## 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 |  |
| 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) |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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. |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 havebytes 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 6 | value2Length < 0  appendTLV()  value2.length = 5  value2Offset = 0  value2Length = -1 | ArrayIndexOutOfBoundsException is thrown |  |
| 7 | Handler overflow exception  Call the appendArray() with length of getCapacity()-1  appendTLV()  value2.length = 256  value2Offset = 0  value2Length = 254 | ToolkitException.HANDLER\_OVERFLOW is thrown |  |
| 8 | Bad parameter exception  Clear the handlerappendTLV()  value2.length = 256  value2Offset = 0  value2Length = 256 | ToolkitException.BAD\_INPUT\_PARAMETER is thrown |  |
| 9 | clear the handler, append the handler with TLVs:  81 03 11 22 33  82 02 99 77 |  |  |
|  | Select Command Details TLV |  |  |
|  | Successful call  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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |  |
|  | 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 |  |
| 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" |
| 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" |
| 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 |
| 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 |
| 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() |  |
| 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 |  |
| 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 |  |
| 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. |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |  |
|  | 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 | result = 253 |  |
|  | Call copy() method |  |  |
|  | Compare the array  compareBuffer = 00 81 F7 03 04 … F9 84 01 00 | 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 |  |
| 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** | result = 253 |  |
|  | Call copy() method |  |  |
|  | Compare handler  compareBuffer = 04 81 FA 00 01…F9 | Result of javacard.framework.Util.arrayCompare() is 00h |  |

#### 5.2.4.21 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 |  |
| 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 |  |
| 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. |
| 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 |  |
| 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 | 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 |  |

