [9].

- CRRN5: If an application has Access Domain Parameter '00' (i.e. Full Access to the File System), all actions can be performed on a file except the ones with NEVER access condition.
- CRRN6: If an application has Access Domain Parameter '02' (i.e. UICC access mechanism). The UICC access mechanism shall be coded as follows:



These access rights shall be checked against SE ID 01 access rules as defined in ETSI TS 102 221 [5].

5.5.7.7.1.2 Parameter 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 application with Access Domain Parameter 'FF' (i.e. No Access to the File System) tries to access a file the CAT Runtime Environement shall throw a UICCException with a SECURITY_STATUS_NOT_SATISFIED reason.
- CRRP3: If an application with Access Domain Parameter '02' (i.e. UICC access mechanism) tries to access a file without the correct rights, the CAT Runtime Environement shall throw a UICCException with a SECURITY_STATUS_NOT_SATISFIED reason.

5.5.7.7.1.3 Context errors

No requirements.

5.5.7.7.2 Test area files

Test Source: Test_Cre_Tin_Acdo.java.

Test Applet: Cre_Tin_Acdo_1.java.

Cre_Tin_Acdo_2.java.

Cre_Tin_Acdo_3.java.

Cre_Tin_Acdo_4.java.

Cre_Tin_Acdo_5.java.

Cre_Tin_Acdo_6.java.

Cre_Tin_Acdo_7.java.

Cre_Tin_Acdo_8.java.

Cap File: Cre_tin_acdo.cap.

5.5.7.7.3 Test coverage

CRR number	Test case number
CRRN1	1 to 6
CRRN2	1 to 6
CRRN3	1 to 6
CRRN4	1 to 6
CRRN5	1
CRRN6	3 to 6
CRRP1	Not tested
CRRP2	2
CRPP3	3 to 6

5.5.7.7.4 Test procedure

The following table summarizes tests performed in the test procedure.

		Cyclic files (EF_{CARRx})								
		Linear fixed files(EF_{LARRx})								
		Transparent files(EF_{TARRx})								
Applet rights	**File rights**	**Activate**	**Deactivate**	**Read Bin/Rec**	**Update Bin/Rec**	**Search**	**Increase**	**Create**	**Delete**	**Resize**
Full access	Always (EF_{xARR1})	OK	OK	OK	OK	OK	OK	OK	OK	OK
	Never (EF_{xARR2})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Global PIN1 & ADM1 (EF_{xARR3})	OK	OK	OK	OK	OK	OK	OK	OK	OK
	Global PIN1 | ADM1 (EF_{xARR4})	OK	OK	OK	OK	OK	OK	OK	OK	OK
	Local PIN 1 / ADM2 (See note) (EF_{xARR5})	OK	OK	OK	OK	OK	OK	OK	OK	OK

		Cyclic files (EF_{CARRx})								
		Linear fixed files(EF_{LARRx})								
		Transparent files(EF_{TARRx})								
Applet rights	**File rights**	**Activate**	**Deactivate**	**Read Bin/Rec**	**Update Bin/Rec**	**Search**	**Increase**	**Create**	**Delete**	**Resize**
No Access	Always (EF_{xARR1})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Never (EF_{xARR2})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Global PIN1 & ADM1 (EF_{xARR3})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Global PIN1 | ADM1 (EF_{xARR4})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Local PIN 1 / ADM2 (See note) (EF_{xARR5})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
Always	Always (EF_{xARR1})	OK	OK	OK	OK	OK	OK	OK	OK	OK
	Never (EF_{xARR2})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Global PIN1 & ADM1 (EF_{xARR3})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Global PIN1 | ADM1 (EF_{xARR4})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Local PIN 1 / ADM2 (See note) (EF_{xARR5})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
Global PIN1	Always (EF_{xARR1})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Never (EF_{xARR2})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Global PIN1 & ADM1 (EF_{xARR3})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Global PIN1 | ADM1 (EF_{xARR4})	OK	OK	OK	OK	OK	OK	OK	OK	OK
	Local PIN 1 / ADM2 (See note) (EF_{xARR5})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
Global PIN1 & ADM1	Always (EF_{xARR1})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Never (EF_{xARR2})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Global PIN1 & ADM1 (EF_{xARR3})	OK	OK	OK	OK	OK	OK	OK	OK	OK
	Global PIN1 | ADM1 (EF_{xARR4})	OK	OK	OK	OK	OK	OK	OK	OK	OK
	Local PIN 1 / ADM2 (See note) (EF_{xARR5})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
Local PIN & ADM2	Always (EF_{xARR1})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Never (EF_{xARR2})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK

		Cyclic files (EF_{CARRx})								
		Linear fixed files(EF_{LARRx})								
		Transparent files(EF_{TARRx})								
Applet rights	**File rights**	**Activate**	**Deactivate**	**Read Bin/Rec**	**Update Bin/Rec**	**Search**	**Increase**	**Create**	**Delete**	**Resize**
	Global PIN1 & ADM1 (EF_{xARR3})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Global PIN1 | ADM1 (EF_{xARR4})	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK	NOK
	Local PIN 1 / ADM2 (See note) (EF_{xARR5})	OK	OK	OK	OK	OK	OK	OK	OK	OK
NOTE:	For EF_{XARR5}, the file access condition is Local PIN 1 if the file is located under the MF and ADM2 if it is located under ADF1.									

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Full access Applet** 0- Applet1 installation with full access right 1- Send an envelope Menu Selection to trigger Applet1 on menu Id 1	1- Applet1 is triggered and gets Fileviews on UICC and ADF1 1.1- For each EF_{CARRx} Applet1 calls all associated methods with success, except on file EF_{CARR2} where UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown. 1.2- For each EF_{LARRx} Applet1 calls all associated methods with success, except on file EF_{LARR2} where UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown. 1.3- For each EF_{TARRx} Applet1 calls all associated methods with success, except on file EF_{TARR2} where UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown.	1- SW = 90 00
	2- Send an envelope Menu Selection to trigger Applet1 on menu Id 2	2- Applet1 is triggered and gets AdminFileviews on UICC and ADF1 2.1- For each EF_{CARRx} Applet1 calls all associated methods with success, except on file EF_{CARR2} where UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown. 2.2- For each EF_{LARRx} Applet1 calls all associated methods with success, except on file EF_{LARR2} where UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown. 2.3- For each EF_{TARRx} Applet1 calls all associated methods with success, except on file EF_{TARR2} where UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown. 2.4- Under each DF_{ARRx} Applet1 resize the EF_{TARxT} with success, except on file EF_{TAR2T} where an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown. 2.5- Under each DF_{ARRx} Applet1 delete the EF_{TARxT} with success, except on file EF_{TAR2T} where an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown. 2.6- Under each DF_{ARRx} Applet1 create the EF_{TARxT} with success, except under DF_{ARR2} where an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown at creation of a file similar to EF_{TAR2T}.	2- SW = 90 00
	3- Applet1 deletion		

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**No access Applet** 0- Applet2 installation with no access right 1- Send an envelope Menu Selection to trigger Applet2 on menu Id 1	1- Applet2 is triggered and gets Fileviews on UICC and ADF1 1.1- For each EF_{CARRx} Applet2 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown. 1.2- For each EF_{LARRx} Applet2 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown. 1.3- For each EF_{TARRx} Applet2 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown.	1- SW = 90 00
	2- Send an envelope Menu Selection to trigger Applet2 on menu Id 2	2- Applet2 is triggered and gets AdminFileviews on UICC and ADF1 2.1- For each EF_{CARRx} Applet2 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown. 2.2- For each EF_{LARRx} Applet2 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown. 2.3- For each EF_{TARRx} Applet2 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown. 2.4- Under each DF_{ARRx} Applet2 resize the $EF_{TARxT;}$ UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown. 2.5- Under each DF_{ARRx} Applet2 delete the $EF_{TARxT;}$ UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown.	2- SW = 90 00
	3- Applet2 deletion	2.6- Under each DF_{ARRx} Applet2 create an EF; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown.	

Id	Description	API/CAT RE Expectation	APDU Expectation
3	**Always access right Applet** 0- Applet3 installation with Always access right 1- Send an envelope Menu Selection to trigger Applet3 on menu Id 1	1- Applet3 is triggered and gets Fileviews on UICC and ADF1 1.1- For each EF_{CARRx} Applet3 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{CARR1} where the execution is successful. 1.2- For each EF_{LARRx} Applet3 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{LARR1} where the execution is successful. 1.3- For each EF_{TARRx} Applet3 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{TARR1} where the execution is sucessful.	1- SW = 90 00
	2- Send an envelope Menu Selection to trigger Applet3 on menu Id 2	2- Applet3 is triggered and gets AdminFileviews on UICC and ADF1 2.1- For each EF_{CARRx} Applet3 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{CARR1} where the execution is successful. 2.2- For each EF_{LARRx} Applet3 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{LARR1} where the execution is successful. 2.3- For each EF_{TARRx} Applet3 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{TARR1} where the execution is sucessful. 2.4- Under each DF_{ARRx} Applet3 resize the EF_{TARxT}; an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown, except on file EF_{TAR1T} where the execution is successful. 2.5- Under each DF_{ARRx} Applet3 delete the EF_{TARxT}; an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown, except on file EF_{TAR1T} where the execution is successful. 2.6- Under each DF_{ARRx} Applet3 create an EF like the EF_{TARxT}; an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown, except on file EF_{TAR1T} where the execution is successful.	2- SW = 90 00
	3- Applet3 deletion		

Id	Description	API/CAT RE Expectation	APDU Expectation
4	**Global PIN1 access right Applet** 0- Applet4 installation with Global PIN1 access right		
	1- Send an envelope Menu Selection to trigger Applet4 on menu Id 1	1- Applet4 is triggered and gets Fileviews on UICC and ADF1 1.1- For each EF_{CARRx} Applet4 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{CARR4} where the execution is successful. 1.2- For each EF_{LARRx} Applet4 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{LARR4} where the execution is successful. 1.3- For each EF_{TARRx} Applet4 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{TARR4} where the execution is sucessful.	1- SW = 90 00
	2- Send an envelope Menu Selection to trigger Applet4 on menu Id 2	2- Applet4 is triggered and gets AdminFileviews on UICC and ADF1 2.1- For each EF_{CARRx} Applet4 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{CARR4} where the execution is successful. 2.2- For each EF_{LARRx} Applet4 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{LARR4} where the execution is successful. 2.3- For each EF_{TARRx} Applet4 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{TARR4} where the execution is sucessful. 2.4- Under each DF_{ARRx} Applet4 resize the EF_{TARxT}; an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown, except on file EF_{TAR4T} where the execution is successful. 2.5- Under each DF_{ARRx} Applet4 delete the EF_{TARxT}; an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown, except on file EF_{TAR4T} where the execution is successful. 2.6- Under each DF_{ARRx} Applet4 create an EF like the EF_{TARxT}; an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown, except on file EF_{TAR4T} where the execution is successful.	2- SW = 90 00
	3- Applet4 deletion		

Id	Description	API/CAT RE Expectation	APDU Expectation
5	**Global PIN1 & ADM1 access right Applet**		
	0- Applet5 installation with Global PIN1 & ADM1 access right		
	1- Send an envelope Menu Selection to trigger Applet5 on menu Id 1	1- Applet5 is triggered and gets Fileviews on UICC and ADF1 1.1- For each EFCARRx Applet5 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on files EFCARR3 and EFCARR4 where the execution is successful. 1.2- For each EFLARRx Applet5 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on files EFCARR3 and EFCARR4 where the execution is successful. 1.3- For each EFTARRx Applet5 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on files EFCARR3 and EFCARR4 where the execution is sucessful.	1- SW = 90 00
	2- Send an envelope Menu Selection to trigger Applet5 on menu Id 2	2- Applet3 is triggered and gets AdminFileviews on UICC and ADF1 2.1- For each EFCARRx Applet5 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on files EF_{CARR3} and EF_{CARR4} where the execution is successful. 2.2- For each EFLARRx Applet5 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on files EFCARR3 and EFCARR4 where the execution is successful. 2.3- For each EFTARRx Applet5 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on files EFCARR3 and EFCARR4 where the execution is sucessful. 2.4- Under each DFARRx Applet5 resize the EFTARxT; an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown, except on files EFCARR3 and EFCARR4 where the execution is successful. 2.5- Under each DFARRx Applet5 delete the EF_{TARxT}; an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown, except on files EF_{CARR3} and EF_{CARR4} where the execution is successful. 2.6- Under each DFARRx Applet5 create an EF like the EF_{TARxT}; an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown, except on files EF_{CARR3} and EF_{CARR4} where the execution is successful.	2- SW = 90 00
	3- Applet5 deletion		

Id	Description	API/CAT RE Expectation	APDU Expectation
6	**Local Pin & ADM2 access right Applet** 0- Applet6 installation with Local PIN & ADM2 access right 1- Send an envelope Menu Selection to trigger Applet6 on menu Id 1	1- Applet6 is triggered and gets Fileviews on UICC and ADF1 1.1- For each EF_{CARRx} Applet6 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{CARR5} where the execution is successful. 1.2- For each EF_{LARRx} Applet6 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{LARR5} where the execution is successful. 1.3- For each EF_{TARRx} Applet6 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{TARR5} where the execution is sucessful.	1- SW = 90 00
	2- Send an envelope Menu Selection to trigger Applet6 on menu Id 2	2- Applet6 is triggered and gets AdminFileviews on UICC and ADF1 2.1- For each EF_{CARRx} Applet6 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{CARR5} where the execution is successful. 2.2- For each EF_{LARRx} Applet6 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{LARR5} where the execution is successful. 2.3- For each EF_{TARRx} Applet6 calls all associated methods; UICCException.SECURITY_STATUS_NOT_SATISFIED are thrown, except on file EF_{TARR5} where the execution is sucessful. 2.4- Under each DF_{ARRx} Applet6 resize the EF_{TARxT}; an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown, except on file EF_{TAR5T} where the execution is successful. 2.5- Under each DF_{ARRx} Applet6 delete the EF_{TARxT}; an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown, except on file EF_{TAR5T} where the execution is successful. 2.6- Under each DF_{ARRx} Applet6 create an EF like the EF_{TARxT}; an UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown, except on file EF_{TAR5T} where the execution is successful.	2- SW = 90 00
	3- Applet6 deletion		

Id	Description	API/CAT RE Expectation	APDU Expectation
7	**AdminFileview and Fileview acces domain parameter differenciation**		
	0- Applet7 installation with Always & ADM1 access right for Fileview access domain		
	1- Send an envelope Menu Selection on menu id 1 to trigger Applet7	1- Applet7 is triggered and gets Fileview on UICC. 1.1- Using the Fileview, Applet7 reads file EF_{TARR1}. No exception is expected 1.2- Using the Fileview, Applet7 reads file EF_{TARR5}. An UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown 1.3- Using the Fileview, Applet7 reads file EF_{TARR4}. No exception is expected	1- SW = 90 00
	2- Applet7 deletion		
	3- Applet7 installation with Global PIN1 & ADM2 access right for AdminFileView access domain		
	4- Send an envelope Menu Selection on menu id 2 to trigger Applet7	4- Applet7 is triggered and gets AdminFileview on UICC. 4.1- Using the AdminFileview, Applet7 reads file EF_{TARR1}. An UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown 4.2- Using the AdminFileview, Applet7 reads file EF_{TARR5}. No exception is expected 4.3- Using the AdminFileview, Applet7 reads file EF_{TARR4}. No exception is expected	4- SW = 90 00
	5- Applet7 deletion		
	6- Applet7 installation with Always & ADM1 access right for Fileview access domain and Global PIN1 & ADM2 access right for AdminFileView access domain,		
	7- Send an envelope Menu Selection on menu id 3 to trigger Applet7	7- Applet7 is triggered and gets Fileview and AdminFileview on UICC. 7.1- Using the Fileview, Applet7 reads file EF_{TARR1}. No exception is expected 7.2- Using the AdminFileview, Applet7 reads file EF_{TARR1}. An UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown 7.3- Using the AdminFileview, Applet7 reads file EF_{TARR5}. No exception is expected 7.4- Using the Fileview, Applet7 reads file EF_{TARR5}. An UICCException.SECURITY_STATUS_NOT_SATISFIED is thrown 7.5- Using the Fileview, Applet7 reads file EF_{TARR4}. No exception is expected 7.6- Using the AdminFileview, Applet7 reads file EF_{TARR4}. No exception is expected	7- SW = 90 00
	8- Applet7 deletion		

5.5.7.8 Priority Level

Test Area Reference: Cre_Tin_Prlv.

