## 5.3 Package uicc.access.fileadministration

### 5.3.1 Interface AdminFileView

#### 5.3.1.1 Method createFile(ViewHandler viewHandler)

Test Area Reference: Api\_4\_Afv\_Crtf.

##### 5.3.1.1.1 Conformance requirement

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

public void createFile(ViewHandler viewHandler)

throws java.lang.NullPointerException,

UICCException,

AdminException,

javacard.framework.TransactionException

5.3.1.1.1.1 Normal execution

* CRRN1: This method creates a new file under the current DF or ADF, as described in ETSI TS 102 222 [7].

5.3.1.1.1.2 Parameter errors

* CRRP1: If viewHandler is null, an instance of java.lang.NullPointerException shall be thrown.
* CRRP2: If the viewHandler parameter includes incorrect parameters, an instance of AdminException shall be thrown. The reason code shall be AdminException.INCORRECT\_PARAMETERS.

5.3.1.1.1.3 Context errors

* CRRC1: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY\_PROBLEM.
* CRRC2: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL\_ERROR.
* CRRC3: If the file identifier of the EF being created already exists, an instance of AdminException shall be thrown. The reason code shall be AdminException.FILE\_ALREADY\_EXISTS.
* CRRC4: If the DF name already exists, an instance of AdminException shall be thrown. The reason code shall be AdminException.DF\_NAME\_ALREADY\_EXISTS.
* CRRC5: If there is not enough memory, an instance of AdminException shall be thrown. The reason code shall be AdminException.NOT\_ENOUGH\_MEMORY\_SPACE.
* CRRC6: If the access conditions are not satisfied, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY\_STATUS\_NOT\_SATISFIED.

##### 5.3.1.1.2 Test area files

Test Source: Test\_Api\_4\_Afv\_ Crtf.java.

Test Applet: Api\_4\_Afv\_ Crtf \_1.java.

Cap File: api\_4\_Afv\_ Crtf.cap.

##### 5.3.1.1.3 Test coverage

|  |  |
| --- | --- |
| CRR number | Test case number |
| N1 | 1, 2 |
| P1 | 3 |
| P2 | 4 |
| C1 | Not testable |
| C2 | Not testable |
| C3 | 5 |
| C4 | 6 |
| C5 | Not testable |
| C6 | 7 |

##### 5.3.1.1.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 1 | Create an EF  1- applet is triggered by sending an unrecognized envelope  2- get an AdminFileView AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR\_ON\_RESET)  3- select MF/DFTEST  4- create transparent EFRFU0 (6F29)  5- select EFRFU0, update binary 12 34 56  6- create Linear fixed EFRFU1  7- select EFRFU1, update record 01  8- create Cyclic EFRFU2  9- select EFRFU2, update record 01  record:1  value: 12 34 56  Applet finalizes  10- select MF/DFTEST/EFRFU0, read binary  11- select MF/DFTEST/EFRFU1, read record 1  12- select MF/DFTEST/EFRFU2 ,read record 1 | 4- no exception shall be thrown | 10- returns: 12 34 56  11- returns: 02  12- returns: 12 34 56 |
| 2 | Create a DF in ADF1  1- applet is triggered by sending an unrecognized envelope  2- get an AdminFileView AdminFileViewBuilder.getTheAdminFileView(AID\_ADF1,CLEAR\_ON\_RESET)  3- select ADF1/DFTEST  4- create DFRFU1 (5F01)  5- select DFRFU1  6- create EFRFU1 (6F2A)  7- select EFRFU1, update binary 12 34 56  Applet finalizes  8- select ADF1/DFTEST/EFRFU1 (6F2A), read binary  9- Reset |  | 8- returns: 12 34 56 |
| 3 | Call createFile with a null viewHandler  1- call createFile() with null. | 1- java.lang.NullPointerException shall be thrown |  |
| 4 | Call createFile with incorrect parameters  1- call createFile with incorrect parameters. | 1- AdminException.INCORRECT\_PARAMETERS shall be thrown |  |
| 5 | EF already exists  1- Select MF/DFTEST  2- Call createFile(EFTARU) | 2- AdminException.FILE\_ALREADY\_EXISTS |  |
| 6 | DF already exists  1- Call createFile(DFTEST) | 1- AdminException.DF\_NAME\_ALREADY\_EXISTS |  |
| 7 | Security status not satisfied  1- Select MF/DFTEST/DFARR2  2- Call createFile() to create some transparent file. | 2-UICCException.SECURITY\_STATUS\_NOT\_SATISFIED |  |

#### 5.3.1.2 Method deleteFile(short fid)

Test Area Reference: Api\_4\_Afv\_Dltf.

##### 5.3.1.2.1 Conformance requirement

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

public void deleteFile(short fid)

throws UICCException,

javacard.framework.TransactionException

5.3.1.2.1.1 Normal execution

* CRRN1: This method initiates the deletion of an EF immediately under the current DF, or a DF with its complete subtree, as described in ETSI TS 102 222 [7].

5.3.1.2.1.2 Parameter errors

Not applicable

5.3.1.2.1.3 Context errors

* CRRC1: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY\_PROBLEM.
* CRRC2: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL\_ERROR.
* CRRC3: If the file cannot be found in the current directory, an instance of UICCException. shall be thrown. The reason code shall be UICCException. FILE\_NOT\_FOUND.
* CRRC4: If the access conditions are not satisfied, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY\_STATUS\_NOT\_SATISFIED.
* CRRC5: If the operation would cause the commit capacity to be exceeded, an instance of javacard.framework.TransactionException shall be thrown.

##### 5.3.1.2.2 Test area files

Test Source: Test\_Api\_4\_Afv\_ Dltf.java.

Test Applet: Api\_4\_Afv\_ Dltf \_1.java.