#### 5.2.4.28 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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() | 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 |  |  |
|  | 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 … 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 |  |
| 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. |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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. |
| 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. |  |
| 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. |  |
| 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 |  | GET INPUT Proactive 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 |  |
| 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 |  | 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 getValueShort(7D) method | Result is 7Dh 7Eh |  |
| 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. |
| 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 EFSUME 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. |
| 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 |  |
| 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. |  |
| 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. |  |
| 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. |  |
| 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. |
| 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. |
| 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. |  |
| 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. |  |
| 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' |  |
| 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' |  |
| 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. |  |
| 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. |  |
| 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. |  |
| 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 |
| 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. |  |
| 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 |  |
| 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. |  |
| 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. |  |
| 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. |  |
| 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  2- Call registerFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  baFileList=”02 3F0011116F03 3F0011116F09” baADFAid=null  3- Call isEventSet (EVENT\_EXTERNAL\_FILE\_UPDATE) method  4- Update binary on MF\DFTEST\EFTARU  5- Increase on MF\DFTEST\EFCARU  6- Update record on MF\DFTEST\EFLARU  7- Update binary on MF\DFTEST\ DFSUB\_TEST\EFTAA  8- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  baFileList=”02 3F0011116F03 3F0011116F09”  baADFAid=null | 1- Returns false  2- No exception is thrown  3- Returns true  4- Applet is triggered  5- Applet is triggered  6- Applet is not triggered  7- Applet is not triggered |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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\DFTEST\EFTARU  2- Call registerFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  aFileView = ADF1View  3- Update binary on ADF1\DFTEST\EFTARU  4- Delete ADF1\DFTEST\EFTARU  5- Create ADF1\DFTEST\EFTARU  6- Update binary on ADF1\DFTEST\EFTARU  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 |  |
| 12 | **Register a deleted and recreated DF under ADF**  1- Applet selects ADF1\DFTEST\DFSUB\_TEST  2- Call registerFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  aFileView = ADF1View  3- Update binary on ADF1\DFTEST\ DFSUB\_TEST\EFTAA  4- Delete EFTAA, delete DFSUB\_TEST  5- Create DFSUB\_TEST, create EFTAA  6- Update binary on ADF1\DFTEST\ DFSUB\_TEST\EFTAA  7- Applet selects ADF1\DFTEST\ DFSUB\_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\DFTEST\EFTARU  4- Increase on MF\DFTEST\EFCARU  5- Update record on MF\DFTEST\EFLARU  6- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  baFileList=”02 3F0011116F03 3F0011116F09” baADFAid=null  7- Update binary on MF\DFTEST\EFTARU  8- Increase on MF\DFTEST\EFCARU  9- Update record on MF\DFTEST\EFLARU  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\DFTEST\EFLARU  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 |  |
| 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\DFTEST\EFTARU  3- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  baFileList=”01 3F001111”  baADFAid=null  4- Update binary on MF\DFTEST\EFTARU | 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\DFTEST\EFTARU  3- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  baFileList=”01 3F0011116F03”  baADFAid=null  4- Update binary on MF\DFTEST\EFTARU  5- Update record on MF\DFTEST\EFLARU | 2- Applet is triggered  3- No exception is thrown  4- Applet is triggered  5- Applet is triggered |  |
| 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\DFTEST\EFTARU  4- Increase on MF\DFTEST\EFCARU  5- Update record on MF\DFTEST\EFLARU  6- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  baFileList=”02 3F007FFF11116F03 3F007FFF11116F09”  baADFAid=”AID ADF1”  7- Update binary on MF\DFTEST\EFTARU  8- Increase on MF\DFTEST\EFCARU  9- Update record on MF\DFTEST\EFLARU  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\DFTEST\EFLARU  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 |  |
| 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\DFTEST\EFTARU  3- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  baFileList=”01 3F007FFF11116F03” baADFAid=”AID ADF1”  4- Update binary on MF\DFTEST\EFTARU  5- Update record on MF\DFTEST\EFLARU  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 |  |
| 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\DFTEST\EFTARU  4- Increase on MF\DFTEST\EFCARU  5- Update record on MF\DFTEST\EFLARU  6- Call UICCView=getTheUICCView()  7- Applet selects EFTARU.  8- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  aFileView = UICCView  9- Update binary on MF\DFTEST\EFTARU  10- Applet selects EFCARU.  11- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  aFileView = UICCView  12- Increase on MF\DFTEST\EFCARU  13- Call isEventSet (EVENT\_EXTERNAL\_FILE\_UPDATE) method  14- Applet selects EFLARU.  15- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  aFileView = UICCView  16- Update record on MF\DFTEST\EFLARU  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\DFTEST\EFTARU  3- Applet selects DFTEST.  4- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  aFileView = UICCView  5- Update binary on MF\DFTEST\EFTARU | 2- Applet is triggered  4- No exception is thrown  5- Applet is triggered |  |
| 3 | Deregister EF does not affect parent DF  1-Select DFTEST  2- Call registerFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  aFileView = UICCView  3- Update binary on MF\DFTEST\EFTARU  4- Applet selects EFTARU.  5- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  aFileView = UICCView  6- Update binary on MF\DFTEST\EFTARU  7- Update record on MF\DFTEST\EFLARU | 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\DFTEST\EFTARU  3- Increase on ADF1\DFTEST\EFCARU  4- Update record on ADF1\DFTEST\EFLARU  5- Call ADF1View=getTheFileView()  6- Applet selects EFTARU.  7- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  aFileView = ADF1View  8- Update binary on ADF1\DFTEST\EFTARU  9- Applet selects EFCARU.  10- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  aFileView = ADF1View  11- Increase on ADF1\DFTEST\EFCARU  12- Applet selects EFLARU.  13- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  aFileView = ADF1View  14- Update record on ADF1\DFTEST\EFLARU | 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 |  |
| 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\DFTEST\EFTARU  3- Applet selects DFTEST.  4- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  aFileView = ADF1View  5- Update binary on ADF1\DFTEST\EFTARU | 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\DFTEST\EFTARU  3- Applet selects EFTARU.  4- Call deregisterFileEvent() method with parameters:  event= EVENT\_EXTERNAL\_FILE\_UPDATE  aFileView = ADF1View  5- Update binary on ADF1\DFTEST\EFTARU  6- Update record on ADF1\DFTEST\EFLARU  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 |  |
| 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 BERTLVtag 0x01, data length 0x22  Call getSize() method | Returns 0x24 |  |
| 2 | Fill the handler with BERTLVtag 0x01, data length 0x7F |  |  |
|  | Call getSize() method | Returns 0x81 |  |
| 3 | Fill the handler with BERTLVtag 0x01, data length 0x80 |  |  |
|  | Call getSize() method | Returns 0x83 |  |
| 4 | Fill the handler with BERTLVtag 0x01, data length 0xFF |  |  |
|  | Call getSize() method | Returns 0x102 |  |
| 5 | Fill the handler with BERTLVtag 0x01, data length 0x100 |  |  |
|  | Call getSize() method | Returns 0x104 |  |
| 6 | Fill the handler with BERTLVtag 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 |  |
| 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. |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 | result = 253 |  |
|  | Call copy() method |  |  |
|  | Compare the array  compareBuffer = 00 81 F7 03 04 … F9 84 01 00 | 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 |  |
| 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** | result = 253 |  |
|  | Call copy() method |  |  |
|  | Compare handler  compareBuffer = 04 81 FA 00 01…F9 | Result of javacard.framework.Util.arrayCompare() is 00h |  |

#### 5.2.10.21 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |
| 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 |  |

#### 5.2.10.25 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 |  |
| 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 |  |
| 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() | Result is 10h |  |
|  | Clear the handler |  |  |
| 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 | Result is 00 |  |
|  | Clear the handler |  |  |
| 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. |  |
| 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 |  |
| 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 |  |
| 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.