5.5.7.8.1 Conformance requirement

5.5.7.8.1.1 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).

5.5.7.8.1.2 Parameter errors

No requirements.

5.5.7.8.1.3 Context errors

No requirements.

5.5.7.8.2 Test area files

Test Source: Test_Cre_Tin_Prlv_x.java, x from 1 to 12, 8A, 8B, 9A, 9B, 10A, 10B.

Test Applet: Cre_Tin_Prlv_x.java, x from 1 to 12, 8A, 8B, 9A, 9B, 10A, 10B.

Cap File: Cre_tin_prlv_x.cap, x from 1 to 12, 8A, 8B, 9A, 9B, 10A, 10B.

5.5.7.8.3 Test coverage

CRR number	Test case number
CRRN1	1, 2, 3, 4, 6, 8, 10, 12
CRRN2	5, 7, 9, 11

5.5.7.8.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
0	All applets are registered on an EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event		
1	**Trigger 2 applets with 2 different maximum Priority Levels** 1- Install (install) Applet1 with priority level '2' and Applet2 with priority level '1', from package Cre_tin_prlv_1 2- Send an Envelope that triggers the 2 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 3- Delete applets instances and packages	2- A static variable is used to validate triggering order: Applet2 is triggered before Applet1	

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**Trigger 2 applets with 2 different maximum Priority Levels** 1- Install (install) Applet1 with priority level '1' and Applet2 with priority level '2', from package Cre_tin_prlv_2. 2- Send an Envelope that triggers the 2 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event. 3- Delete applets instances and packages	2- A static variable is used to validate triggering order: Applet1 is triggered before Applet2	
3	**Trigger 2 applets with 2 different Priority Levels** 1- Install (install) Applet1 with priority level '80' and Applet2 with priority level '7F', from package Cre_tin_prlv_3. 2- Send an Envelope that triggers the 2 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 3- Delete applets instances and packages	2- A static variable is used to validate triggering order: Applet2 is triggered before Applet1	
4	**Trigger 2 applets with 2 different Priority Levels** 1- Install (install) Applet1 with priority level '7F' and Applet2 with priority level '80', from package Cre_tin_prlv_4 2- Send an Envelope that triggers the 2 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 3- Delete applets instances and packages	2- A static variable is used to validate triggering order: Applet2 is triggered before Applet1	
5	**Trigger 3 applets with the same Priority Level** 1- Install (install) applet 1, 2, 3 in this order with same priority level from package Cre_tin_prlv_5 2- Send an Envelope that triggers the 3 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 3- Delete applets instances and packages	2- 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** 1- Install (install) Applet1 from class A with priority level '2' Install (install) Applet2 from class B with priority level '1' 2- Send an Envelope that triggers the 2 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 3- Delete applets instances and packages	2- A static variable is used to validate triggering order: Applet2 is triggered before Applet1	

Id	Description	API/CAT RE Expectation	APDU Expectation
7	**Trigger 2 applets from 2 classes, with the same Priority Level** 1- Install (install) Applet1 from class A with priority level '1' Install (install) Applet2 from class B with priority level '1' 2- Send an Envelope that triggers the 2 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 3- Delete applets instances and packages	2- 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** 1- Install package Cre_tin_prlv_8. Install (install) Applet1 from package Cre_tin_prlv_8A with priority level '2' Install (install) Applet2 from package Cre_tin_prlv_8B with priority level '1' 2- Send an Envelope that triggers the 2 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 3- Delete applets instances ad packages	2- 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** 1- Install package Cre_tin_prlv_9. Install (install) Applet1 from package Cre_tin_prlv_9A and Applet2 from package Cre_tin_prlv_9B in this order, with same priority level 2- Send an Envelope that triggers the 2 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 3- Delete applets instances and packages	2- A static variable is used to validate triggering order: Applet2 is triggered before Applet1	
10	**Trigger 4 applets from 2 packages** 1- Install packages Cre_tin_prlv_10, Cre_tin_prlv_10A and Cre_tin_prlv_10B Install (install) 2 applets 1 then 2 from package Cre_tin_prlv_10A, with respectively priority levels 1 and 2 2- Send an Envelope that triggers the 2 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 3- Install (install) 2 applets 3 then 4 from package Cre_tin_prlv_10B, with respectively priority levels 1 and 2 4- Send an Envelope that triggers the 4 applets 5- Delete applets instances and packages	2- A static variable is used to validate triggering order: Applet1 is triggered before Applet2 4- Applet3 is triggered before applets 1, 4, then 2	

Id	Description	API/CAT RE Expectation	APDU Expectation
11	**Trigger 4 applets with the same Priority Level then delete them one after another and trigger them each time** 1- Install (install) applets 1, 2, 3, 4 in this order with same priority level from package Cre_tin_prlv_11 2- Send an Enveloppe that triggers the 4 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 3- Delete applet instance 4 4- Send an Enveloppe that triggers the 3 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 5- Delete applet instance 3 6- Send an Enveloppe that triggers the 2 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 7- Delete remaining applet instances and packages	2- A static variable is used to validate triggering order: applets are triggered in order 4, 3, 2, 1 4- Applets are triggered in order 3, 2, 1 6- Applets are triggered in order 2, 1	
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 2- Send an Enveloppe that triggers the 4 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 3- Delete applet instance 1 and install (install) applet5 with priority level 2 4- Send an Enveloppe that triggers the 4 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event 5- Re-install (install) Applet1 with priority level 1 6- Send an Enveloppe that triggers the 5 applets with the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY event	2- A static variable is used to validate triggering order: applets are triggered in order 3, 1, 4, 2 4- Applets are triggered in order 3, 5, 4, 2 6- Applets are triggered in order 1, 3, 5, 4, 2	

5.5.7.9 Channel Allocation

Test Area Reference: Cre_Tin_Chal.

5.5.7.9.1 Conformance requirement

5.5.7.9.1.1 Normal execution

- CRRN1: One toolkit applet can register to several channels, but a channel can only be allocated to one toolkit applet.

5.5.7.9.1.2 Parameter errors

No requirements.

5.5.7.9.1.3 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 ETSI TS 102 223 [6]), the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

5.5.7.9.2 Test area files

Test Source: Test_Cre_Tin_Chal.java.

Test Applet: Cre_Tin_Chal_1.java.

Cre_Tin_Chal_2.java.

Cre_Tin_Chal_3.java.

Cap File: Cre_tin_chal.cap.

5.5.7.9.3 Test coverage

CRR number	Test case number
CRRN1	2,3
CRRC1	1, 7
CRRC2	4,5,6

5.5.7.9.4 Test procedure

Id	Description	API/CAT RE 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 2- Select the Applet1		2- SW = 6A 80
	Reset the card		
2	**Good installation of Applet2** Install for install of Applet2 (maximum 4 channels allocated).		The UICC answers 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 UICC 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 UICC answers with status words 90 00

Id	Description	API/CAT RE Expectation	APDU Expectation
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 UICC 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 UICC with 'No Channel Available' as Additional Information on Result

5.5.7.10 Minimum Security Level

Test Area Reference: Cre_Tin_Mslv.

5.5.7.10.1 Conformance requirement

5.5.7.10.1.1 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 check fails, the Receiving Entity shall reject the messages and 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.

5.5.7.10.1.2 Parameter errors

No requirements.

5.5.7.10.1.3 Context errors

No requirements.

5.5.7.10.2 Test area files

None.

5.5.7.10.3 Test coverage

CRR number	Test case number
CRRN1	Not applicable
CRRN2	Not applicable
CRRN3	Not applicable
CRRN4	Not applicable
CRRN5	Not applicable

5.5.7.10.4 Test procedure

Not applicable.

5.5.7.11 TAR Value(s) of the Toolkit Application instance

Test Area Reference: Cre_Tin_Tarv.

5.5.7.11.1 Conformance requirement

5.5.7.11.1.1 Normal execution

- CRRN1: It is possible to define several TAR Values at the installation of a Toolkit Application.
- CRRN2: If the length of TAR Value(s) is zero, the TAR may be taken out of the AID, if any.
- CRRN3: If the length of the TAR Value(s) is greater than zero then the application instance shall be installed with the TAR Value(s) field defined above and the TAR indicated in the AID if any shall be ignored.

5.5.7.11.1.2 Parameter errors

No requirements.

5.5.7.11.1.3 Context errors

- CRRC1 If a TAR Value(s) is already assigned on the card for a Toolkit Application instance, the card shall return the Status Word '6A80', incorrect parameters in data field, to the INSTALL [for install] command.
- CRRC1 If the length of TAR Value(s) field is incorrect, the card shall return the Status Word '6A80', incorrect parameters in data field, to the INSTALL [for install] command.

5.5.7.11.2 Test area files

Test Source: Test_Cre_Tin_Tarv.java.

Test Applet: Cre_Tin_Tarv_1.java.

Cre_Tin_Tarv_2.java.

Cap File: Cre_tin_tarv.cap.

5.5.7.11.3 Test coverage

CRR number	Test case number
CRRN1	1 (but partially tested only)
CRRN2	Not applicable
CRRN3	Not applicable
CRRC1	1
CRRC2	2

5.5.7.11.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**TAR value(s) already allocated** 1- Install Applet2 with the TAR "020202" 2- Install Applet1 with the TAR values: "010101" "020202" "030303"		1- SW = RAPDU = 00 90 00 2- SW = 6A 80
2	**Bad TAR value(s)** 1- Install Applet1 with the TAR values: "010101" "0303"		1- SW = 6A 80

5.5.7.12 Services Allocation

Test Area Reference: Cre_Tin_Sval.

5.5.7.12.1 Conformance requirement

5.5.7.12.1.1 Normal execution

- CRRN1: One toolkit applet can allocates several services, but a service can only be allocated to one toolkit applet.

5.5.7.12.1.2 Parameter errors

No requirements.

5.5.7.12.1.3 Context errors

- CRRC1: Allocated services shall not exceed the maximum number of services allowed for this applet instance defined during installation.
- CRRC2: The total number of services allocated for all the applets shall not exceed 8. If the maximum number of services required is greater than '08' (maximum numbers of services specified in ETSI TS 102 223 [6]), the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

5.5.7.12.2 Test area files

Test Source: Test_Cre_Tin_Sval.java.

Test Applet: Cre_Tin_Sval_1.java.

Cre_Tin_Sval_2.java.

Cre_Tin_Sval_3.java.

Cap File: Cre_tin_sval.cap.

5.5.7.12.3 Test coverage

CRR number	Test case number
CRRN1	2, 3, 8
CRRC1	1, 7
CRRC2	4, 5, 6

5.5.7.12.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**More than 8 services at the instantiation of Applet1: check that Applet1 is not installed.** 1- Install Applet1 with maximum 9 services allocated		1- SW = 6A 80
	Reset the card		
2	**Good installation of Applet2** 1- Install Applet2 with maximum 4 services allocated		1- RAPDU = 00 90 00
3	**Allocate 4 services Applet2**	No exception shall be thrown	
4	**Allocate one more service Applet2**	Shall throw a ToolkitException with reason NO_SERVICE_ID_AVAILABLE	
5	**Good installation of applet3** 1- Install Applet3 with maximum 8 services allocated		1- RAPDU = 00 90 00
6	**Allocate 4 services Applet3**	No exception shall be thrown	
7	**Allocate one more service Applet3**	Shall throw a ToolkitException with reason NO_SERVICE_ID_AVAILABLE	
8	**Check that each service identifier (allocated by Applet2 and applet3) is between 0 and 7 and is different from each other**		

5.5.8 UICC File Access

5.5.8.1 FileView

Test Area Reference: Cre_Ufa_View.

5.5.8.1.1 Conformance requirement

5.5.8.1.1.1 Normal execution

- CRRN1: Any Applet (not only Toolkit Applets) is allowed to retrieve and use a *FileView*.
- CRRN2: The UICC *FileView* can be retrieved by invoking the *getTheUICCView()* method from the *UICCSystem*.
- CRRN3: An ADF *FileView* can be retrieved by invoking the *getTheFileView(…)* method with passing as parameter the full AID of the application owning the ADF.
- CRRN4: The UICC *FileView* allows to access the MF and all DFs and EFs that are located under the MF, including DF Telecom and any access technology specific DF located under the MF, but not the files located under any ADF.
- CRRN5: An ADF *FileView* allows to access only the DFs and EFs located under the ADF.
- CRRN6: Each *FileView* object shall be provided as a permanent JCRE entry point object.

- CRRN7: A separate and independent file context shall be associated with each and every *FileView* object: the operation performed on files in a given *FileView* object shall not affect the file context associated with any other *FileView* object.
- CRRN8: The file context can be transient or persistent depending on what was required by the Applet during the creation of the *FileView* object.
- CRRN9: Each *FileView* shall be given the access control privileges associated with the UICC File System or the corresponding ADF for the Applet.
- CRRN10: The access control privileges are checked each time a method of the *FileView* object is invoked. The access control privileges are defined by the access domain parameters specified in ETSI TS 102 226 [8].
- CRRN11: The root of the context of a *FileView* object is the MF for the UICC *FileView*.
- CRRN12: The root of the context of a *FileView* object is the ADF for an ADF *FileView*.
- CRRN13: When the transient context of a *FileView* is cleared, the current DF becomes the root of the *FileView*.

5.5.8.1.1.2 Parameter errors

No requirements.

5.5.8.1.1.3 Context errors

- CRRC1: It is not possible to access the MF or any DF or EF located under the MF from an ADF *FileView*.

5.5.8.1.2 Test area files

Test Source: Test_Cre_Ufa_View.java.

Test Applet: Cre_Ufa_View_1.java.

Cap File: Cre_ufa_view.cap.

5.5.8.1.3 Test coverage

CRR number	Test case number
CRRN1	1, 2, 3
CRRN2	1, 2, 3 and see also Api_1_Cont test cases for CRRN1
CRRN3	1, 2, 3 and see also Api_1_Cont test cases for CRRN1
CRRN4	2 and see also Api_1_Cont test cases for CRRN1
CRRN5	3 and see also Api_1_Cont test cases for CRRN1
CRRN6	4
CRRN7	5, 7 and see also Api_1_Cont test cases for CRRN1
CRRN8	6
CRRN9	See Cre_Tin_Acdo
CRRN10	See Cre_Tin_Acdo
CRRN11	5, 6
CRRN12	5, 6
CRRN13	6
CRRC1	3

5.5.8.1.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Applets can get a *FileView*** 1- A toolkit applet, Applet1, and a non toolkit applet, Applet2, are installed with full access to the file system 2- An envelope menu selection is sent to trigger Applet1 3- Applet1 gets the UICC *FileView* using method getUICCFileView() and the ADF1 *FileView* getTheFileView() 4- Send an APDU to trigger Applet2 5- Applet2 gets the UICC *FileView* using method getUICCFileView() and the ADF1 *FileView* getTheFileView()	2- Applet1 is triggered 3- No exception is thrown 4- Applet2 is triggered 5- No exception is thrown	
2	**Applets can only access all files under the MF using the UICC file view** 1- An envelope menu selection is sent to trigger Applet1 2- Applet1 gets the UICC *FileView* 3- Applet1 selects the MF 4- Applet1 selects the EF_{DIR} 5- Applet1 selects the $DF_{TELECOM}$ 6- Applet1 selects the MF 7- Applet1 selects the DF_{TEST} 8- Applet1 selects the DF_{TEST1} 9- Applet1 selects the EF_{TAA} 10- Applet1 tries to select DF_{ADF1} 11- Send an APDU to trigger Applet2 12- Applet2 gets the UICC *FileView* 13- Applet2 selects the MF 14- Applet2 selects the EF_{DIR} 15- Applet2 selects the $DF_{TELECOM}$ 16- Applet2 selects the MF 17- Applet2 selects the DF_{TEST} 18- Applet2 selects the DF_{TEST1} 19- Applet2 selects the EF_{TAA} 20- Applet2 tries to select DF_{ADF1}	1- Applet1 is triggered 3- No exception is thrown 4- No exception is thrown 5- No exception is thrown 6- No exception is thrown 7- No exception is thrown 8- No exception is thrown 9- No exception is thrown 10- An exception is thrown 11- Applet2 is triggered 13- No exception is thrown 14- No exception is thrown 15- No exception is thrown 16- No exception is thrown 17- No exception is thrown 18- No exception is thrown 19- No exception is thrown 20- An exception is thrown	
3	**Applets can access all files under the ADF1 using the ADF1 file view** 1- An envelope menu selection is sent to trigger Applet1 2- Applet1 gets the ADF1 *FileView* 3- Applet1 selects the ADF1 4- Applet1 selects the $DF_{TELECOM}$ 5- Applet1 selects the ADF1 6- Applet1 selects the DF_{TEST} 7- Applet1 selects the DF_{TEST1} 8- Applet1 selects the EF_{TAA} 9- Applet1 tries to select the MF 10- Applet1 tries to select the EF_{DIR} 11- Applet1 tries to select DF_{ADF2} 12- Send an APDU to trigger Applet2 13- Applet2 gets the ADF1 *FileView* 14- Applet2 selects the ADF1 15- Applet2 selects the $DF_{TELECOM}$ 16- Applet2 selects the ADF1 17- Applet2 selects the DF_{TEST} 18- Applet2 selects the DF_{TEST1} 19- Applet2 selects the EF_{TAA} 20- Applet1 tries to select the MF 21- Applet1 tries to select the EF_{DIR} 22- Applet2 tries to select DF_{ADF2}	1- Applet1 is triggered 3- No exception is thrown 4- No exception is thrown 5- No exception is thrown 6- No exception is thrown 7- No exception is thrown 8- No exception is thrown 9- An exception is thrown 10- An exception is thrown 11- An exception is thrown 12- Applet2 is triggered 14- No exception is thrown 15- No exception is thrown 16- No exception is thrown 17- No exception is thrown 18- No exception is thrown 19- No exception is thrown	