Cap File: api\_4\_Afv\_ Dltf.cap.

##### 5.3.1.2.3 Test coverage

|  |  |
| --- | --- |
| CRR number | Test case number |
| N1 | 1, 2, 3, 4 |
| C1 | Not testable |
| C2 | Not testable |
| C3 | 5 |
| C4 | 6 |
| C5 | Not testable |

##### 5.3.1.2.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 0 | Initialization  1- applet is triggered by sending an unrecognized envelope  2- get an AdminFileView AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR\_ON\_RESET)  3- select MF/DFTEST  4- create DFRFU1  5- select DFRFU1  6- create DFRFU2 , create EFRFU1, create EFRFU2  7- select DFRFU2  8- create EFRFU1  9- get an AdminFileView AdminFileViewBuilder.getTheAdminFileView(AID\_ADF1,CLEAR\_ON\_RESET)  10- select MF/DFTEST  11- create DFRFU1  12- select DFRFU1  13- create DFRFU2 , create EFRFU1, create EFRFU2  14- select DFRFU2  15- create EFRFU1 |  |  |
| 1 | Delete EF  1- Select MF/DFTEST/DFRFU1  2- call deleteFile(EFRFU1)  3- Select MF/DFTEST/DFRFU1/EFRFU1 | 2- no exception shall be thrown  3- UICCException.FILE\_NOT\_FOUND is thrown |  |
| 2 | Delete EF in ADF1  1- Select ADF1/DFTEST/DFRFU1  2- call deleteFile(EFRFU1)  3- Select ADF1/DFTEST/DFRFU1/EFRFU1 | 2- no exception shall be thrown  3- UICCException.FILE\_NOT\_FOUND is thrown |  |
| 3 | Delete DF and its subtree  1- Select MF/DFTEST  2- call deleteFile(DFRFU1)  3- Select MF/DFTEST/DFRFU1 | 2- no exception shall be thrown  3- UICCException.FILE\_NOT\_FOUND is thrown |  |
| 4 | Delete DF and its subtree in ADF1  1- Select ADF1/DFTEST  2- call deleteFile(DFRFU1)  3- Select ADF1/DFTEST/DFRFU1 | 2- no exception shall be thrown  3- UICCException.FILE\_NOT\_FOUND is thrown |  |
| 5 | File not found  1- Select MF/DFTEST  2- call deleteFile(DFRFU1)  3- Select ADF1/DFTEST  4- call deleteFile(EFRFU1) | 2- UICCException.FILE\_NOT\_FOUND  4- UICCException.FILE\_NOT\_FOUND |  |
| 6 | Security status not satisfied  1- Select MF/DFTEST/DFARR2  2- call deleteFile(EFTAR2T) | 2- UICCException.SECURITY\_STATUS\_NOT\_SATISFIED |  |

#### 5.3.1.3 Method resizeFile(ViewHandler viewHandler)

Test Area Reference: Api\_4\_Afv\_Rszf.

##### 5.3.1.3.1 Conformance requirement

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

public void resizeFile(ViewHandler viewHandler)

throws java.lang.NullPointerException,

UICCException,

AdminException,

javacard.framework.TransactionException

5.3.1.3.1.1 Normal execution

* CRRN1: This method resizes a file under the current DF or ADF, as described in ETSI TS 102 222 [7].

5.3.1.3.1.2 Parameter errors

* CRRP1: If viewHandler is null, an instance of java.lang.NullPointerException shall be thrown.
* CRRP2: If the viewHandler parameter includes incorrect parameters, an instance of AdminException shall be thrown. The reason code shall be AdminException.INCORRECT\_PARAMETERS.

5.3.1.3.1.3 Context errors

* CRRC1: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY\_PROBLEM.
* CRRC2: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL\_ERROR.
* CRRC3: If the file cannot be found in the current directory, an instance of UICCException. shall be thrown. The reason code shall be UICCException. FILE\_NOT\_FOUND.
* CRRC4: If the access conditions are not satisfied, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY\_STATUS\_NOT\_SATISFIED.
* CRRC5: If the operation would cause the commit capacity to be exceeded, an instance of javacard.framework.TransactionException shall be thrown.
* CRRC6: If there is not enough memory, an instance of AdminException shall be thrown. The reason code shall be AdminException.NOT\_ENOUGH\_MEMORY\_SPACE.
* CRRC7: If the conditions of use are not satisfied, an instance of AdminException shall be thrown. The reason code shall be AdminException.CONDITIONS\_OF\_USE\_NOT\_SATISFIED.
* CRRC8: If the method resizeFile() is applied to a non compatible file, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND\_INCOMPATIBLE.
* CRRC9: If the method resizeFile() is applied to invalidated data, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF\_DATA\_INVALIDATED.

##### 5.3.1.3.2 Test area files

Test Source: Test\_Api\_4\_Afv\_ Rszf.java.

Test Applet: Api\_4\_Afv\_ Rszf \_1.java.

Cap File: api\_4\_Afv\_ Rszf.cap.

##### 5.3.1.3.3 Test coverage

|  |  |
| --- | --- |
| CRR number | Test case number |
| N1 | 1, 2 |
| P1 | 3 |
| P2 | 4 |
| C1 | Not testable |
| C2 | Not testable |
| C3 | 5 |
| C4 | 6 |
| C5 | Not testable |
| C6 | Not testable |
| C7 | Not testable |
| C8 | 7 |
| C9 | 8 |

##### 5.3.1.3.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 1 | Resize a Transparent EF  1- Trigger the applet with an unrecognized envelope  2- Select MF/DFTEST  3- Call ResizeFile() on EFTARU  new size: 10 bytes  Applet finalizes  4- Select MF/DFTEST/EFTARU and check size in the returned FCP template.  5- Restore EFTARU |  | 4- should return a size of 10 bytes |
| 2 | Resize a Linear Fixed EF  1- Trigger the applet with an unrecognized envelope  2- Select MF/DFTEST  3- Call ResizeFile() on EFLARU  add 2 records.  Applet finalizes  4- Select MF/DFTEST/EFLARU and check size in the returned FCP template.  5- Restore EFLARU |  | 4- should return a size of 16 bytes |
| 3 | Call resizeFile with a null viewHandler  1- Call resizeFile with null. | 1- java.lang.NullPointerException shall be thrown |  |
| 4 | Call createFile with incorrect parameters  1- Call createFile with incorrect parameters. | 1- AdminExceptiogn.INCORRECT\_PARAMETERS shall be thrown |  |
| 5 | File not found  1- Select MF/DFTEST  2- Call resizeFile(DFRFU1)  3- Select ADF1/DFTEST  4- Call resizeFile(EFRFU1) | 2- UICCException.FILE\_NOT\_FOUND shall be thrown  4- UICCException.FILE\_NOT\_FOUND shall be thrown |  |
| 6 | Security status not satisfied  1- Select MF/DFTEST/DFARR2  2- Call resizeFile(EFTAR2T) | 2- UICCException.SECURITY\_STATUS\_NOT\_SATISFIED shall be thrown |  |
| 7 | Command incompatible  1- Select MF/DFTEST, call resizeFile(EFCARU) | 1- UICCException.COMMAND\_INCOMPATIBLE shall be thrown |  |
| 8 | Invalidated data  1- Select MF/DFTEST  2- Invalidate EFTARU  3- Call resizeFile(EFTARU)  4- Validate EFTARU | 3- UICCException.REF\_DATA\_INVALIDATED shall be thrown. |  |

#### 5.3.1.4 Method select (byte sfi)

Test Area Reference: Api\_4\_Afv\_Slctb.

##### 5.3.1.4.1 Conformance requirement

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

public void select(byte sfi)

throws UICCException

5.3.1.4.1.1 Normal execution

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

5.3.1.4.1.2 Parameter errors

* CRRP1: If the file which sfi matches is not in the current directory or no file matches the sfi, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE\_NOT\_FOUND.

5.3.1.4.1.3 Context errors

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

##### 5.3.1.4.2 Test area files

Test Source: Test\_Api\_4\_Afv\_ Slctb.java.

Test Applet: Api\_4\_Afv\_ Slctb \_1.java.

Cap File: api\_4\_Afv\_ slctb.cap.

##### 5.3.1.4.3 Test coverage

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

##### 5.3.1.4.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 1 | Selection possibilities, UICC file system  1- get an AdminFileView AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR\_ON\_RESET)  2- select DFTEST, fid=1111  3- select with sfi EFTNR, sfi=0x01  4- select with sfi EFTNU, sfi=0x02  5- select with sfi EFCNU, sfi=0x05 | 2- no exception shall be thrown  3- no exception shall be thrown  4- no exception shall be thrown  5- no exception shall be thrown |  |
| 2 | Selection possibilities, ADF1  1- get an AdminFileView AdminFileViewBuilder.getTheAdminFileView(AID\_ADF1,CLEAR\_ON\_RESET)  2- select DFTEST, fid=1111  3- select with sfi EFTNR, sfi=0x01  4- select with sfi EFTNU, sfi=0x02  5- select with sfi EFCNU, sfi=0x05 | 2- no exception shall be thrown  3- no exception shall be thrown  4- no exception shall be thrown  5- no exception shall be thrown |  |
| 3 | Current EF itself can be selected  1- get an AdminFileView AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR\_ON\_RESET)  2- select DFTEST, fid=1111  3- select with sfi EFTNR, sfi=0x01  4- select with sfi EFTNR, sfi=0x01 | 4- no exception shall be thrown |  |
| 4 | FILE\_NOT\_FOUND  1- try to select a file with sfi=0x55 | 1- shall throw an uicc.access.UICCException with reason code FILE\_NOT\_FOUND |  |
| 5 | File context changed  1- select EFTARU, sfi=0x03  read 3 first bytes  2- select EFTNU, sfi=0x02  read file content | 1- file content should be {0xFF,0xFF,0xFF}  2- file content should be  {0x55,0x55,0x55} |  |

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

Test Area Reference: Api\_4\_Afv\_Slctb\_bss.

##### 5.3.1.5.1 Conformance requirement

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

public short select(short fid,

byte[] fcp,

short fcpOffset,

short fcpLength)

throws java.lang.NullPointerException,

java.lang.ArrayIndexOutOfBoundsException,

UICCException

5.3.1.5.1.1 Normal execution

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

5.3.1.5.1.2 Parameter errors

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

5.3.1.5.1.3 Context errors

* CRRC1: If the file with a File-ID which matches fid could not be found according to the selection rules, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE\_NOT\_FOUND.
* CRRC2: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY\_PROBLEM.
* CRRC3: If the method call causes an error to occur that is not expected and thus not handled, an instance of SIMViewException shall be thrown. The reason code shall be UICCException.INTERNAL\_ERROR.

##### 5.3.1.5.2 Test area files

Test Source: Test\_Api\_4\_Afv\_Slctb\_bss.java.

Test Applet: Api\_4\_Afv\_Slctb\_bss\_1.java.

Cap File: Api\_4\_Afv\_slctb\_bss.cap.