Id	Description	API/CAT RE Expectation	APDU Expectation
		20- An exception is thrown 21- An exception is thrown 22- An exception is thrown	
4	**FileView object shall be provided as a permanent JCRE entry point object** 1- An envelope menu selection is sent to trigger Applet1 2- Applet1 gets an UICC *FileView* in a static field 3- Applet1 gets an ADF1 *FileView* in a static field 4- Applet1 gets an UICC *FileView* in a field of the toolkit applet 5- Applet1 gets an ADF1 *FileView* in a field of the toolkit applet 6- Send an APDU to trigger Applet2 7- Applet2 gets an UICC *FileView* in a static field 8- Applet2 gets an ADF1 *FileView* in a static field 9- Applet2 gets an UICC *FileView* in a field of the toolkit applet 10- Applet2 gets an ADF1 *FileView* in a field of the toolkit applet	 1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown 4- No exception is thrown 5- No exception is thrown 6- Applet2 is triggered 7- No exception is thrown 8- No exception is thrown 9- No exception is thrown 10- No exception is thrown	
5	**Context independence on FileView object** *The following sequence shall be performed twice, once with the UICC FileView, then once with the ADF1 FileView* 1- An envelope menu selection is sent to trigger Applet1 2- Applet1 gets 2 UICC(*or ADF1*) *FileView* and stores them in objects FileView1 and FileView2 3- Applet1 selects DF_{TEST}/EF_{CARU} using the FileView1 object 4- Applet1 calls the readRecord() method in the NEXT mode using the FileView1 object 5- Applet1 calls the readRecord() method in the NEXT mode using the FileView1 object 6- Applet1 calls the readRecord() method in the NEXT mode using the FileView2 object 7- Applet1 selects EF_{LARU} using the FileView2 object 8- Applet1 selects DF_{test} using the FileView2 object 9- Applet1 selects EF_{CARU} using the FileView2 object 10- Applet1 calls the readRecord() method in the NEXT mode using the FileView2 object *The same test sequence is done using the non toolkit applet Applet2*	 1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown 4- The record value is “55 55 55” 5- The record value is “AA AA AA” 6- An exception is thrown 7- An exception is thrown 8- No exception is thrown 9- No exception is thrown 10- The record value is “55 55 55”	

Id	Description	API/CAT RE Expectation	APDU Expectation
6	**File Context can be transient or persistent** *The following sequence shall be performed twice, once with the UICC FileView, then once with the ADF1 FileView* 1- An envelope menu selection is sent to trigger Applet1 2- Applet1 gets 2 UICC(*or ADF1*) *FileView*and stores one in a transient object FileView1 and the other in a persistent object FileView2 3- Applet1 selects DF_{TEST}/EF_{CARU} using the FileView1 object then the FileView2 object 4- Applet1 calls the readRecord() method in the NEXT mode using the FileView1 object then the FileView2 object 5- Reset the card 6- An envelope menu selection is sent to trigger Applet1 7- Applet1 calls the status() command using the FileView1 8- Applet1 calls the status() command using the FileView2 9- Applet1 calls the readRecord() method in the NEXT mode using the FileView2 10- Applet1 calls the readRecord() method in the NEXT mode using the FileView1	1- Applet1 is triggered 2- No exception is thrown 3- No exception is thrown 4- The record value is “55 55 55” 6- Applet1 is triggered 7- The current DF is DF_{TEST} 8- The current DF is the root (MF or ADF1) 9- The record value is “AA AA AA” 10- An exception is thrown	
7	**File Context integrity** 1- An envelope menu selection is sent to trigger Applet1 2- Applet1 gets a UICC *FileView* and selects DF_{TEST}, with it 3- Applet1 sends a Display Text proactive command 4- Send a fetch command 5- An envelope call control by NAA is sent 6- Applet1 selects DF_{TEST}/DF_{TEST1}, using the previous UICC *FileView*, then finalizes 7- Send terminal response of Display Text command 8- Applet1 resumes and calls status() command, using the same UICC *FileView*	1- Applet1 is triggered 2- No exception is thrown 3- SW = 91 XX 4- Display Text is fetched 5- Applet1 is triggered 6- No exception is thrown 7- SW = 90 00 8- The current DF is DF_{TEST1}	

5.5.8.2 File Access

Test Area Reference: Cre_Ufa_Facc.

Shall be covered in the API access part.

5.5.9 Other parts transferred to framework from API

5.5.9.1 A handler is a temporary JCRE Entry Point object

Test Area Reference: Cre_Api_Hepo.

5.5.9.1.1 Conformance requirement

5.5.9.1.1.1 Normal execution

- CRRN1: The EnvelopeHandler is a Temporary JCRE Entry Point Object (see Javacard 2.2.1 Runtime Environment (JCRE) Specification [2]).
- CRRN2: The EnvelopeResponseHandler is a Temporary JCRE Entry Point Object (see Javacard 2.2.1 Runtime Environment (JCRE) Specification [2]).

- CRRN3: The ProactiveHandler is a Temporary JCRE Entry Point Object (see Javacard 2.2.1 Runtime Environment (JCRE) Specification [2]).
- CRRN4: The ProactiveResponseHandler is a Temporary JCRE Entry Point Object (see Javacard 2.2.1 Runtime Environment (JCRE) Specification [2]).

5.5.9.1.1.2 Parameter errors

No requirements.

5.5.9.1.1.3 Context errors

No requirements.

5.5.9.1.2 Test area files

Test Source: Test_Cre_Api_Hepo.java.

Test Applet: Cre_Api_Hepo_1.java.

Cap File: Cre_api_hepo.cap.

5.5.9.1.3 Test coverage

CRR number	Test case number
CRRN1	1, 2
CRRN2	3, 4
CRRN3	5, 6
CRRN4	7, 8

5.5.9.1.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**EnvelopeHandlerSystem.getTheHandler()** `Store it in a static field of the toolkit applet`	SecurityException is thrown	
2	**EnvelopeHandlerSystem.getTheHandler()** `Store it in a field of the toolkit applet`	SecurityException is thrown	
3	**EnvelopeResponseHandlerSystem.getTheHandler()** `Store it in a static field of the toolkit applet`	SecurityException is thrown	
4	**EnvelopeResponseHandlerSystem.getTheHandler()** `Store it in a field of the toolkit applet`	SecurityException is thrown	
5	**ProactiveHandlerSystem.getTheHandler()** `Store it in a static field of the toolkit applet`	SecurityException is thrown	
6	**ProactiveHandlerSystem.getTheHandler()** `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`
	ProactiveResponseHandlerSystem.getTheHandler()
	`Store it in a static field of the toolkit applet`
		SecurityException is thrown	Proactive command is fetched and terminal response is issued

Id	Description	API/CAT RE Expectation	APDU Expectation
8	**ProactiveResponseHandlerSystem.getTheHandler()** `Store it in a field of the toolkit applet`	SecurityException is thrown	

5.5.9.2 Transaction

Test Area Reference: Cre_Api_Tran.

5.5.9.2.1 Conformance requirement

5.5.9.2.1.1 Normal execution

- CRRN1: A pending toolkit applet transaction at the ProactiveHandler.send() method invocation is aborted.
- CRRN2: A pending toolkit applet transaction is aborted on the termination of the toolkit applet (return from the *processToolkit()* method).
- CRRN3: At the invocation of the *processToolkit()* method there shall be no transaction in progress.

5.5.9.2.1.2 Parameter errors

No requirements.

5.5.9.2.1.3 Context errors

No requirements.

5.5.9.2.2 Test area files

Test Source: Test_Cre_Api_Tran.java.

Test Applet: Cre_Api_Tran_1.java.

Cre_Api_Tran_2.java.

Cap File: Cre_api_tran.cap.

5.5.9.2.3 Test coverage

CRR number	Test case number
CRRN1	1
CRRN2	2
CRRN3	Not testable

5.5.9.2.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Verify that transaction is aborted when a proactive command is sent** 1- Applet1 is triggered and performed the following sequence - Initialize a byte field with 0x05 - Build a display text proactive command - beginTransaction() - Update the byte with 0x02 - send the proactive command 3- Applet is resumed - Verify that the byte value is 0x05 - JCSystem.getTransactionDepth()	JCSystem.getTransactionDepth() shall return 0	2- Proactive command fetched and terminal response is issued
2	**Verify that transaction is aborted when a proactive command is sent** 1- Applet2 is triggered and send a display text proactive command Applet1 is triggered and performed the following sequence - Initialize a static byte field with 0x05 - beginTransaction() - Update the byte with 0x02 - Finalize 3- Applet2 is resumed and - Verify that the byte value is 0x05 - JCSystem.getTransactionDepth()	JCSystem.getTransactionDepth() shall return 0	2- SW = 91 XX Proactive command fetched and terminal response is issued

5.5.9.3 Timer Id between Applets

Test Area Reference: Cre_Api_Tmid.

5.5.9.3.1 Conformance requirement

5.5.9.3.1.1 Normal execution

No requirements.

5.5.9.3.1.2 Parameter errors

No requirements.

5.5.9.3.1.3 Context errors

- CRRC1: The method ToolkitRegistry.releaseTimer() shall throw a ToolkitException with INVALID_TIMER_ID reason if the timer is valid but is not allocated to this applet.

5.5.9.3.2 Test area files

Test Source: Test_Cre_Api_Tmid.java.

Test Applet: Cre_Api_Tmid_1.java.

Cap File: Cre_api_tmid.cap.

5.5.9.3.3 Test coverage

CRR number	Test case number
CRRC1	1

5.5.9.3.4 Test procedure

Id	Description	API/CAT RE 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. Trigger second instance and try to call releaseTimer() method with the static field value.	releaseTimer() method call shall throw a ToolkitException with INVALID_TIMER_ID reason	

5.5.10 Registration

5.5.10.1 Event registration

Test Area Reference: Cre_Reg_Evtr.

5.5.10.1.1 Conformance requirement

5.5.10.1.1.1 Normal execution

- CRRN1: A Toolkit Applet can change its registration to toolkit events during its whole life cycle.
- CRRN2: The registration of a Toolkit Applet to an event shall not be affected by its life cycle state.
- CRRN3: The *getShareableInterfaceObject()* has to be called before the applet is triggered the first time.
- CRRN4: The byte parameter of *getShareableInterfaceObject()* method has to be set to '01'.
- CRRN5: The AID parameter of the *getShareableInterfaceObject()* method shall be set to null.

5.5.10.1.1.2 Parameter errors

No requirements.

5.5.10.1.1.3 Context errors

No requirements.

5.5.10.1.2 Test area files

Test Source: Test_Cre_Reg_Evtr.java.

Test Applet: Cre_Reg_Evtr_1.java.

Cre_Reg_Evtr_2.java.

Cap File: Cre_reg_evtr.cap.

5.5.10.1.3 Test coverage

CRR number	Test case number
CRRN1	1 but partially tested
CRRN2	1, and see also CRRN9 of Cre_Pcs_Spco and CRRN3 of Cre_Apt_Eccn
CRRN3	2
CRRN4	2
CRRN5	2

5.5.10.1.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Toolkit applet can change its registration during its whole life cycle** 1- Install Applet1 to let it in the INSTALL state During its install() method, Applet1 registers to EVENT_EVENT_DOWNLOAD_USER_ACTIVITY and EVENT_EVENT_DOWNLOAD_MT_CALL 2- Make selectable Applet1 3- An envelope Event Download User Activity is sent to trigger Applet1 4- Applet1 registers to EVENT_EVENT_DOWNLOAD_LOCATION_STATUS and deregisters to EVENT_EVENT_DOWNLOAD_USER_ACTIVITY	3- Applet1 is triggered	2- Set Up Event List proactive command is fetched with User Activity and MT Call events 4- Set Up Event List proactive command is fetched with Location Status and MT Call events
2	***getShareableInterfaceObject()* has to be called before the first triggering** 1- Install of Applet2 Applet2 *getShareableInterfaceObject()* method increments a counter 2- Trigger Applet2	2- Applet2 performs the following checks: - the counter is incremented - byte parameter of *getShareableInterfaceObject()* method is set to '01' - AID parameter *getShareableInterfaceObject()* method is null	

5.5.11 UICC Toolkit Applet

5.5.11.1 Data and function sharing

Test Area Reference: Cre_Uta_Dafs.

5.5.11.1.1 Conformance requirement

The sharing mechanism defined in "Java Card™ 2.2.1 Application Programming Interface Specification" ([1]) and "Java Card™ 2.2.1 Runtime Environment Specification" ([2]) shall be used by the Toolkit Applet(s) to share data and function.

5.5.11.1.1.1 Normal execution

- CRRN1:The interface shall extend the javacard.framework.shareable interface.
- CRRN2: The server Applet shall overwrite the Applet.getShareableInterfaceObject() method.
- CRRN3: The client Applet shall use the JCSystem.getAppletShareableInterfaceObject() to retrieve a reference to the server Applet shareable interface.
- CRRN4: When the client Applet calls JCSystem.getAppletShareableInterfaceObject() method the Applet.getShareableInterfaceObject() method of the server Applet is called by the CAT RE.

5.5.11.1.1.2 Parameter errors

No requirements.

5.5.11.1.1.3 Context errors

No requirements.

5.5.11.1.2 Test area files

This clause uses 2 packages:

uicc.test.catre.cre_uta_dafs.cre_uta_dafs_a

uicc.test.catre.cre_uta_dafs.cre_uta_dafs_b

Test Source: Test_Cre_Uta_Dafs.java.

Test Applet: Cre_Uta_Dafs_A_1.java (server applet).

Cre_Uta_Dafs_A_2.java (server interface).

Cre_Uta_Dafs_A_3.java (class).

Cre_Uta_Dafs_B_1.java (client applet).

Cap File: Cre_uta_dafs_a.cap.

Cre_uta_dafs_b.cap.

5.5.11.1.3 Test coverage

CRR number	Test case number
CRRN1	1
CRRN2	1
CRRN3	1
CRRN4	1

5.5.11.1.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
0	1- Install packages PackageA then PackageB 2- Install AppletA1 and AppletB1		
1	**Use of a shareable interface** 1- Send an envelope Menu Selection to trigger AppletA1 (Menu Id = 01) 2- AppletA1 stores the menu Id 3- Send an envelope Menu Selection to trigger AppletB1 4- AppletB1 uses the shareable interface of AppletA1 to retrieve the Menu Id that was used to trigger AppletA1 previously	1- AppletA1 is triggered 3- AppletB1 is triggered 4- Menu Id retrieved shall be 01	

5.5.11.2 Package deletion

Test Area Reference: Cre_Uta_Pdel.

5.5.11.2.1 Conformance requirement

The Package deletion mechanism defined in "Java Card™ 2.2.1 Runtime Environment Specification" ([2]) shall be used to delete the content from the UICC.

5.5.11.2.1.1 Normal execution

- CRRN1: If the applet/library package is resident in mutable memory, then the Java Card RE shall delete the applet/library package.
- CRRN2: Following a successful applet/library package deletion, it shall not be possible to install another package which depends on the deleted package.

5.5.11.2.1.2 Parameter errors

No requirements.

5.5.11.2.1.3 Context errors

- CRRC1: The deletion shall fail if a reachable (non-garbage) instance of a class belonging to the package being deleted exists on the card.
- CRRC2: The deletion shall fail if another package on the card depends on this package (as expressed in the CAP file's import component).