##### 5.3.1.5.3 Test coverage

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

##### 5.3.1.5.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 0 | Get an AdminFileView object, UICC file system  get an AdminFileView FileView.getTheUICCAdminFileView(CLEAR\_ON\_RESET) |  |  |
| 1 | Select EFTARU in MF (Transparent EF)  Select DFTEST  select EFTARU,  fid=6F03  byte[] fcp = new byte[132]  fcpOffset = 0  fcpLength = 127 | Shall return at least 19.  fcp[] shall contain following TLVs:  1. 82 02 41 21 //file descriptor  2. 83 02 2F E2 //file id  3. 8A 01 05 //life cycle status  4. 80 02 00 0A // file size |  |
| 2 | Select EFTARU in MF (Transparent EF)  select EF**TARU**,  fid=6F03  fcpOffset = 0  fcpLength = 7  select() | Shall return 7.  fcp[] shall contain the first 7 bytes of the FCP structure and contain following TLV.  1. 82 02 41 21 //file descriptor |  |
| 3 | Select DFTEST in MF  fid = DFTEST, fid=1111  fcpOffset = 0  fcpLength = 127  select() | Shall return at least 17.  fcp[] shall contain following TLVs  1. 82 02 78 21 //file descriptor  2. 83 02 11 11 //file id  3. 8A 01 05 //life cycle status |  |
| 4 | Select EFCARU in DFTEST (Cyclic EF)  select EFCARU,  fid=6f09  fcpOffset = 0  fcpLength = 11  select() | Shall return: 11  fcp[] shall contain following TLV:  82 05 46 21 00 03 02 |  |
| 5 | Select ADF1  select ADF  fid=7FFF  fcp[0:5]=0x00  fcpOffset=5  fcpLength=127  select | Shall return: at least 27  The first 5 bytes of fcp[] shall be 0x00 and contains following TLVs:  1. 82 02 78 21 //file descriptor  2. 84 10 A0 00 00 00 09 00 05 FF FF FF FF 89 60 00 00 00 //DF Name  3. 8A 01 05 //life cycle |  |
| 6 | Select MF  select MF,  fid= 3F00  fcpOffset = 0  fcpLength = 11  select() | Shall return: 11  fcp[] shall contain following TLVs:  1. 82 02 38/78 21 //file descriptor  2. 83 02 3F 00 //file ID |  |
| 7 | Select DFTELECOM in MF  select DFTELECOM,  fid=7F10  fcp[0] = fcp[1] = 0x05  fcpOffset = 2  fcpLength = 13  select() | Shall return 13.  The first 2 bytes of fcp[] shall be 0x05 and fcp[] shall contain following TLVs  1. 82 02 38/78 21 //file descriptor  2. 83 02 7F 10 //file id |  |
| 8 | Select EFLARU in DFTELECOM (Linear FixedEF)  select EFLARU,  fid = 6F0C  fcpOffset = 0  fcpLength = 14 | Shall return 14.  fcp[] shall contain following TLVs:  1. 82 05 42 21 00 04 02  2. 83 02 6F 0C |  |
| 9 | fcp is null  select EFLARU,  fid = 6F0C  byte[] nullBuffer = null  fcpOffset = 0  fcpLength = 15 | Shall throw java.lang.NullPointerException |  |
| 10 | fcpOffset < 0  select EFLARU,  fid = 6F0C  fcpOffset = -1  fcpLength = 15 | Shall throw java.lang.ArrayIndexOutOfBoundsException |  |
| 11 | fcpLength < 0  select EFLARU,  fid = 6F0C  fcpOffset = 0  fcpLength = -1 | Shall throw java.lang.ArrayIndexOutOfBoundsException |  |
| 12 | fcpOffset + fcpLength > fcp.length  select EFLARU,  fid = 6F0C  fcpOffset = 115  fcpLength = 18 | Shall throw java.lang.ArrayIndexOutOfBoundsException |  |
| 13 | fcpOffset + fcpLength > fcp.length  select EFLARU,  fid = 6F0C  fcpOffset = fcp.length+1  fcpLength = 0 | Shall throw java.lang.ArrayIndexOutOfBoundsException |  |
| 14 | Selection possibilities  0- select MF, fid=3F00  1- select EFUICC, fid=2FF0  2- select DFTEST, fid=1111  3- select EFCNU, fid=6F05  4- select EFTAAA, fid=6F16  5- select DFSUB\_TEST, fid=2211  6- select DFTEST, fid=1111  7- select EFTAAA, fid=6F16  8- select DFTEST, fid=1111  9- select MF, fid=3F00  10- select DFTEST, fid=1111  11- select EFTAAA, fid=6F16  12- select MF, fid=3F00 | No exception shall be thrown. |  |
| 15 | EF not selected after MF/DF selection  1- select MF,  fid = 3F00  select EFICCID,  fid = 2FE2  2 - select MF  fid = 3F00  select()  readBinary() | 2 - Shall throw uicc.access.UICCException with reason code NO\_EF\_SELECTED. |  |
| 16 | No selection of non-reachable file  1 - select MF,  fid = 3F00  2 - select EFCARU,  fid= 0x6F09 | 2 - Shall throw uicc.access.UICCException with reason code FILE\_NOT\_FOUND. |  |
| 17 | No record is selected after selecting linear fixed EF  1- select MF,  fid = 3F00  2- select DFTEST,  fid=1111  3- select EFLARU,  fid=6F0C  4 - recNumber = 0  mode = REC\_ACC\_MODE\_ CURRENT  readRecord() | 4 - Shall throw uicc.access.UICC Exception with reason code RECORD\_NOT\_FOUND. |  |
| 18 | Record pointer in selected cyclic EF  1- select MF,  fid = 3F00  2- select DFTEST,  fid=1111  3- select EFCARU,  fid=6F09  4- byte[] data1 = { 1,2,3 }  mode = REC\_ACC\_MODE\_PREVIOUS  updateRecord(data1)  5- select EFCARU  fid = 6F09  select()  mode = REC\_ACC\_MODE\_PREVIOUS readRecord()  readRecord(data2)  compare data1 to data2  6- restore original data of EFCARU | 5 - The contents of data1 and data2 shall be identical. |  |
| 19 | EF not selected after ADF/DF selection  1- get an AdminFileView AdminFileViewBuilder.getTheAdminFileView(AID\_ADF1,CLEAR\_ON\_RESET)  2- select ADF,  fid = 7FFF  select EFUICC,  fid = 2FF0  3 - select ADF  fid = 7FFF  select()  readBinary() | 3 - Shall throw uicc.access.UICCException with reason code NO\_EF\_SELECTED. |  |
| 20 | Reselection  1- Using the ADF FileView  select ADF, fid=7FFF  select ADF, fid=7FFF  2- Using the UICC FileView  select MF, fid=3F00  select MF, fid=3F00  3- select DFTEST, fid=1111  select DFTEST, fid=1111  5- select EFTAAA, fid=6F16  select EFTAAA, fid=6F16 | No exceptions shall be thrown |  |
| 21 | Security attributes  1- Using the ADF FileView  select ADF, fid=7FFF  select DFTEST, fid=1111  select EFLARR1, fid=6FA1  2- Using the UICC FileView  select MF, fid=3F00  select DFTEST, fid=1111  select EFTARR3, fid=6FB3 | 1- fcp[] shall contain the following TLV  8B 03 AC 00 01 or  8B 06 AC 00 00 01 01 01  2- fcp[] shall contain the following TLV  8B 03 AC 00 03 or  8B 06 AC 00 00 03 01 03 |  |