5.5.11.2.2 Test area files

This clause uses 2 packages:

uicc.test.catre.cre_uta_pdel.cre_uta_pdel_a

uicc.test.catre.cre_uta_pdel.cre_uta_pdel_b (depends on uicc.test.catre.cre_uta_pdel.cre_uta_pdel_a)

Test Source: Test_Cre_Uta_Pdel.java.

Test Applet: Cre_Uta_Pdel_A_1.java.

Cre_Uta_Pdel_A_2.java.

Cre_Uta_Pdel_A_3.java (server interface) Cre_Uta_Pdel_B_1.java (use class Cre_Uta_Pdel_A_2).

Cre_Uta_Pdel_B_2.java.

Cap File: Cre_uta_pdel_a.cap.

Cre_uta_pdel_b.cap.

5.5.11.2.3 Test coverage

CRR number	Test case number
CRRN1	1 to 5
CRRN2	3
CRRC1	1
CRRC2	2

5.5.11.2.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
0	1- Install packages PackageA then PackageB 2- Install AppletB1		
1	**The package deletion is unsuccessful if a reachable instance of a class belonging to the package exists on the card** 1- Delete PackageB 2- Install AppletB2 3- Delete AppletB1 and AppletB2		1- SW shall be different from 90 00 2- RAPDU = 00 90 00
2	**The package deletion is unsuccessful if another package on the card depends on this package** 1- Delete PackageA 2- Install AppletA1 3- Delete AppletA1		1- SW shall be different from 90 0 02- RAPDU = 00 90 00
3	**The installation of a package which depends on a deleted package shall fail** 1- Delete PackageB 2- Delete PackageA 3- Install PackageB		1- SW = 90 00 2- SW = 90 00 3- SW shall be different from 90 00
4	**Once a package is deleted, it shall not be possible to install an applet from this package** 1- Install AppletA1 2- Install AppletB2		1- SW = SW shall be different from 90 00 2- SW = SW shall be different from 90 00
5	**This test checks that it is possible to re-install the same package** 1- Install PackageA 2- Install AppletA1		2- RAPDU = 00 90 00

5.5.11.3 Applet deletion

Test Area Reference: Cre_Uta_Adel.

5.5.11.3.1 Conformance requirement

The Applet deletion mechanism defined in "Java Card™ 2.2.1 Runtime Environment Specification" ([2]) shall be used to delete the content from the UICC.

5.5.11.3.1.1 Normal execution

- CRRN1: Following a successful applet instance deletion, the Java Card RE shall delete the applet instance.
- CRRN2: Following an unsuccessful applet instance deletion, the applet instance shall be selectable, and all objects owned by the applet shall remain unchanged.
- CRRN3: Following a successful applet instance deletion, it shall not be possible to select that applet, and no object owned by the applet can be accessed by any applet currently on the card or by a new applet created in the future.

5.5.11.3.1.2 Parameter errors

No requirements.

5.5.11.3.1.3 Context errors

- CRRC1: The deletion shall fail if any object owned by the applet instance is referenced from an object owned by another applet instance on the card.
- CRRC2: The deletion shall fail if any object owned by the applet instance is referenced from a static field on any package on the card.
- CRRC3: The deletion shall fail if an applet instance, belonging to the context of the applet instance being deleted, is active (selected) on the card.

5.5.11.3.2 Test area files

This clause uses 2 packages:

uicc.test.catre.cre_uta_adel.cre_uta_adel_a

uicc.test.catre.cre_uta_adel.cre_uta_adel_b

Test Source: Test_Cre_Uta_Adel.java.

Test Applet: Cre_Uta_Adel_A_1.java.

Cre_Uta_Adel_A_2.java.

Cre_Uta_Adel_A_3.java.

Cre_Uta_Adel_B_1.java.

Cre_Uta_Adel_B_2. java.

Cap File: Cre_uta_adel_a.cap.

Cre_uta_adel_b.cap.

5.5.11.3.3 Test coverage

CRR number	Test case number
CRRN1	3, 4
CRRN2	1, 2, 3
CRRN3	4, 5
CRRC1	1
CRRC2	2
CRRC3	3

5.5.11.2.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
0	1- Install packages PackageA then PackageB 2- Install AppletA1, AppletA2, AppletB1 and AppletB2		
1	**The deletion shall fail if any object owned by the applet instance is referenced from an object owned by another applet instance on the card** 1- Send an envelope Menu Selection to trigger AppletA1 2- AppletA1 store the menu Id 3- Send an envelope Menu Selection to trigger AppletA2 4- AppletA2 gets a reference to the shareable interface to retrieve the menu Id used to trigger AppletA1 5- Delete AppletA1 6- Send an envelope Menu Selection to trigger AppletA1 7- Send an envelope Menu Selection to trigger AppletA2 8- AppletA2 frees the reference to the shareable interface 9- Delete AppletA1 10- Install AppletA1 11- Send an envelope Menu Selection to trigger AppletA1 12- AppletA1 store the menu Id 13- Send an envelope Menu Selection to trigger AppletB2 14- AppletB2 gets a reference to the shareable interface to retrieve the menu Id used to trigger AppletA1 15- Delete AppletA1 16- Send an envelope Menu Selection to trigger AppletA1 17- Send an envelope Menu Selection to trigger AppletB2 18- AppletB2 frees reference to the shareable interface 19- Delete AppletA1 20- Install AppletA1	1- AppletA1 is triggered 3- AppletA2 is triggered 6- AppletA1 is triggered 7- AppletA2 is triggered 11- AppletA1 is triggered 13- AppletB2 is triggered 16- AppletA1 is triggered 17- AppletB2 is triggered	5- SW shall be different from 90 00 7- SW = 90 00 9- SW = 90 00 10- RAPDU = 00 90 00 15- SW shall be different from 90 00 17- SW = 90 00 19- SW = 90 00 20- RAPDU = 00 90 00

Id	Description	API/CAT RE Expectation	APDU Expectation
2	**The deletion shall failed if any object owned by the applet instance is referenced from a static field on any package on the card**		
	1- Send an envelope Menu Selection to trigger AppletA1	1- AppletA1 is triggered	
	2- AppletA1 store the menu Id 3- Send an envelope Menu Selection to trigger AppletA2		
	4- AppletA2 gets a reference to the shareable interface and stores it in a static field of PackageA, then retrieves the menu Id used to trigger AppletA1	3- AppletA2 is triggered	
	5- Delete AppletA2		5- SW = 90 00
	6- Delete AppletA1		6- SW shall be different from 90 00
	7- Send an envelope Menu Selection to trigger AppletA1	7- AppletA1 is triggered	
	8- AppletA1 frees the reference to the shareable interface		
	9- Delete AppletA1		9- SW = 90 00
	10- Install AppletA1		10- RAPDU = 00 90 00
	11- Send an envelope Menu Selection to trigger AppletA1	11- AppletA1 is triggered	
	12- AppletA1 store the menu Id 13- Send an envelope Menu Selection to trigger AppletB2	13- AppletB2 is triggered	
	14- AppletB2 gets a reference to the shareable interface and stores it in a static field of PackageB, then retrieves the menu Id used to trigger AppletA1		
	15- Delete AppletB2		15- SW = 90 00
	16- Delete AppletA1		16- SW shall be different from 90 00
	17- Send an envelope Menu Selection to trigger AppletA1	17- AppletA1 is triggered	
	18- Send an envelope Menu Selection to trigger AppletB1	18- AppletB1 is triggered	
	19- AppletB1 then frees the reference to the shareable interface		
	20- Delete AppletA1		20- SW = 90 00
	21- Install AppletA1		21- RAPDU = 00 90 00
3	**Deletion of an active applet instance**		
	1- Delete AppletB1		1- SW = 90 00
	2- Install AppletB1		2- RAPDU = 00 90 00
	3- Open another channel and select AppletB1 the new open channel		3- RAPDU = <applet selected data> 90 00
	4- Delete AppletB1 on channel 0		4- SW shall be different from 90 00
	5- Select AppletB1 on Channel 0		5- RAPDU = <applet selected data> 90 00
	6- Reset		

Id	Description	API/CAT RE Expectation	APDU Expectation
4	**Selection of a deleted applet instance** 1- Delete AppletB1 2- Select AppletB1 on Channel 0 3- Install AppletB1		1- SW = 90 00 2- SW shall be different from 90 00 3- RAPDU = 00 90 00
5	**Object owned by a deleted applet can't be accessed by other applets** 1- Delete AppletA1 2- Send an envelope Menu Selection to trigger AppletB1 3- AppletB1 gets a reference to the shareable interface to retrieve the menu Id used to trigger AppletA1	2- AppletB1 is triggered 3- An exception is thrown	1- SW = 90 00

5.5.11.4 Object deletion

Test Area Reference: Cre_Uta_Odel.

5.5.11.4.1 Conformance requirement

5.5.11.4.1.1 Normal execution

- CRRN1: If an object deletion mechanism is supported then the one defined in "Java Card™ 2.2.1 Application Programming Interface Specification" ([1]) shall be used.

5.5.11.4.1.2 Parameter errors

No requirements.

5.5.11.4.1.3 Context errors

No requirements.

5.5.11.4.2 Test area files

Test Script: Test_Cre_Uta_Odel.java

Test Applet: Cre_Uta_Odel_1.java

Cap File: Cre_uta_odel.cap

5.5.11.4.3 Test coverage

CRR number	Test case number
CRRN1	1

5.5.11.4.4 Test procedure

Id	Description	API/CAT RE Expectation	APDU Expectation
1	**Object deletion** 1- Applet calls JCSystem.isObjectDeletionSupported()	2- Returns TRUE	

5.5.12 Proactive Command Handling

5.5.12.1 General behaviour

Test Area Reference: Cre_Pch_Genb.

5.5.12.1.1 Conformance requirement

5.5.12.1.1.1 Normal execution

- CRRN1: On the call to the *send()* method, the CAT Runtime Environment shall handle the transmission of the proactive command to the terminal, and the reception of the response.
- CRRN2: The CAT Runtime Environment shall resume the Toolkit Applet execution on the return from the *send()* method.

5.5.12.1.1.2 Parameter errors

No requirements.

5.5.12.1.1.3 Context errors

No requirements.

5.5.12.1.2 Test area files

None.

5.5.12.1.3 Test coverage

CRR number	Test case number
CRRN1	See test case 2 of Cre_Hin_Prhd

5.5.12.1.4 Test procedure

None.

5.5.13 CAT Runtime Environment behaviour

5.5.13.1 Context

Test Area Reference: Cre_Cat_Ctxt.

5.5.13.1.1 Conformance requirement

5.5.13.1.1.1 Normal execution

- CRRN1: At the invocation of the *processToolkit()* method the context as defined in Java Card shall be set to the context of the Toolkit Applet. The previous context (context of the caller) shall be the context of the CAT Runtime Environment.
- CRRN2: During the execution there might be other context switches, but at the return of the *send()* method the toolkit applet context is restored.

5.5.13.1.1.2 Parameter errors

No requirements.

5.5.13.1.1.3 Context errors

No requirements.

5.5.13.1.2 Test area files

Test Source: Test_Cre_Cat_Ctxt.java.

Test Applet: Cre_Cat_Ctxt_1.java.

Cap File: Cre_cat_ctxt.cap.

5.5.13.1.3 Test coverage

CRR number	Test case number
CRRN1	Not Testable
CRRN2	see Cre_Hin_Enhd

5.5.13.1.4 Test procedure

Not applicable.

5.5.14 UICC and ADF File System Administration API

5.5.14.1 AdminFile View

Test Area Reference: Cre_Fsa_View

5.5.14.1.1 Conformance requirement

5.5.14.1.1.1 Normal execution

- CRRN1: *AdminFileView* objects follow the behaviour of *FileView* objects and inherit *FileView* functionality.
- CRRN2: An *AdminFileView* object can be retrieved by invoking one of the *getAdminFileView()* methods defined in the AdminFileViewBuilder class.
- CRRN3: Each *AdminFileView* shall be given the access control privileges associated with the UICC or the corresponding ADF for the Applet.
- CRRN4: The access control privileges are checked against the access rules defined in ETSI TS 102 221 [5] each time a method of the *AdminFileView* object is invoked.

5.5.14.1.1.2 Parameter errors

No requirements.

5.5.14.1.1.3 Context errors

No requirements.

5.5.14.1.2 Test area files

None.

5.5.14.1.3 Test coverage

CRR number	Test case number
CRRN1	See API_4_Afv_xxxx
CRRN2	See API_4_Afb_xxxx
CRRN3	See Cre_Tin_Acdo
CRRN4	See Cre_Tin_Acdo

5.5.14.1.4 Test procedure

None.

5.5.14.2 AdminFile Access

Test Area Reference: Cre_Fsa_Aacc.

See API access.administration part.

Annex A (normative): Class and methods acronyms

A.1 uicc.access package

Interface / Class Name	Acronyms
FileView	Fvw
UICCConstants	Uct
UICCSystem	Usy
UICCException	Uex

A.1.1 FileView methods

Method Name	Acronyms
public void activateFile()	Actf
public void deactivateFile()	Dacf
public short increase(byte[] incr, short incrOffset, short incrLength, byte[] resp, short respOffset)	Incr
public short readBinary(short fileOffset, byte[] resp, short respOffset, short respLength)	Redb
public short readRecord(short recNumber, byte mode, short recOffset, byte[] resp, short respOffset, short respLength)	Redr
public short searchRecord(byte mode, short recordNum, short searchIndication, byte[] patt, short pattOffset, short pattLength, short[] response, short respOffset, short respLength)	Sear
public void select(byte sfi)	Slctb
public void select(short fid)	Slcts
public short select(short fid, byte[] fcp, short fcpOffset, short fcpLength)	Slct_Bss
public short status(byte[] fcp, short fcpOffset, short fcpLength)	Stat
public void updateBinary(short fileOffset, byte[] data, short dataOffset, short dataLength)	Updb
public void updateRecord(short recNumber, byte mode, short recOffset, byte[] data, short dataOffset, short dataLength)	Updr

A.1.2 UICCConstants

Method Name	Acronyms
Constants	Cnst

A.1.3 UICCSystem methods

Method Name	Acronyms
public static FileView getTheFileView(javacard.framework.AID aid, byte event)	Getfob
public static FileView getTheFileView(byte[] buffer, short bOffset, byte bLength, byte event)	Getf_Bsbb
public static FileView getTheUICCView(byte event)	Getfb

A.1.4 UICCException methods

Method Name	Acronyms
public UICCException(short reason)	Coor
public static void throwIt(short reason)	Thit
Constants	Cons

A.2 uicc.toolkit package

BERTLVEditHandler	Bte
BERTLVViewHandler	Btv
EditHandler	Edh
EnvelopeHandler	Enh
EnvelopeResponseHandler	Erh
ProactiveHandler	Pah
ProactiveResponseHandler	Prh
ToolkitConstants	Tkc
ToolkitInterface	Tki
ToolkitRegistry	Tkr
ViewHandler	Vwh
EnvelopeHandlerSystem	Ehs
EnvelopeResponseHandlerSystem	Ers
ProactiveHandlerSystem	Phs
ProactiveResponseHandlerSystem	Prs
TerminalProfile	Tep
ToolkitRegistrySystem	Trs
ToolkitException	Tke

A.2.1 BERTLVEditHandler methods

Method Name	Acronyms
public void setTag(byte bBERTag)	Sttg
Inherited method name: EditHandler	
void appendArray(byte[] buffer, short offset, short length)	Apda
void appendTLV(byte tag, byte value)	Aptlbb
void appendTLV(byte tag, byte[] value, short valueOffset, short valueLength)	Aptlb_Bss
void appendTLV(byte tag, byte[] value1, short value1Offset, short value1Length, byte[] value2, short value2Offset, short value2Length)	Aptlb_Bss_Bss
void appendTLV(byte tag, byte value1, byte value2)	Aptlbbb
void appendTLV(byte tag, byte value1, byte[] value2, short value2Offset, short value2Length)	Aptlbb_Bss
void appendTLV(byte tag, byte value1, short value2)	Aptlbbs
void appendTLV(byte tag, short value)	Aptlbs
void appendTLV(byte tag, short value1, short value2)	Aptlbss
void clear()	Cler
Inherited method name: ViewHandler	
byte compareValue(short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)	Cprv
short copy(byte[] dstBuffer, short dstOffset, short dstLength)	Copy
short copyValue(short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	Cpyv
byte findAndCompareValue(byte tag, byte[] compareBuffer, short compareOffset)	Facrb_Bs

Method Name	Acronyms
byte findAndCompareValue(byte tag, byte occurence, short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)	Facrbbs_Bss
short findAndCopyValue(byte tag, byte[] dstBuffer, short dstOffset)	Facyb_Bs
short findAndCopyValue(byte tag, byte occurence, short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	Facybbs_Bss
byte findTLV(byte tag, byte occurrence)	Find
short getCapacity()	Gcap
short getLength()	Glen
byte getValueByte(short valueOffset)	Gvby
short getValueLength()	Gvle
short getValueShort(short valueOffset)	Gvsh
Inherited method name: BERTLVViewHandler	
short getSize()	Gtsz
byte getTag()	Gttg

A.2.2 BERTLVViewHandler methods

Tests are done in inheriting interfaces BERTLVEditHandler and envelopeHandler.

A.2.3 EditHandler methods

Tests are done in inheriting interfaces EnvelopeResponseHandler and ProactiveHandler.

A.2.4 EnvelopeHandler methods

Method Name	Acronyms
Byte getChannelIdentifier()	Gcid
Short getChannelStatus(byte channelIdentifier)	Gcst
Byte getItemIdentifier()	Giid
Inherited method name: BERTLVViewHandler	
short getSize()	Gtsz
byte getTag()	Gttg
Inherited method name: ViewHandler	
byte compareValue(short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)	Cprv
short copy(byte[] dstBuffer, short dstOffset, short dstLength)	Copy
short copyValue(short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	Cpyv
byte findAndCompareValue(byte tag, byte[] compareBuffer, short compareOffset)	Facrb_Bs
byte findAndCompareValue(byte tag, byte occurence, short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)	Facrbbs_Bss
short findAndCopyValue(byte tag, byte[] dstBuffer, short dstOffset)	Facyb_Bs
short findAndCopyValue(byte tag, byte occurence, short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	Facybbs_Bss
byte findTLV(byte tag, byte occurrence)	Find
short getCapacity()	Gcap
short getLength()	Glen
byte getValueByte(short valueOffset)	Gvby
short getValueLength()	Gvle
short getValueShort(short valueOffset)	Gvsh

A.2.5 EnvelopeResponseHandler methods

Method Name	Acronyms
void post(boolean value)	Post
void postAsBERTLV(boolean value, byte tag)	Pabt
Inherited method name: EditHandler	
void appendArray(byte[] buffer, short offset, short length)	Apda
void appendTLV(byte tag, byte value)	Aptlbb
void appendTLV(byte tag, byte[] value, short valueOffset, short valueLength)	Aptlb_Bss
void appendTLV(byte tag, byte[] value1, short value1Offset, short value1Length, byte[] value2, short value2Offset, short value2Length)	Aptlb_Bss_Bss
void appendTLV(byte tag, byte value1, byte value2)	Aptlbbb
void appendTLV(byte tag, byte value1, byte[] value2, short value2Offset, short value2Length)	Aptlbb_Bss
void appendTLV(byte tag, byte value1, short value2)	Aptlbbs
void appendTLV(byte tag, short value)	Aptlbs
void appendTLV(byte tag, short value1, short value2)	Aptlbss
void clear()	Cler
Inherited method name: ViewHandler	
byte compareValue(short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)	Cprv
short copy(byte[] dstBuffer, short dstOffset, short dstLength)	Copy
short copyValue(short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	Cpyv
byte findAndCompareValue(byte tag, byte[] compareBuffer, short compareOffset)	Facrb_Bs
byte findAndCompareValue(byte tag, byte occurence, short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)	Facrbbs_Bss
short findAndCopyValue(byte tag, byte[] dstBuffer, short dstOffset)	Facyb_Bs
short findAndCopyValue(byte tag, byte occurence, short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	Facybbs_Bss
byte findTLV(byte tag, byte occurrence)	Find
short getCapacity()	Gcap
short getLength()	Glen
byte getValueByte(short valueOffset)	Gvby
short getValueLength()	Gvle
short getValueShort(short valueOffset)	Gvsh

A.2.6 ProactiveHandler methods

Method Name	Acronyms
void init(byte type, byte qualifier, byte dstDevice)	Init
void initCloseChannel(byte bChannelId)	Icch
void initDisplayText(byte qualifier, byte dcs, byte[] buffer, short offset, short length)	Indt
void initGetInkey(byte qualifier, byte dcs, byte[] buffer, short offset, short length)	Ingk
void initGetInput(byte qualifier, byte dcs, byte[] buffer, short offset, short length, short minRespLength, short maxRespLength)	Ingi
void initMoreTime()	Inmt
byte send()	Send
Inherited method name: EditHandler	
void appendArray(byte[] buffer, short offset, short length)	Apda
void appendTLV(byte tag, byte value)	Aptlbb
void appendTLV(byte tag, byte[] value, short	Aptlb_Bss

Method Name	Acronyms
valueOffset, short valueLength)	
void appendTLV(byte tag, byte[] value1, short value1Offset, short value1Length, byte[] value2, short value2Offset, short value2Length)	Aptlb_Bss_Bss
void appendTLV(byte tag, byte value1, byte value2)	Aptlbbb
void appendTLV(byte tag, byte value1, byte[] value2, short value2Offset, short value2Length)	Aptlbb_Bss
void appendTLV(byte tag, byte value1, short value2)	Aptlbbs
void appendTLV(byte tag, short value)	Aptlbs
void appendTLV(byte tag, short value1, short value2)	Aptlbss
void clear()	Cler
Inherited method name: ViewHandler	
byte compareValue(short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)	Cprv
short copy(byte[] dstBuffer, short dstOffset, short dstLength)	Copy
short copyValue(short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	Cpyv
byte findAndCompareValue(byte tag, byte[] compareBuffer, short compareOffset)	Facrb_Bs
byte findAndCompareValue(byte tag, byte occurence, short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)	Facrbbs_Bss
short findAndCopyValue(byte tag, byte[] dstBuffer, short dstOffset)	Facyb_Bs
short findAndCopyValue(byte tag, byte occurence, short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	Facybbs_Bss
byte findTLV(byte tag, byte occurrence)	Find
short getCapacity()	Gcap
short getLength()	Glen
byte getValueByte(short valueOffset)	Gvby
short getValueLength()	Gvle
short getValueShort(short valueOffset)	Gvsh

A.2.7 ProactiveResponseHandler methods

Method Name	Acronyms
short copyAdditionalInformation(byte[] dstBuffer, short dstOffset, short dstLength)	Cpai
short copyChannelData(byte[] dstBuffer, short dstOffset, short dstLength)	Cchd
short copyTextString(byte[] dstBuffer, short dstOffset)	Cpts
short getAdditionalInformationLength()	Gtil
byte getChannelIdentifier()	Gcid
short getChannelStatus(byte channelIdentifier)	Gcst
byte getGeneralResult()	Gtgr
byte getItemIdentifier()	Gtii
byte getTextStringCodingScheme()	Gtcs
short getTextStringLength()	Gttl
Inherited method name: ViewHandler	
byte compareValue(short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)	Cprv
short copy(byte[] dstBuffer, short dstOffset, short dstLength)	Copy
short copyValue(short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	Cpyv
byte findAndCompareValue(byte tag, byte[] compareBuffer, short compareOffset)	Facrb_Bs
byte findAndCompareValue(byte tag, byte occurence, short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)	Facrbbs_Bss

Method Name	Acronyms
short findAndCopyValue(byte tag, byte[] dstBuffer, short dstOffset)	Facyb_Bs
short findAndCopyValue(byte tag, byte occurence, short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)	Facybbs_Bss
byte findTLV(byte tag, byte occurrence)	Find
short getCapacity()	Gcap
short getLength()	Glen
byte getValueByte(short valueOffset)	Gvbys
short getValueLength()	Gvle
short getValueShort(short valueOffset)	Gvsh

A.2.8 ToolkitConstants methods

Method Name	Acronyms
Constants	Cons

A.2.9 ToolkitInterface methods

Method Name	Acronyms
void processToolkit(short event)	Prtk

A.2.10 ToolkitRegistry methods

Method Name	Acronyms
byte allocateServiceIdentifier()	Asid
byte allocateTimer()	Atim
void changeMenuEntry(byte id, byte[] menuEntry, short offset, short length, byte nextAction, boolean helpSupported, byte iconQualifier, short iconIdentifier)	Cmet
void clearEvent(short event)	Cevt
public void deregisterFileEvent(short event, byte[] baFileList, short sOffset1, short sLength1, byte[] baADFAid, short sOffset2, byte bLength2)	Drfes_Bss_Bss
public void deregisterFileEvent(short event, FileView aFileView)	Drfeso
void disableMenuEntry(byte id)	Dmet
void enableMenuEntry(byte id)	Emet
short getPollInterval()	Gpol
byte initMenuEntry(byte[] menuEntry, short offset, short length, byte nextAction, boolean helpSupported, byte iconQualifier, short iconIdentifier)	Imet
boolean isEventSet(short event)	Ievs
public void registerFileEvent(short event, byte[] baFileList, short sOffset1, short sLength1, byte[] baADFAid, short sOffset2, byte bLength2)	Rgfes_Bss_Bsb
public void registerFileEvent(short event, FileView aFileView)	Rgfeso
void releaseServiceIdentifier(byte serviceIdentifier)	Rsid
void releaseTimer(byte timerIdentifier)	Rtim
void requestPollInterval(short duration)	Rpol
void setEvent(short event)	Sevt
void setEventList(short[] eventList, short offset, short length)	Sevl
public void setMenuEntryTextAttribute(byte id, byte[] textAttribute, short offset, short length)	Smta

A.2.11 ViewHandler methods

Tests are done in inheriting interfaces EnvelopeHandler, EnvelopeResponseHandler, ProactiveHandler and ProactiveResponseHandler.

A.2.12 EnvelopeHandlerSystem methods

Method Name	Acronyms
`public static EnvelopeHandler getTheHandler()`	Gthd

A.2.13 EnvelopeResponseHandlerSystem methods

Method Name	Acronyms
`public static EnvelopeResponseHandler getTheHandler()`	Gthd

A.2.14 ProactiveHandlerSystem methods

Method Name	Acronyms
`public static ProactiveHandler getTheHandler()`	Gthd

A.2.15 ProactiveResponseHandlerSystem methods

Method Name	Acronyms
`public static ProactiveResponseHandler getTheHandler()`	Gthd

A.2.16 TerminalProfile methods

Method Name	Acronyms
`static boolean check(byte index)`	Checb
`static boolean check(byte[] mask, short offset, short length)`	Chec_Bss
`static boolean check(short index)`	Checs
`static short copy(short startOffset, byte[] dstBuffer, short dstOffset, short dstLength)`	Copy
`static short getValue(short idx)`	Gvals
`static short getValue(short indexMSB, short indexLSB)`	Gvalss

A.2.17 ToolkitRegistrySystem methods

Method Name	Acronyms
`public static ToolkitRegistry getEntry()`	Gety

A.2.18 ToolkitException methods

Method Name	Acronyms
`Constants`	Cons
`public ToolkitException(short reason)`	Coor
`public static void throwIt(short reason)`	Thit

A.3 uicc.system package

HandlerBuilder	Hbd
UICCPlatform	Upt

A.3.1 BERTLVEditHandler methods

Method Name	Acronyms
static ViewHandler buildTLVHandler(byte type, short capacity)	Bthdbs
static ViewHandler buildTLVHandler(byte type, short capacity, byte[] buffer, short bOffset, short bLength)	Bthdbs_Bss

A.3.2 UICCPlatform methods

Method Name	Acronyms
public static byte[] getTheVolatileByteArray()	Gvba

A.4 uicc.access.fileadministration package

Interface / Class Name	Acronyms
AdminFileView	Afv
AdminFileViewBuilder	Afb
AdminException	Aex

A.4.1 AdminFileView methods

Method Name	Acronyms
public void createFile(ViewHandler viewHandler)	Crtf
public void deleteFile(short fid)	Dltf
public void resizeFile(ViewHandler viewHandler)	Rszf
Inherited method name: FileView	
public void activateFile()	Actf
public void deactivateFile()	Dacf
public short increase(byte[] incr, short incrOffset, short incrLength, byte[] resp, short respOffset)	Incr
public short readBinary(short fileOffset, byte[] resp, short respOffset, short respLength)	Redb
public short readRecord(short recNumber, byte mode, short recOffset, byte[] resp, short respOffset, short respLength)	Redr
public short searchRecord(byte mode, short recordNum, short searchIndication, byte[] patt, short pattOffset, short pattLength, short[] response, short respOffset, short respLength)	Sear
public void select(byte sfi)	Slctb
public void select(short fid)	Slcts
public short select(short fid, byte[] fcp, short fcpOffset, short fcpLength)	Slct_Bss
public short status(byte[] fcp, short fcpOffset, short fcpLength)	Stat
public void updateBinary(short fileOffset, byte[] data, short dataOffset, short dataLength)	Updb
public void updateRecord(short recNumber, byte mode, short recOffset, byte[] data, short dataOffset, short dataLength)	Updr

A.4.2 AdminFileViewBuilder methods

Method Name	Acronyms
`public static AdminFileView getTheUICCAdminFileView(byte bType)`	Gtafb
`public static AdminFileView getTheAdminFileView(byte[] buffer, short bOffset, byte bLength, byte bType)`	Gtaf_Bsbb
`public static AdminFileView getTheAdminFileView(javacard.framework.AID aid, byte bType)`	Gtafob

A.4.3 AdminException methods

Method Name	Acronyms
`Constants`	Cons
`public AdminException(short reason)`	Coor
`public static void throwIt(short reason)`	Thit

A.5 Acronyms for CAT Runtime Environment tests

Minimum handler availability	Mha
Handler integrity	Hin
Applet triggering	Apt
Proactive command sending by the CAT Runtime Environment	Pcs
Exception handling	Exh
Envelope response posting	Erp
Toolkit installation	Tin
UICC file access	Ufa
Other parts transferred from API to CAT RE	Api
Registration	Reg
UICC toolkit applet	Uta
Proactive command handling	Pch
CAT Runtime Environment behaviour	Cat

A.5.1 Minimum handler availability

Method Name	Acronyms
ProactiveHandler	Pahd
ProactiveResponseHandler	Prhd
EnvelopeHandler	Enhd
EnvelopeResponseHandler	Erhd

A.5.2 Handler integrity

Method Name	Acronyms
ProactiveHandler	Pahd
ProactiveResponseHandler	Prhd
EnvelopeHandler	Enhd
EnvelopeResponseHandler	Erhd

A.5.3 Applet triggering

Method Name	Acronyms
General behaviour	Genb
EVENT_PROFILE_DOWNLOAD	Epdw
EVENT_MENU_SELECTION	Emse
EVENT_MENU_SELECTION_HELP_REQUEST	Emsh
EVENT_CALL_CONTROL_BY_NAA	Eccn
EVENT_TIMER_EXPIRATION	Etex
EVENT_EVENT_DOWNLOAD_MT_CALL	Edmc
EVENT_EVENT_DOWNLOAD_CALL_CONNECTED	Edcc
EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED	Edcd
EVENT_EVENT_DOWNLOAD_LOCATION_STATUS	Edls
EVENT_EVENT_DOWNLOAD_USER_ACTIVITY	Edua
EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE	Edis
EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS	Edcr
EVENT_UNRECOGNIZED_ENVELOPE	Euev
EVENT_STATUS_COMMAND	Estc
EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION	Edlg
EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION	Edbt
EVENT_FIRST_COMMAND_AFTER_ATR	Efca
EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE	Edda
EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS	Edcs
EVENT_EVENT_DOWNLOAD_ACCESS_TECHNOLOGY_CHANGE	Edat
EVENT_EVENT_DOWNLOAD_DISPLAY_PARAMETER_CHANGED	Eddp
EVENT_EVENT_DOWNLOAD_LOCAL_CONNECTION	Edlc
EVENT_APPLICATION_DESELECT	Eade
EVENT_PROACTIVE_HANDLER_AVAILABLE	Epha
EVENT_EVENT_DOWNLOAD_NETWORK_SEARCH_MODE_CHANGE	Edns
EVENT_EVENT_DOWNLOAD_BROWSING_STATUS	Edbs

A.5.4 Proactive command sending by the CAT Runtime Environment

Method Name	Acronyms
System Proactive Commands	Spco
Interaction with UICC commands	Igco
Proactive Command Control	Pcco

A.5.5 Exception handling

Method Name	Acronyms
General Behaviour	Genb
Interaction with Multiple Triggering	Imtg

A.5.6 Envelope response posting

Method Name	Acronyms
EVENT_CALL_CONTROL_BY_NAA	Eccn
EVENT_UNRECOGNIZED_ENVELOPE	Euen
General Behaviour	Genb

A.5.7 Toolkit installation

Method Name	Acronyms
General Behaviour	Genb
Timers Allocation	Tmal
Item Identifier	Itid
Item Position	Itpo
Maximum Text Length for a menu entry	Mlme
Maximum number of menu entries	Nbme
Access Domain	Acdo
Priority Level	Prlv
Channel Allocation	Chal
Minimum Security Level	Mslv
TAR Value(s) of the Toolkit Application instance	Tarv
Services Allocation	Sval

A.5.8 UICC file access

Method Name	Acronyms
File View	View
File Access	Facc

A.5.9 Other parts transferred from API to CAT RE

Method Name	Acronyms
A handler is a temporary JCRE Entry Point object	Hepo
Transaction	Tran
Timer Id between Applets	Tmid

A.5.10 Registration

Method Name	Acronyms
Event registration	Evtr

A.5.11 UICC toolkit applet

Method Name	Acronyms
Data and function sharing	Dafs
Package deletion	Pdel
Applet deletion	Adel
Object deletion	Odel

A.5.12 Proactive command handling

Method Name	Acronyms
General behaviour	Genb

A.5.13 CAT Runtime Environment behaviour

Method Name	Acronyms
Context	Ctxt

Annex B (normative): Global prepersonalization

This annex defines the file systems used to pass the test suite. It is composed of the UICC File System Server (including MF and all DFs and EFs that are located under the MF) and of two Application Dedicated Files System Servers, ADF1 and ADF2.