#### 5.3.1.6 Method select (short fid)

Test Area Reference: Api\_4\_Afv\_Slcts.

##### 5.3.1.6.1 Conformance requirement

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

public public void select(short fid)

throws UICCException

5.3.1.6.1.1 Normal execution

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

5.3.1.6.1.2 Parameter errors

No requirements.

5.3.1.6.1.3 Context errors

* CRRC1: If the file with a File Identifier which matches fid could not be found according to the selection rule listed in CCRN3, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE\_NOT\_FOUND.
* CRRC2: If the file with a File Identifier which matches fid could not be found according to the selection rule listed in CCRN4, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE\_NOT\_FOUND.
* CRRC3: If the file with a File Identifier which matches fid could not be found according to the selection rule listed in CCRN5, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE\_NOT\_FOUND.
* CRRC4: If the file with a File Identifier which matches fid could not be found according to the selection rule listed in CCRN6, an instance of UICCException shall be thrown. The reason code shall be UICCException.FILE\_NOT\_FOUND.
* CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY\_PROBLEM.
* CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL\_ERROR.

##### 5.3.1.6.2 Test area files

Test Source: Test\_Api\_4\_Afv\_Slcts.java.

Test Applet: Api\_4\_Afv\_Slcts\_1.java.

Cap File: api\_4\_Afv\_slcts.cap.

##### 5.3.1.6.3 Test coverage

|  |  |
| --- | --- |
| CRR number | Test case number |
| N1 | 1,2 |
| N2 | 5 |
| N3 | 5 |
| N4 | 6 |
| N5 | 1, 2 |
| N6 | 1, 2 |
| N7 | 1, 2 |
| N8 | 1,2 |
| N9 | 4 |
| N10 | Tested in Api\_1\_Cont, test case 1 and 2 |
| C1 | 3 |
| C2 | 3 |
| C3 | 3 |
| C4 | 3 |
| C5 | Not testable |
| C6 | Not testable |

##### 5.3.1.6.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 0 | Get an AdminFileView object, UICC file system  1- get an AdminFileView AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR\_ON\_RESET) |  |  |
| 1 | Selection possibilities  1- select EFUICC, fid=2FF0  2- select DFTEST, fid=1111  3- select EFCNU, fid=6F05  4- select EFTAAA, fid=6F16  5- select DFSUB\_TEST, fid=2211  6- select DFTEST, fid=1111  7- select EFTAAA, fid=6F16  8- select DFTEST, fid=1111  9- select MF, fid=3F00  10- select DFTEST, fid=1111  11- select EFTAAA, fid=6F16  12- select MF, fid=3F00 | No exception shall be thrown |  |
| 2 | Selection possibilities, ADF1  1- get an AdminFileView AdminFileViewBuilder.getTheAdminFileView(AID\_ADF1,CLEAR\_ON\_RESET)  2- select EFUICC, fid=2FF0  3- select DFTEST, fid=1111  4- select EFCNU, fid=6F05  5- select EFTAAA, fid=6F16  6- select DFSUB\_TEST, fid=2211  7- select DFTEST, fid=1111  8- select EFTAAA, fid=6F16  9- select DFTEST, fid=1111 | No exception shall be thrown |  |
| 3 | No selection of unreachable file  1- get an AdminFileView AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR\_ON\_RESET)  2- select EFCNU, fid=6F05  3- select DFTEST, fid=1111  4- select EFTAA, fid=2222  5- select EFCNU, fid=6F05  6- select DFSUB\_TEST, fid=2211  7- select EFTAA, fid=2222  8- select DFTELECOM, fid=7F10 | 2- A UICCException.FILE\_NOT\_FOUND shall be thrown.  3- No exception shall be thrown  4- A UICCException.FILE\_NOT\_FOUND shall be thrown.  5- No exception shall be thrown  6- No exception shall be thrown  7- No exception shall be thrown  8- A UICCException.FILE\_NOT\_FOUND shall be thrown. |  |
| 4 | Self selection  1- select MF, fid=3F00  2- select MF, fid=3F00  3- select DFTEST, fid=1111  4- select DFTEST, fid=1111  5- select EFTAAA, fid=6F16  6- select EFTAAA, fid=6F16  7- get an AdminFileView AdminFileViewBuilder.getTheAdminFileView(AID\_ADF1,CLEAR\_ON\_RESET)  8- select ADF, fid=7FFF  9- select ADF, fid=7FFF | 2- No exception shall be thrown  4- No exception shall be thrown  6- No exception shall be thrown  8- No exception shall be thrown  9- No exception shall be thrown |  |
| 5 | EF not selected after MF/DF selection  1- select MF, fid=3F00  2- updateRecord()  3- select DFTEST, fid=1111  4- updateRecord() | 2- A UICCException.NO\_EF\_SELECTED shall be thrown  4- A UICCException.NO\_EF\_SELECTED shall be thrown |  |
| 6 | No record is selected after selecting linear fixed EF  1- select MF,  fid = 3F00  2- select DFTEST,  3- select EFLARU,  4 - recNumber = 0  mode = REC\_ACC\_MODE\_ CURRENT  readRecord()  5- select EFCARU,  6 - recNumber = 0  mode = REC\_ACC\_MODE\_ CURRENT  readRecord() | 1 - No exception shall be thrown.  2 - No exception shall be thrown.  3 - No exception shall be thrown.  4 - Shall throw uicc.access.UICC Exception with reason code RECORD\_NOT\_FOUND.  5 - No exception shall be thrown.  6 - Shall throw uicc.access.UICC Exception with reason code RECORD\_NOT\_FOUND. |  |