ADF1 and ADF2 shall use the following AIDs (as defined in section "4.7 AID Coding"):

- ADF1: A0 00 00 00 09 00 05 FF FF FF FF 89 E0 00 00 02
- ADF2: A0 00 00 00 09 00 05 FF FF FF FF 89 D0 00 00 02

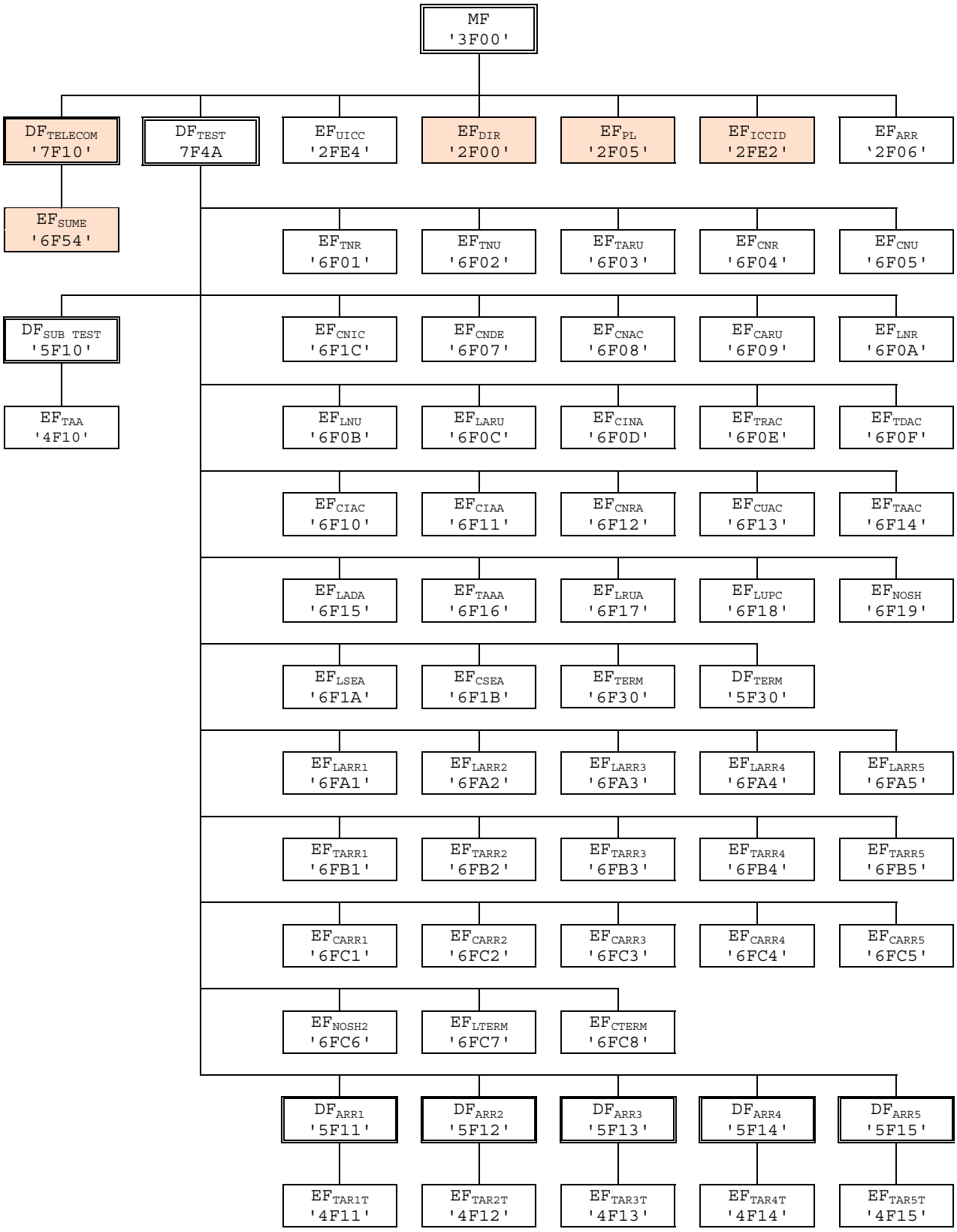
Under the MF, the mandatory files defined in TS 102 221 [5] are present, as the $DF_{TELECOM}$ with the EF_{SUME} as defined in TS 102 222 [7] and some other files necessary for the tests.

The files defined for ADF1 and ADF2 are only necessary for the tests.

All files are shareable if not specified otherwise.

Except if specified the access conditions on DFs shall be set to ALWAYS for all commands.

The global directory structures are defined in figures B.1 to B.3:



NOTE: Files with shaded background are mandatory and defined in ETSI TS 102 221 [5].

Figure B.1: File identifiers and directory structures under MF

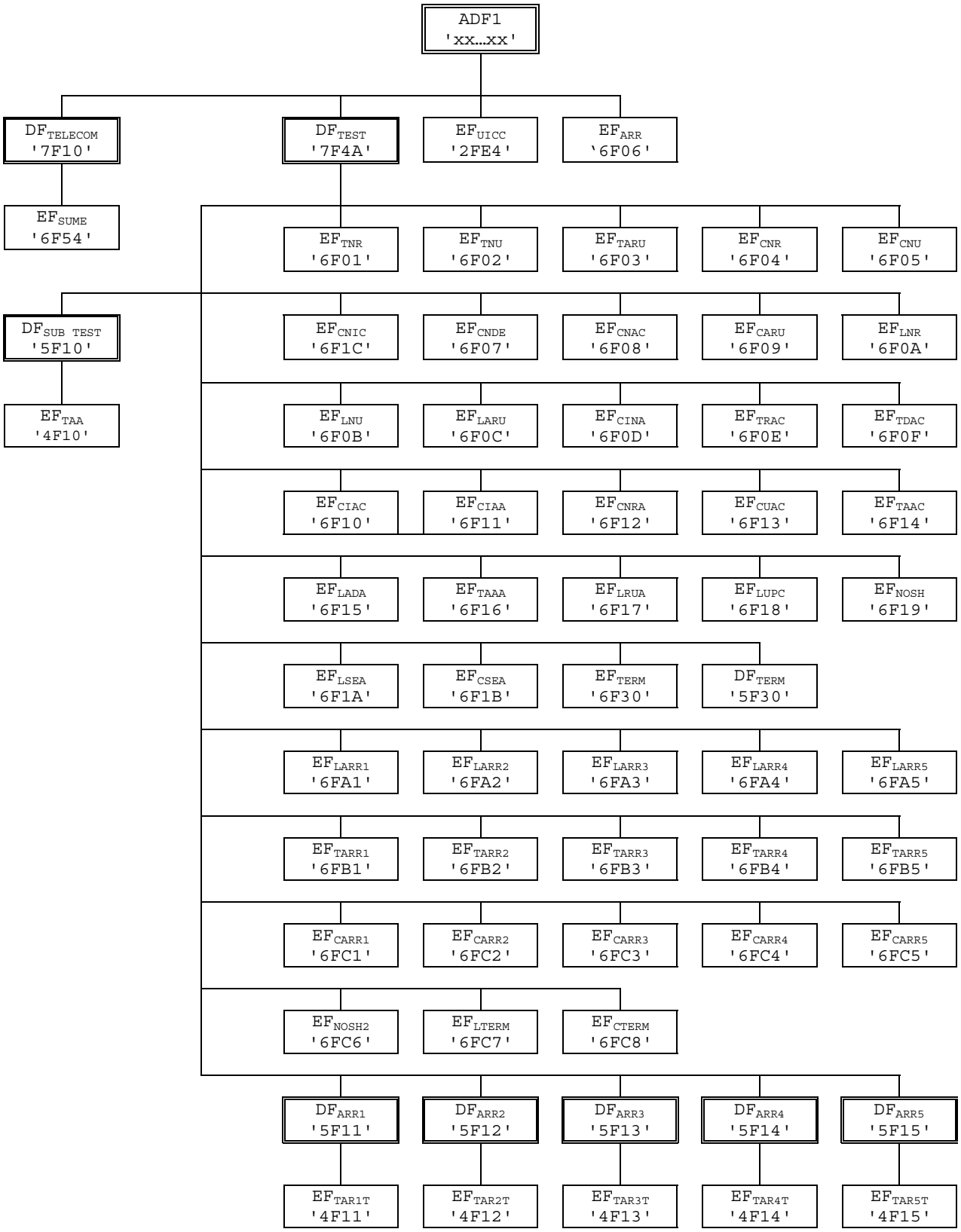


Figure B.2: File identifiers and directory structures under ADF1

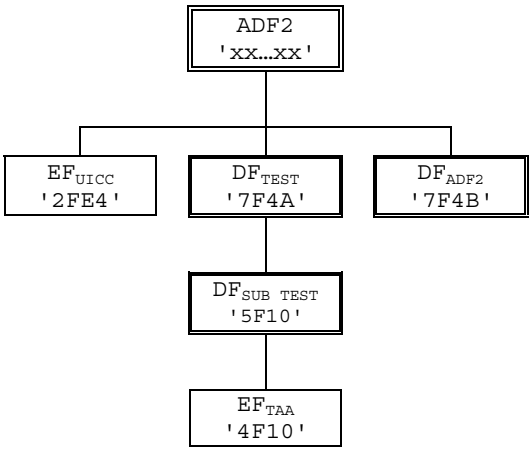


Figure B.3: File identifiers and directory structures under ADF2

Application PIN 1(Global PIN 1) shall be associated to ADF1.

Second Application PIN 1 (Local PIN 1) shall be associated to ADF1.

The value for Application PIN 1(Global PIN 1) shall be "0x31 0x31 0x31 0x31 0xFF 0xFF 0xFF 0xFF" and its state shall be 'disabled' during test applets execution.

The value for Second Application PIN 1(Local PIN 1) shall be "0x31 0x32 0x33 0x34 0xFF 0xFF 0xFF 0xFF" and its state shall be 'enabled' during test applets execution.

The value for ADM1 shall be "0x31 0x31 0x31 0x31 0x31 0x31 0x31 0x31" and its state shall be 'enabled' during test applets execution.

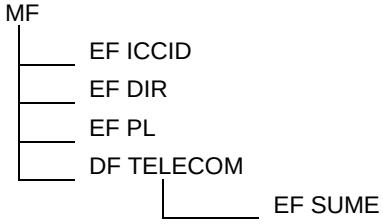
The value for ADM2 shall be "0x32 0x32 0x32 0x32 0x32 0x32 0x32 0x32" and its state shall be 'enabled' during test applets execution.

B.1 UICC file system server mandatory pre-personalization

Table B.1 shows the mandatory pre-personalization, the file system and the files' content, that the test UICC cards shall contain.

Table B.1

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 TS 102 221 [5]
EF_{dir}	2F00	Record 1: 61 12 4F 10 AID_1 Record 2: 61 12 4F 10 AID_2	
EF_{PL}	2F05	01 FF	
EF_{sume}	6F54	85 09 55 49 43 43 20 54 45 53 54 FF FF FF FF FF FF	Under DF Telecom, compliant with TS 102 222 [7]



B.2 UICC file system server test default pre-personalization

Figure B.1 shows the additional pre-personalization, the file system and files' content under the Master File, that the test UICC card shall contain.

B.2.1 EF_{UICC} (UICC Test EF)

Identifier: '2FE4'	Structure: transparent	Mandatory
File size: 3 bytes	Update activity: low	
Access Conditions: READ ALWAYS UPDATE ALWAYS DEACTIVATE ALWAYS ACTIVATE ALWAYS		

Bytes	Description	Default Value	M/O	Length
1 to 3	Test Data	AA AA AA	M	3 bytes

B.2.2 EF_{ARR} (UICC Test Access Rules EF)

Identifier: '2F06'	Structure: Linear fixed	Optional
SFI : none		
Record length : n bytes	Update activity: low	
Access Conditions: READ ALWAYS UPDATE ALWAYS DEACTIVATE ALWAYS ACTIVATE ALWAYS		

Bytes	Description	Default Value	M/O	Length
1 to n	Access Rule 1	Always for all commands	M	n bytes
n+1 to 2n	Access Rule 2	Never for all commands	M	n bytes
2n+1 to 3n	Access Rule 3	Global PIN1 & ADM1 for all commands	M	n bytes
3n+1 to 4n	Access Rule 4	Global PIN1 | ADM1 for all commands	M	n bytes
4n+1 to 5n	Access Rule 5	ADM2 for all commands	M	n bytes

Access rules definition:

Record nb	Delete selft	Terminat card Usage/ Terminate DF	Activate File	Deactivate File	Create File (DF creation)	Create File (EF creation)	Delete File (child)	
	Delete File	Terminate EF	Activate File	Deactivate File	Write Binary, Write Record, Append Record	Update Binary, Update Record, Erase Binary	Read binary, Read Record, Search Binary, Search Record	Increase, resize
1	Always	Always	Always	Always	Always	Always	Always	Always
2	Never	Never	Never	Never	Never	Never	Never	Never
3	Global PIN1 & ADM1	Global PIN1 & ADM1	Global PIN1 & ADM1	Global PIN1 & ADM1	Global PIN1 & ADM1	Global PIN1 & ADM1	Global PIN1 & ADM1	Global PIN1 & ADM1
4	Global PIN1 | ADM1	Global PIN1 | ADM1	Global PIN1 | ADM1	Global PIN1 | ADM1	Global PIN1 | ADM1	Global PIN1 | ADM1	Global PIN1 | ADM1	Global PIN1 | ADM1
5	ADM2	ADM2	ADM2	ADM2	ADM2	ADM2	ADM2	ADM2

B.2.3 DF_{TEST} (UICC Access Tests DF)

Identifier: '7F4A'

MF
└── DF TEST

B.2.3.1 EF_{TNR} (Transparent Never Read)

MF
└── DF TEST
    └── EF TNR

Identifier: '6F01'		Structure: transparent		Mandatory
SFI: '01'				
File size: 3 bytes		Update activity: low		
Access Conditions: READ NEVER UPDATE ALWAYS DEACTIVATE ALWAYS ACTIVATE ALWAYS				
Bytes	Description	Default Value	M/O	Length
1 to 3	Test Data	AA AA AA	M	3 bytes

B.2.3.2 EF_{TNU} (Transparent Never Update)

MF
└ DF TEST
└ EF TNU

Identifier: '6F02'	Structure: transparent	Mandatory
SFI: '02'		
File size: 3 bytes	Update activity: low	
Access Conditions: READ ALWAYS UPDATE NEVER DEACTIVATE ALWAYS ACTIVATE ALWAYS		

Bytes	Description	Default Value	M/O	Length
1 to 3	Test Data	55 55 55	M	3 bytes

B.2.3.3 EF_{TARU} (Transparent Always Read and Update)

MF
└ DF TEST
└ EF TARU

Identifier: '6F03'	Structure: transparent	Mandatory
SFI: '03'		
File size: 260 bytes	Update activity: low	
Access Conditions: READ ALWAYS UPDATE ALWAYS DEACTIVATE ALWAYS ACTIVATE ALWAYS		

Bytes	Description	Default Value	M/O	Length
1 to 260	Test Data	FF ... FF	M	260 bytes

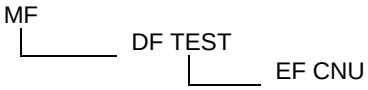
B.2.3.4 EF_{CNR} (Cyclic Never Read)

MF
└ DF TEST
└ EF CNR

Identifier: '6F04'	Structure: cyclic	Mandatory
SFI: '04'		
Record length: 3 bytes	Update activity: high	
Access Conditions: READ NEVER UPDATE ALWAYS INCREASE ALWAYS DEACTIVATE ALWAYS ACTIVATE 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

B.2.3.5 EF_{CNU} (Cyclic Never Update)



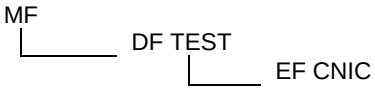
Identifier: '6F05'	Structure: cyclic	Mandatory
SFI: '05'		
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	NEVER
INCREASE	NEVER
DEACTIVATE	ALWAYS
ACTIVATE	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

B.2.3.6 EF_{CNIC} (Cyclic Never Increase)



Identifier: '6F1C'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	NEVER
DEACTIVATE	ALWAYS
ACTIVATE	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

B.2.3.7 EF_{CNDE} (Cyclic Never Deactivate)

MF
└ DF TEST
└ EF CNDE

Identifier: '6F07	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
DEACTIVATE	NEVER
ACTIVATE	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

B.2.3.8 EF_{CNAC} (Cyclic Never Activate)

MF
└ DF TEST
└ EF CNAC

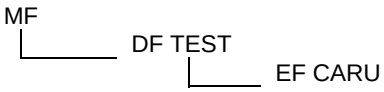
Identifier: '6F08'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
DEACTIVATE	ALWAYS
ACTIVATE	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

B.2.3.9 EF_{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
DEACTIVATE	ALWAYS
ACTIVATE	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

B.2.3.10 EF_{LNR} (Linear Fixed Never Read)

Identifier: '6F0A'	Structure: linear fixed	Mandatory
Record length: 4 bytes	Update activity: low	

Access Conditions:

READ	NEVER
UPDATE	ALWAYS
DEACTIVATE	ALWAYS
ACTIVATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data - Record 1	FF FF FF FF	M	4 bytes
2	Test Data - Record 2	FF FF FF FF	M	4 bytes

B.2.3.11 EF_{LNU} (Linear Fixed Never Update)

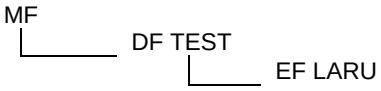
Identifier: '6F0B'	Structure: linear fixed	Mandatory
Record length: 4 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	NEVER
DEACTIVATE	ALWAYS
ACTIVATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data - Record 1	FF FF FF FF	M	4 bytes
2	Test Data - Record 2	FF FF FF FF	M	4 bytes

B.2.3.12 EF_{LARU} (Linear Fixed Always Read and Update)



Identifier: '6F0C'	Structure: linear fixed	Mandatory
Record length: 4 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
DEACTIVATE	ALWAYS
ACTIVATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data - Record 1	55 55 55 55	M	4 bytes
2	Test Data - Record 2	AA AA AA AA	M	4 bytes

B.2.3.13 EF_{CINA} (Cyclic Increase Not Allowed)

MF
└ DF TEST
└ EF CINA

Identifier: '6F0D'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	NEVER
DEACTIVATE	ALWAYS
ACTIVATE	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

B.2.3.14 EF_{TRAC} (Transparent Read Access Condition ADM 2)

MF
└ DF TEST
└ EF TRAC

Identifier: '6F0E'	Structure: transparent	Mandatory
File size: 3 bytes	Update activity: low	

Access Conditions:

READ	ADM 2
UPDATE	ALWAYS
DEACTIVATE	ALWAYS
ACTIVATE	ALWAYS

Bytes	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes

B.2.3.15 EF_{TDAC} (Transparent Deactivate Access Condition Application PIN 1)

MF
└ DF TEST
└ EF TDAC

Identifier: '6F0F'	Structure: transparent	Mandatory
File size: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
DEACTIVATE	Application PIN 1
ACTIVATE	ALWAYS

Bytes	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes

B.2.3.16 EF_{CIAC} (Cyclic Increase Access Condition ADM 2)

MF

MF

└── DF TEST

└── EF CIAC

Identifier: '6F10'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ADM 2
DEACTIVATE	ALWAYS
ACTIVATE	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

B.2.3.17 EF_{CIAA} (Cyclic Increase Access Condition ADM 1)

MF

└── DF TEST

└── EF CIAA

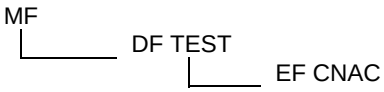
Identifier: '6F11'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ADM 1
DEACTIVATE	ALWAYS
ACTIVATE	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

B.2.3.18 EF_{CNRA} (Cyclic Never Activate)



Identifier: '6F12'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

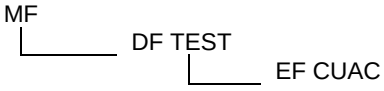
Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
DEACTIVATE	ALWAYS
ACTIVATE	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 Operational state bit of Life Cycle Status Integer shall be DEACTIVATED as defined in TS 102 221 [5].

B.2.3.19 EF_{CUAC} (Cyclic Update Access Condition Application PIN 1)



Identifier: '6F13'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	Application PIN 1
UPDATE	Application PIN 1
INCREASE	Application PIN 1
DEACTIVATE	ADM 1
ACTIVATE	ADM 1

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
3	Test Data	00 00 00	M	3 bytes

B.2.3.20 EF_{TAAC} (Transparent Activate Access Condition Application PIN 1)

MF
└ DF TEST
└ EF TAAC

Identifier: '6F14'	Structure: transparent	Mandatory
File size: 3 bytes	Update activity: high	
Access Conditions: READ Application PIN 1 UPDATE ADM 1 DEACTIVATE ADM 1 ACTIVATE Application PIN 1		

Bytes	Description	Default Value	M/O	Length
1	Test Data	FF FF FF	M	3 bytes

B.2.3.21 EF_{LADA} (Linear Fixed Activate Deactivate Access Condition ADM 2)

MF
└ DF TEST
└ EF LADA

Identifier: '6F15'	Structure: linear fixed	Mandatory
Record length: 3 bytes	Update activity: high	
Access Conditions: READ Application PIN 1 UPDATE Application PIN 1 DEACTIVATE ADM 2 ACTIVATE ADM 2		

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	FF FF FF	M	3 bytes

B.2.3.22 EF_{TAAA} (Transparent All Access Conditions ADM 1)

MF
└ DF TEST
└ EF TAAA

Identifier: '6F16'	Structure: transparent	Mandatory
File size: 3 bytes	Update activity: high	
Access Conditions: READ ADM 1 UPDATE ADM 1 DEACTIVATE ADM 1 ACTIVATE ADM 1		

Bytes	Description	Default Value	M/O	Length
1	Test Data	FF FF FF	M	3 bytes

B.2.3.23 EF_{LRUA} (Linear Fixed Read Update Access Condition Application PIN 1)

MF
└ DF TEST
└ EF LRUA

Identifier: '6F17'	Structure: linear fixed	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	Application PIN 1
UPDATE	Application PIN 1
DEACTIVATE	ADM 1
ACTIVATE	ADM 1

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	FF FF FF	M	3 bytes

B.2.3.24 EF_{LUPC} (Linear Fixed Update Access Condition ADM 2)

MF
└ DF TEST
└ EF LUPC

Identifier: '6F18'	Structure: linear fixed	Mandatory
Record length: 10 bytes	Update activity: high	

Access Conditions:

READ	Application PIN 1
UPDATE	ADM 2
DEACTIVATE	ADM 1
ACTIVATE	ADM 1

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 … 11	M	10 bytes
2	Test Data	22 … 22	M	10 bytes

B.2.3.25 EF_{NOSH} (Not Shareable)

MF
└ DF TEST
└ EF NOSH

Identifier: '6F19'	Structure: transparent	Mandatory
File size: 3 bytes	Update activity: high	

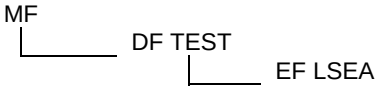
Access Conditions:

NOT SHAREABLE

READ	ALWAYS
UPDATE	ALWAYS
DEACTIVATE	ALWAYS
ACTIVATE	ALWAYS

Bytes	Description	Default Value	M/O	Length
1	Test Data	FF FF FF	M	3 bytes

B.2.3.26 EF_{LSEA} (Linear File for SearchRecord tests)



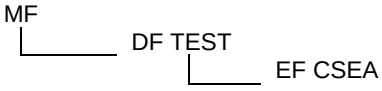
Identifier: '6F1A'	Structure: Linear Fixed	Mandatory
Record length: 15 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
DEACTIVATE	ALWAYS
ACTIVATE	ALWAYS

Bytes	Description	Default Value	M/O	Length
1	Test Data	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	M	15 bytes
2	Test Data	0A 0B 0C 0D 0E 0F 01 02 03 04 05 06 07 08 09	M	15 bytes
3	Test Data	04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 01 02 03	M	15 bytes
4	Test Data	07 08 09 0A 0B 0C 0D 0E 0F 01 02 03 04 05 06	M	15 bytes
5	Test Data	0A 01 02 03 0B 01 02 03 0C 01 02 03 0D 01 02	M	15 bytes
6	Test Data	03 02 01 03 02 01 03 02 01 03 02 01 03 02 01	M	15 bytes

B.2.3.27 EF_{CSEA} (Cyclic File for SearchRecord tests)



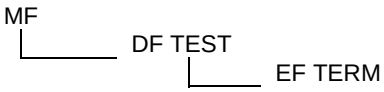
Identifier: '6F1B'	Structure: Cyclic	Mandatory
Record length: 15 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
DEACTIVATE	ALWAYS
ACTIVATE	ALWAYS

Bytes	Description	Default Value	M/O	Length
1	Test Data	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	M	15 bytes
2	Test Data	0A 0B 0C 0D 0E 0F 01 02 03 04 05 06 07 08 09	M	15 bytes
3	Test Data	04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 01 02 03	M	15 bytes
4	Test Data	07 08 09 0A 0B 0C 0D 0E 0F 01 02 03 04 05 06	M	15 bytes
5	Test Data	0A 01 02 03 0B 01 02 03 0C 01 02 03 0D 01 02	M	15 bytes
6	Test Data	03 02 01 03 02 01 03 02 01 03 02 01 03 02 01	M	15 bytes

B.2.3.28 EF_{TERM} (Terminated)



Identifier: '6F30'	Structure: Transparent	Mandatory
File size: 3 bytes	Update activity: low	
LCSI: TERMINATED Access Conditions: READ ALWAYS UPDATE ALWAYS DEACTIVATE ALWAYS ACTIVATE ALWAYS		

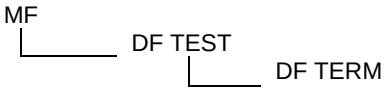
Bytes	Description	Default Value	M/O	Length
1	Test Data	FF FF FF	M	3 bytes

B.2.3.29 DF_{TERM} (DF Terminated)

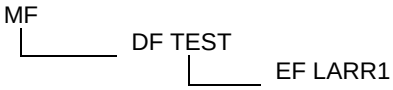
Identifier: '5F30'

Access Conditions: Record 1 of EFARR (0x2F06 under MF and 0x6F06 under ADF1)

Life Cycle Status Information: Terminated



B.2.3.30 EF_{LARR1} (Linear Fixed on Access Rule Reference 1)



Identifier: '6FA1'	Structure: linear fixed	Mandatory
Record length: 3 bytes	Update activity: high	
Access Conditions: Record 1 of EF_{ARR}		

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11	M	3 bytes
2	Test Data	22 22 22	M	3 bytes

B.2.3.31 EF_{LARR2} (Linear Fixed on Access Rule Reference 2)

MF
└ DF TEST
└ EF LARR2

Identifier: '6FA2'	Structure: linear fixed		Mandatory	
Record length: 3 bytes		Update activity: high		
Access Conditions: Record 2 of EF_{ARR}				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11	M	3 bytes
2	Test Data	22 22 22	M	3 bytes

B.2.3.32 EF_{LARR3} (Linear Fixed on Access Rule Reference 3)

MF
└ DF TEST
└ EF LARR3

Identifier: '6FA3'	Structure: linear fixed		Mandatory	
Record length: 3 bytes		Update activity: high		
Access Conditions: Record 3 of EF_{ARR}				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11	M	3 bytes
2	Test Data	22 22 22	M	3 bytes

B.2.3.33 EF_{LARR4} (Linear Fixed on Access Rule Reference 4)

MF
└ DF TEST
└ EF LARR4

Identifier: '6FA4'	Structure: linear fixed		Mandatory	
Record length: 3 bytes		Update activity: high		
Access Conditions: Record 4 of EF_{ARR}				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11	M	3 bytes
2	Test Data	22 22 22	M	3 bytes

B.2.3.34 EF_{LARR5} (Linear Fixed on Access Rule Reference 5)

MF
└ DF TEST
└ EF LARR5

Identifier: '6FA5'	Structure: linear fixed		Mandatory	
Record length: 3 bytes		Update activity: high		
Access Conditions: Record 5 of EF_{ARR}				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11	M	3 bytes
2	Test Data	22 22 22	M	3 bytes

B.2.3.35 EF_{TARR1} (Transparent on Access Rule Reference 1)

MF
└ DF TEST
└ EF TARR1

Identifier: '6FB1'	Structure: transparent		Mandatory	
File size: 5 bytes		Update activity: high		
Access Conditions: Record 1 of EF_{ARR}				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11 11 11	M	5 bytes

B.2.3.36 EF_{TARR2} (Transparent on Access Rule Reference 2)

MF
└ DF TEST
└ EF TARR2

Identifier: '6FB2'	Structure: transparent		Mandatory	
File size: 5 bytes		Update activity: high		
Access Conditions: Record 2 of EF_{ARR}				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11 11 11	M	5 bytes

B.2.3.37 EF_{TARR3} (Transparent on Access Rule Reference 3)

MF
└ DF TEST
└ EF TARR3

Identifier: '6FB3'	Structure: transparent		Mandatory	
File size: 5 bytes		Update activity: high		
Access Conditions: Record 3 of EF_{ARR}				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11 11 11	M	5 bytes

B.2.3.38 EF_{TARR4} (Transparent on Access Rule Reference 4)

MF
└ DF TEST
└ EF TARR4

Identifier: '6FB4'	Structure: transparent		Mandatory	
File size: 5 bytes		Update activity: high		
Access Conditions: Record 4 of EF_{ARR}				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11 11 11	M	5 bytes

B.2.3.39 EF_{TARR5} (Transparent on Access Rule Reference 5)

MF
└ DF TEST
└ EF TARR5

Identifier: '6FB5'	Structure: transparent		Mandatory	
File size: 5 bytes		Update activity: high		
Access Conditions: Record 5 of EF_{ARR}				
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11 11 11	M	5 bytes

B.2.3.40 EF_{CARR1} (Cyclic Access Rule Reference 1)

MF
└ DF TEST
└ EF CARR1

Identifier: '6FC1'	Structure: cyclic		Mandatory
Record length: 3 bytes		Update activity: high	
Access Conditions: Record 1 of EF_{ARR}			
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11	M	3 bytes
2	Test Data	22 22 22	M	3 bytes

B.2.3.41 EF_{CARR2} (Cyclic Access Rule Reference 2)

MF
└ DF TEST
└ EF CARR2

Identifier: '6FC2'	Structure: cyclic		Mandatory
Record length: 3 bytes		Update activity: high	
Access Conditions: Record 2 of EF_{ARR}			
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11	M	3 bytes
2	Test Data	22 22 22	M	3 bytes

B.2.3.42 EF_{CARR3} (Cyclic Access Rule Reference 3)

MF
└ DF TEST
└ EF CARR3

Identifier: '6FC3'	Structure: cyclic		Mandatory
Record length: 3 bytes		Update activity: high	
Access Conditions: Record 3 of EF_{ARR}			
Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11	M	3 bytes
2	Test Data	22 22 22	M	3 bytes

B.2.3.43 EF_{CARR4} (Cyclic Access Rule Reference 4)

MF
└ DF TEST
└ EF CARR4

Identifier: '6FC4'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	
Access Conditions: Record 4 of EF_{ARR}		

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11	M	3 bytes
2	Test Data	22 22 22	M	3 bytes

B.2.3.44 EF_{CARR5} (Cyclic Access Rule Reference 5)