#### 5.3.1.7 Method status

Test Area Reference: Api\_4\_Afv\_Stat.

##### 5.3.1.7.1 Conformance requirement

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

public short status(byte[] fcp,

short fcpOffset,

short fcpLength)

throws java.lang.NullPointerException,

java.lang.ArrayIndexOutOfBoundsException,

UICCException

5.3.1.7.1.1 Normal execution

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

5.3.1.7.1.2 Parameter errors

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

5.3.1.7.1.3 Context errors

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

##### 5.3.1.7.2 Test area files

Test Source: Test\_Api\_4\_Afv\_Stat.java.

Test Applet: Api\_4\_Afv\_Stat\_1.java.

Cap File: Api\_4\_Afv\_Stat.cap.

##### 5.3.1.7.3 Test coverage

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

##### 5.3.1.7.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 1 | Status of MF  1- Get an AdminFileView object, UICC file get an AdminFileView FileView.getTheUICCAdminFileView(CLEAR\_ON\_RESET)  2- select MF  byte[] fcp = new byte[127]  fcp[0:2]= 0xCC  fcpOffset = 3  fcpLength = 11  status() | 2- Shall return 11.  The first 3 bytes of fcp[] shall contain 0xCC.  fcp[] shall contain following TLVs:  1. 82 02 38/78 21 //file descriptor  2. 83 02 3F 00 //file ID |  |
| 2 | Status after select EFTARU in MF  1 - select DF**TEST**  select EF**TARU**, fid = 6F03  fcpOffset = 0  fcpLength = 127  select()  status() | 1- Shall return at least 19.  fcp[] shall contain following TLVs:  1. 82 02 38/78 21 //file descriptor  2. 83 02 11 11 //file ID |  |
| 3 | Status of DFTELECOM  1 - fid = 7F10  fcpOffset = 0  fcpLength = 127  status() | 1 -  Shall return at least 17 and the entire structure of the file control parameters. The file identifier shall be contain the fid of DFTELECOM. |  |
| 4 | Status DFTELECOM  Select DFTELECOM, fid=7F10  fcpOffset = 0  fcpLength = 11  status() | Shall return 11.  fcp shall contain the first 11 bytes of the FCP structure starting at index 0.  fcp[] shall contain following TLVs:  1. 82 02 38/78 21 //file descriptor  2. 83 02 7F 10 //file id |  |
| 5 | Status ADF1  Select ADF, fid=7FFF  fcpOffset = 0  fcpLength = 127 | Shall return at least 27  fcp[] shall contain the entire FCP structure  fcp[] shall contain following TLVs:  1. 82 02 78 21 //file descriptor  2. 84 10 A0 00 00 00 09 00 05 FF FF FF FF 89 60 00 00 00 //DF Name  3. 8A 01 05 //life cycle |  |
| 6 | fcp is null  byte[] nullBuffer = null  fcpOffset = 0  fcpLength = 34  status() | Shall throw java.lang.NullPointerException. |  |
| 7 | fcpOffset < 0  fcpOffset = -1  fcpLength = 34  status() | Shall throw java.lang.ArrayIndexOutOfBoundsException. |  |
| 8 | fcpLength < 0  fcpOffset = 0  fcpLength = -1  status() | Shall throw java.lang.ArrayIndexOutOfBoundsException. |  |
| 9 | fcpOffset + fcpLength > fcp.length  fcpOffset = fcp.length-1  fcpLength = 15  status() | Shall throw java.lang.ArrayIndexOutOfBoundsException. |  |
| 10 | fcpOffset + fcpLength > fcp.length  fcpOffset = fcp.length+1  fcpLength = 0  status() | Shall throw java.lang.ArrayIndexOutOfBoundsException. |  |
| 11 | Security attributes  1- Using the ADF FileView  select ADF, fid=7FFF  select DFTEST,  select DFARR2,  2- Using the UICC FileView  select MF, fid=3F00  select DFTEST,  select DFARR4, | 1- fcp[] shall contain the following TLV  8B 03 AC 00 02 or  8B 06 AC 00 00 02 01 02  2- fcp[] shall contain the following TLV  8B 03 AC 00 04 or  8B 06 AC 00 00 04 01 04 |  |

#### 5.3.1.8 Method readBinary

Test Area Reference: Api\_4\_Afv\_Redb.