MF
└ DF TEST
└ EF CARR5

Identifier: '6FC5'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	
Access Conditions: Record 5 of EF_{ARR}		

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	11 11 11	M	3 bytes
2	Test Data	22 22 22	M	3 bytes

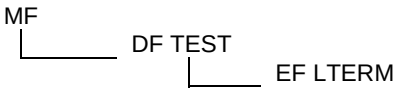
B.2.3.45 EF_{NOSH2} (Not Shareable)

MF
└ DF TEST
└ EF NOSH2

Identifier: '6FC6'	Structure: transparent	Mandatory
SFI: '06'		
File size: 3 bytes	Update activity: high	
Access Conditions: READ ALWAYS UPDATE ALWAYS DEACTIVATE ALWAYS ACTIVATE ALWAYS		

Bytes	Description	Default Value	M/O	Length
1 to 3	Test Data	FF FF FF	M	3 bytes

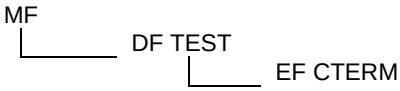
B.2.3.46 EF_{LTERM} (Linear File Terminated)



Identifier: '6FC7'	Structure: Linear Fixed	Mandatory
Record length: 15 bytes	Update activity: high	
LCSI: TERMINATED Access Conditions: READ ALWAYS UPDATE ALWAYS DEACTIVATE ALWAYS ACTIVATE ALWAYS		

Bytes	Description	Default Value	M/O	Length
1	Test Data	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	M	15 bytes

B.2.3.47 EF_{CTERM} (Cyclic File Terminated)

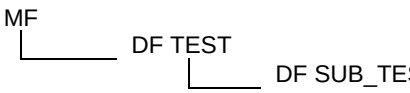


Identifier: '6FC8'	Structure: Cyclic	Mandatory
Record length: 15 bytes	Update activity: high	
LCSI: TERMINATED Access Conditions: READ ALWAYS UPDATE ALWAYS DEACTIVATE ALWAYS ACTIVATE ALWAYS		

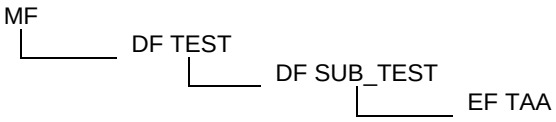
Bytes	Description	Default Value	M/O	Length
1	Test Data	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	M	15 bytes

B.2.4 DF_{SUB_TEST} (Test DF under DF TEST)

Identifier: '5F10'



B.2.4.1 EF_{TAA} (Test EF)

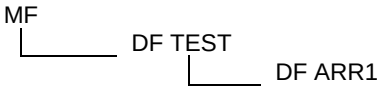


Identifier: '4F10'		Structure: transparent		Mandatory
File size: 3 bytes		Update activity: low		
Access Conditions: READ ALWAYS UPDATE ALWAYS DEACTIVATE ALWAYS ACTIVATE ALWAYS				
Bytes	Description	Default Value	M/O	Length
1 to 3	Test Data	AA AA AA	M	3 bytes

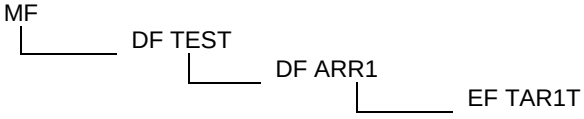
B.2.5 DF_{ARR1} (DF Access Rule Reference 1)

Identifier: '5F11'

Access Conditions: Record 1 of EFARR (0x2F06 under the MF and 0x6F06 under ADF1).



B.2.5.1 EF_{TAR1T} (Transparent Access Rule 1 Test EF)

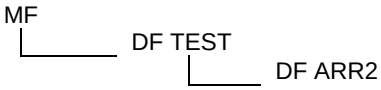


Identifier: '4F11'		Structure: transparent		Mandatory
File size: 3 bytes		Update activity: low		
Access Conditions: Record 1 of EF_{ARR}				
Bytes	Description	Default Value	M/O	Length
1 to 3	Test Data	AA AA AA	M	3 bytes

B.2.6 DF_{ARR2} (DF Access Rule Reference 2)

Identifier: '5F12'

Access Conditions: Record 2 of EFARR (0x2F06 under the MF and 0x6F06 under ADF1).



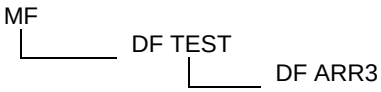
B.2.6.1 EF_{TAR2T} (Transparent Access Rule 2 Test EF)

Identifier: '4F12'		Structure: transparent		Mandatory
File size: 3 bytes		Update activity: low		
Access Conditions: Record 2 of EF_{ARR}				
Bytes	Description	Default Value	M/O	Length
1 to 3	Test Data	AA AA AA	M	3 bytes

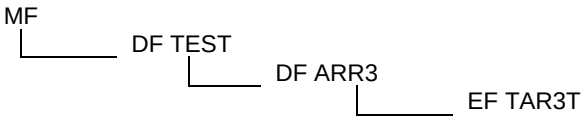
B.2.7 DF_{ARR3} (DF Access Rule Reference 3)

Identifier: '5F13'

Access Conditions: Record 3 of EFARR (0x2F06 under the MF and 0x6F06 under ADF1).



B.2.7.1 EF_{TAR3T} (Transparent Access Rule 3 Test EF)

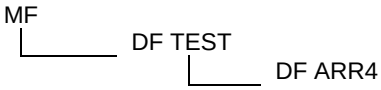


Identifier: '4F13'		Structure: transparent		Mandatory
File size: 3 bytes		Update activity: low		
Access Conditions: Record 3 of EF_{ARR}				
Bytes	Description	Default Value	M/O	Length
1 to 3	Test Data	AA AA AA	M	3 bytes

B.2.8 DF_{ARR4} (DF Access Rule Reference 4)

Identifier: '5F14'

Access Conditions: Record 4 of EFARR (0x2F06 under the MF and 0x6F06 under ADF1).



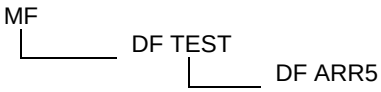
B.2.8.1 EF_{TAR4T} (Transparent Access Rule 4 Test EF)

Identifier: '4F14'		Structure: transparent		Mandatory
File size: 3 bytes			Update activity: low	
Access Conditions: Record 4 of EF_{ARR}				
Bytes	Description	Default Value	M/O	Length
1 to 3	Test Data	AA AA AA	M	3 bytes

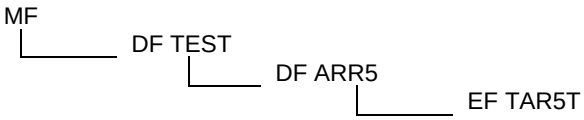
B.2.9 DF_{ARR5} (DF Access Rule Reference 5)

Identifier: '5F15'

Access Conditions: Record 5 of EFARR (0x2F06 under the MF and 0x6F06 under ADF1).



B.2.9.1 EF_{TAR5T} (Transparent Access Rule 5 Test EF)



Identifier: '4F15'		Structure: transparent		Mandatory
File size: 3 bytes			Update activity: low	
Access Conditions: Record 5 of EF_{ARR}				
Bytes	Description	Default Value	M/O	Length
1 to 3	Test Data	AA AA AA	M	3 bytes

B.3 First application dedicated files system ADF1

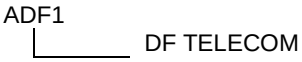
The prepersonalization of UICC cards shall contain ADF1.

Figure B.2 shows the additional prepersonalization, the file system and files' content under ADF1 system.

All the files are defined as in the UICC file system server.

B.3.1 $DF_{TELECOM}$

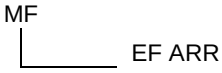
Identifier: '7F10'



B.3.4.1 EF_{SUME} (EF SetUpMenu)

Identifier: '6F54'		Structure: transparent		Mandatory
File size: 17 bytes			Update activity: low	
Access Conditions: READ ALWAYS UPDATE ALWAYS DEACTIVATE ALWAYS ACTIVATE ALWAYS				
Bytes	Description	Default Value	M/O	Length
1 to 17	Test Data	85 09 55 49 43 43 20 54 45 53 54 FF FF FF FF FF FF	M	17 bytes

B.3.4.2 EF_{ARR} (UICC Test Access Rules EF)



Identifier: '6F06'		Structure: Linear fixed		Optional
SFI : none				
Record length : n bytes			Update activity: low	
Access Conditions: READ ALWAYS UPDATE ALWAYS DEACTIVATE ALWAYS ACTIVATE ALWAYS				
Bytes	Description	Default Value	M/O	Length
1 to n	Access Rule 1	Always for all commands	M	n bytes
n+1 to 2n	Access Rule 2	Never for all commands	M	n bytes
2n+1 to 3n	Access Rule 3	Global PIN1 & ADM1 for all commands	M	n bytes
3n+1 to 4n	Access Rule 4	Global PIN1 | ADM1 for all commands	M	n bytes
4n+1 to 5n	Access Rule 5	Local PIN1 for all commands	M	n bytes

Access rules definition:

Record nb	Delete selft	Terminat card Usage/ Terminate DF	Activate File	Deactivate File	Create File (DF creation)	Create File (EF creation)	Delete File (child)	
	Delete File	Terminate EF	Activate File	Deactivate File	Write Binary, Write Record, Append Record	Update Binary, Update Record, Erase Binary	Read binary, Read Record, Search Binary, Search Record	Increase, resize
1	Always	Always	Always	Always	Always	Always	Always	Always
2	Never	Never	Never	Never	Never	Never	Never	Never
3	Global PIN1 & ADM1	Global PIN1 & ADM1	Global PIN1 & ADM1	Global PIN1 & ADM1	Global PIN1 & ADM1	Global PIN1 & ADM1	Global PIN1 & ADM1	Global PIN1 & ADM1
4	Global PIN1 | ADM1	Global PIN1 | ADM1	Global PIN1 | ADM1	Global PIN1 | ADM1	Global PIN1 | ADM1	Global PIN1 | ADM1	Global PIN1 | ADM1	Global PIN1 | ADM1
5	Local PIN1	Local PIN1	Local PIN1	Local PIN1	Local PIN1	Local PIN1	Local PIN1	Local PIN1

B.4 Second application dedicated files system ADF2

The prepersonalization of UICC cards shall contain ADF2.

Figure B.3 shows the additional prepersonalization, the file system and files' content under ADF2 system.

B.4.1 EF_{UICC} (UICC Test EF)

ADF2
└ EF UICC

Identifier: '2FE4'	Structure: transparent	Mandatory
File size: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
DEACTIVATE	ALWAYS
ACTIVATE	ALWAYS

Bytes	Description	Default Value	M/O	Length
1 to 3	Test Data	CC CC CC	M	3 bytes

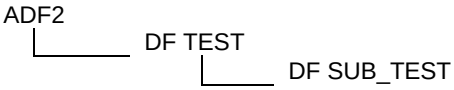
B.4.2 DF_{TEST} (1st Test DF under ADF2)

Identifier: '7F4A'

ADF2
└ DF TEST

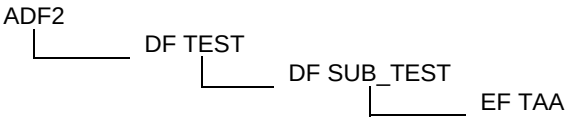
B.4.2.1 DF_{SUB_TEST} (1^{st} DF under DF_TEST)

Identifier: '5F10'



B.4.2.1.1 EF_{TAA} (Test EF)

Identifier: '4F10'		Structure: transparent		Mandatory
File size: 3 bytes		Update activity: low		
Access Conditions: READ ALWAYS UPDATE ALWAYS DEACTIVATE ALWAYS ACTIVATE ALWAYS				
Bytes	Description	Default Value	M/O	Length
1 to 3	Test Data	AA AA AA	M	3 bytes



B.4.3 DF_{ADF2} (2^{nd} Test DF under ADF2)

Identifier: '7F4B'

Annex C (normative): Test file description

Every test source is written in java™ and shall use methods defined in Annex D interfaces to communicate with the card, or to check status word or received data.

In order to be more readable, data specified as method string parameters shall be presented in 4 blocks of 4 bytes per line. Every block is separated by a space character. Every string line is appended to previous one and shall be aligned. An example is provided in Annex D.

Every test file shall start with a call to reset() method.

Except otherwise stated, each reset() method shall be followed by a profile download.

Annex D (normative): uicc.test.util package, Uicc interfaces and testing script example

See attached files contained in ts_102268v070000p0.zip:

- Annex_D_UiccTestUtil.zip
- Annex_D_UiccInterfaces.zip
- Annex_D_Example.zip

Annex E (normative): Test Area files

See attached file contained in ts_102268v070000p0.zip:

- Annex_E_SourceCode.zip

Annex F (informative): Bibliography

- ISO/IEC 7816-3 (2006): "Identification cards - Integrated circuit cards - Part 3: Cards with contacts - Electrical interface and transmission protocols".
- ETSI TS 102 225: "Smart Cards; Secured packet structure for UICC based applications (Release 6)".
- ETSI TS 102 240: "Smart Cards; UICC Application Programming Interface and Loader Requirements; Service description; (Release 6)".
- ETSI TS 151 011: "Digital cellular telecommunications system (Phase 2+); Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface (3GPP TS 51.011 Release 4)".

Annex G (informative): Change history

The table below indicates all changes that have been made to the present document since first publication.

Change history								
Date	**SCP #**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**Old**	**New**
2007-01	29	SCP-070014				Initial publication		6.0.0
2013-01	58	SCP(13)000019	005	-	F	Correction of the test case number 2 of the Test Area Reference: Cre_Reg_Evtr	6.0.0	6.1.0
2013-04	59	SCP(13)000064	004	-	F	Improvement of results reporting	6.0.0	6.1.0
2013-07	60	SCP(13)000124	006	-	F	CR 102 268 R6 #006: Correction of test coverage for test case 5.2.3.25	6.1.0	6.2.0
2013-10	61	SCP(13)000204	007	-	F	CR 102 268 R6 #007: Correction of test coverage for TC 5.5.9.3	6.1.0	6.2.0
2013-10	61	SCP(13)000203	010	-	F	CR 102 268 R6 #010: Addition of new test applet for TC 5.2.5.13	6.1.0	6.2.0
2014-02	62	SCP(14)000010	012	-	F	CR 102 268 R6 #012: Correction of test case 5.2.8.18 ID4/5/6 to correct MF to ADF1	6.1.0	6.2.0
2014-02	62	SCP(14)000012	008	1	F	CR 102 268 R6 #008r1: Modification of conformance requirement structure for TC 5.2.10.25	6.1.0	6.2.0
2014-02	62	SCP(14)000011	011	-	F	CR 102 268 R6 #011: Correction of test case 5.1.1.3	6.1.0	6.2.0
2014-06	64	SCP(14)000146	013	-	F	CR 102 268 R6 #013: Correction of restore functionality of EF_CARU content	6.1.0	6.2.0
2014-12	66	SCP(14)000293	015	-	F	CR 102 268 R6 #015: Adjust TC 5.2.10.25 to specification	6.1.0	6.2.0
2014-12	66	SCP(14)000295r1	017	1	F	CR 102 268 R6 #017r1: Step deleted from 5.2.16.5	6.1.0	6.2.0
2014-12	66	SCP(14)000296	018	-	F	CR 102 268 R6 #018: Bit setting for COMPREHENSION-TLV	6.1.0	6.2.0
2014-12	66	SCP(14)000292	014	-	F	CR 102 268 R6 #014: Correction of test Java file for TC 5.2.5.13	6.1.0	6.2.0
2014-12	66	SCP(14)000294	016	-	F	CR 102 268 R6 #016: Step added to TC 5.2.10.12	6.1.0	6.2.0
2015-02	67	SCP(15)000009	019	-	F	CR 102 268 R6 #019: Change in the Terminal Profile to avoid proactive command pending	6.1.0	6.2.0
2015-02	67	SCP(15)000010	020	-	F	CR 102 268 R6 #020: Missing SW check	6.1.0	6.2.0
2015-02	67	SCP(15)000011	021	-	F	CR 102 268 R6 #021: No exception due to large buffer	6.1.0	6.2.0
2012-12	57	SCP(12)000230r1	001	-	D	Alignment of core spec and test spec with respect to the method uicc.system.HandlerBuilder.buildTLVHandler()	6.2.0	7.0.0
2013-01	58	SCP(13)000017	002	-	F	Additional files for the test file system	6.2.0	7.0.0
2013-01	58	SCP(13)000018	003	-	C	Update of the reference section	6.2.0	7.0.0

History

Document history		
V7.0.0	April 2015	Publication