##### 5.3.1.8.1 Conformance requirement

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

public short readBinary(short fileOffset,

byte[] resp,

short respOffset,

short respLength)

throws java.lang.NullPointerException,

java.lang.ArrayIndexOutOfBoundsException,

UICCException

5.3.1.8.1.1 Normal execution

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

5.3.1.8.1.2 Parameter errors

* CRRP1: If fileOffset is negative, an instance of UICCException.OUT\_OF\_FILE\_BOUNDARIES shall be thrown.
* CRRP2: If respOffset is negative, an instance of ArrayIndexOutOfBoundsException shall be thrown.
* CRRP3: If respLength is negative, an instace of ArrayIndexOutOfBoundsException shall be thrown.
* CRRP4: If the array resp is null, an instance of NullPointerException shall be thrown.
* CRRP5: If respOffset plus respLength is greater than the length of the array resp.length, an instance of ArrayIndexOutOfBoundsException shall be thrown and no read is performed.
* CRRP6: If fileOffset plus respLength exceeds the length of the file, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT\_OF\_FILE\_BOUNDARIES.

5.3.1.8.1.3 Context errors

* CRRC1: If the method call causes an error to occur that is not expected and thus not handled, an instace of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL\_ERROR.
* CRRC2: If the currently selected EF is not transparent, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND\_INCOMPATIBLE.
* CRRC3: If the calling applet does not fulfil the access condition, READ, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY\_STATUS\_NOT\_SATISFIED.
* CRRC4: If the currently selected EF is deactivated and the file status of the EF does not allow for the reading of an deactivated file, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF\_DAT\_INVALIDATED.
* CRRC5:If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO\_EF\_SELECTED.

##### 5.3.1.8.2 Test area files

Test Source: Test\_Api\_4\_Afv\_Redb.java.

Test Applet: Api\_4\_Afv\_Redb\_1.java.

Cap File: Api\_4\_Afv\_redb.cap.

##### 5.3.1.8.3 Test coverage

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

##### 5.3.1.8.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 1 | Read from EFTARU  1- select DFTEST, fid=1111  select EFTARU, fid=6F03  2- fileOffset = 0  resp.length = 260  resp[0:259] = 0x55  respOffset = 10  respLength = 250  readBinary() | 2 - shall return 20.  resp shall contain the contents of EFTARU starting at index 10.  <Description of resp:  55 55 55 55 55 55 55 55 55 55  FF FF FF FF FF FF .. .. FF> |  |
| 2 | Read from EFTARU  fileOffset = 0x80  resp.length = 260  resp[0:259] = 0x55  respOffset = 5  respLength = 0x80  readBinary() | shall return 15  resp shall contain the last 5 bytes of EFTARU starting at index 10.  <Description of resp:  55 55 55 55 55 55 55 55 55 55  FF FF FF FF FF … 55 55 .. .. 55 FF FF … FF > |  |
| 3 | FileOffset is negative  fileOffset = -1  respOffset = 0  respLength = 10  readBinary() | Shall throw uicc.access.UICC Exception with reason code OUT\_OF\_FILE\_BOUNDARIES. |  |
| 4 | FileOffset + respLength > EF length  fileOffset = 259  respOffset = 0  respLength = 2  readBinary() | Shall throw uicc.access.UICC Exception with reason code OUT\_OF\_FILE\_BOUNDARIES. |  |
| 5 | resp[] is null  fileOffset = 0  resp = null  respOffset = 0  respLength = 10  readBinary() | Shall throw java.lang.NullPointerException. |  |
| 6 | respOffset < 0  fileOffset = 0  respOffset = -1  respLength = 10  readBinary() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 7 | respLength < 0  fileOffset = 0  respOffset = 0  respLength = -1  readBinary() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 8 | RespOffset + respLength > resp.length  fileOffset = 0  resp.length = 20  respOffset = 10  respLength = 11  readBinary() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 9 | EF is not Transparent   1. select EFLARU, fid=6F0C 2. fileOffset = 0   respOffset = 0  respLength = 1  readBinary() | 2 - Shall throw uicc.access.UICC Exception with reason code COMMAND\_INCOMPATIBLE. |  |
| 10 | Access condition not fulfilled  1- select EFTRAC, fid=6F0E  2- fileOffset = 0  respOffset = 0  respLength = 1  readBinary() | 2- Shall throw uicc.access.UICC Exception with reason code SECURITY\_STATUS\_NOT\_SATISFIED. |  |
| 11 | EF is deactivated  1 - select EFTARU, fid=6F03  2 - deactivateFile()  3 - readBinary()  4 - activateFile()) | 3 - Shall throw uicc.access.UICCException with reason code REF\_DATA\_INVALIDATED. |  |
| 12 | No EF selected  1- select DFTEST fid=1111  2 readBinary() | 2 - Shall throw uicc.access.UICCException with reason code NO\_EF\_SELECTED. |  |

#### 5.3.1.9 Method updateBinary

Test Area Reference: Api\_4\_Afv\_Updb.

##### 5.3.1.9.1 Conformance requirement

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

public void updateBinary(short fileOffset,

byte[] data,

short dataOffset,

short dataLength)

throws java.lang.NullPointerException,

java.lang.ArrayIndexOutOfBoundsException,

UICCException

5.3.1.9.1.1 Normal execution

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

5.3.1.9.1.2 Parameter errors

* CRRP1: If recOffset is less than 0, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT\_OF\_FILE\_BOUNDARIES.
* CRRP2: : If fileOffset plus dataLength exceeds the length of the file, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT\_OF\_FILE\_BOUNDARIES.
* CRRP3: If the array data is null, an instance of NullPointerException shall be thrown.
* CRRP4: If dataOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
* CRRP5: If dataLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
* CRRP6: If dataOffset plus dataLength greater than the length of the array data.length an instance of ArrayIndexOutOfBoundsException shall be thrown.

5.3.1.9.1.3 Context errors

* CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO\_EF\_SELECTED.
* CRRC2: If the currently selected EF is not transparent, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND\_INCOMPATIBLE.
* CRRC3: If the calling applet does not fulfil the access condition, UPDATE, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY\_STATUS\_NOT\_SATISFIED.
* CRRC4: If the currently selected EF is deactivated and the file status of the EF does not allow for updating of a deactivated file, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF\_DATA\_INVALIDATED.
* CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY\_PROBLEM.
* CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL\_ERROR.

##### 5.3.1.9.2 Test area files

Test Source: Test\_Api\_4\_Afv\_Updb.java.

Test Applet: Api\_4\_Afv\_Updb \_1.java.

Cap File: Api\_4\_Afv\_updb.cap.

##### 5.3.1.9.3 Test coverage

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

##### 5.3.1.9.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 0 | Get the UICC FileView  AdminFileViewBuilder.getTheUICCAdminFileView(CLEAR\_ON\_RESET) |  | 0 |
| 1 | No EF selected  fileOffset = 0  byte[] data = new byte[20]  data[0] = '55'  dataOffset = 0  dataLength = 10  updateBinary() | Shall throw uicc.access.UICC Exception with reason code NO\_EF\_SELECTED. | 1 |
| 2 | Update Transparent EF  1- select DFTEST, fid = 1111  2- select EFTARU, fid = 6F03  3- fileOffset = 3  data[0] = '55'  dataOffset = 0  dataLength = 1  updateBinary()  4- fileOffset = 3  respOffset = 0  respLength = 1  readBinary() | 1- No exception shall be thrown.  2- No exception shall be thrown.  3- No exception shall be thrown.  4- No exception shall be thrown.  Data in resp[0] shall be '55'. |  |
| 3 | fileOffset = 254  1- fileOffset = 254  data[0] = '55'  data[1] = 'AA'  data[2] = '66'  dataOffset = 0  dataLength = 3  updateBinary()  2- fileOffset = 254  respOffset = 0  respLength = 3  readBinary() | 1- No exception shall be thrown.  2- No exception shall be thrown.  Data in resp shall be  resp[0] = '55'  resp[1] = 'AA'  resp[2] = '66' |  |
| 4 | Offset into File out of bounds  fileOffset = -1  dataOffset = 0  dataLength = 10  updateBinary() | Shall throw uicc.access.UICCException with reason code OUT\_OF\_FILE\_BOUNDARIES. |  |
| 5 | fileOffset + dataLength > EF length  fileOffset = 259  dataOffset = 0  dataLength = 2  updateBinary() | Shall throw uicc.access.UICC Exception with reason code OUT\_OF\_FILE\_BOUNDARIES. |  |
| 6 | data is null  byte[] nullBuffer = null  fileOffset = 0  dataOffset = 0  dataLength = 10  updateBinary() | Shall throw java.lang.NullPointerException. |  |
| 7 | dataOffset < 0  fileOffset = 0  dataOffset = -1  dataLength = 10  updateBinary() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 8 | dataLength < 0  fileOffset = 0  dataOffset = 0  dataLength = -1  updateBinary() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 9 | dataOffset + dataLength > data.length  fileOffset = 0  dataOffset = 10  dataLength = 11  updateBinary() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 10 | EF is not Transparent  1- select DFTEST, fid = 1111  2- select DFLARU, fid = 6F0C  3 - fileOffset = 0  data[0] = '55'  dataOffset = 0  dataLength = 1  updateBinary() | 1- No exception shall be thrown.  2- No exception shall be thrown.  3- Shall throw uicc.access.UICC Exception with reason code COMMAND\_INCOMPATIBLE. |  |
| 11 | Access condition not fulfilled  1- select DFTEST, fid = 1111  2- select EFTNU, fid = 6F02  3- fileOffset = 0  data[0] = '55'  dataOffset = 0  dataLength = 1  updateBinary() | 1- No exception shall be thrown.  2- No exception shall be thrown.  3- Shall throw uicc.access.UICCException with reason code SECURITY\_STATUS\_NOT\_SATISFIED. |  |
| 12 | EF is deactivated  1- select EFTNR, fid = 6F01  deactiveFile()  2- fileOffset = 0  data[0] = '55'  dataOffset = 0  dataLength = 1  updateBinary()  3- activateFile() | 1- No exception shall be thrown.  2- Shall throw uicc.access.UICCException with reason code REF\_DATA\_INVALIDATED  3- No exception shall be thrown. |  |

#### 5.3.1.10 Method readRecord

Test Area Reference: Api\_4\_Afv\_Redr.

##### 5.3.1.10.1 Conformance requirement

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

public short readRecord(short recNumber,

byte mode,

short recOffset,

byte[] resp,

short respOffset,

short respLength)

throws java.lang.NullPointerException,

java.lang.ArrayIndexOutOfBoundsException,

UICCException

5.3.1.10.1.1 Normal execution

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

5.3.1.10.1.2 Parameter errors

* CRRP1: If the currently selected EF is linear fixed and the access mode is REC\_ACC\_MODE\_ABSOLUTE and recNumber is less than 0 or greater than records available, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD\_NOT\_FOUND.
* CRRP2: If the currently selected EF is linear fixed and the access mode is REC\_ACC\_MODE\_CURRENT, recNumber is 0 and there is no current record selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD\_NOT\_FOUND.
* CRRP3: If the currently selected EF is linear fixed and the access mode is REC\_ACC\_MODE\_NEXT and the current record pointer is set to the last record, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD\_ NOT\_FOUND.
* CRRP4: If the currently selected EF is linear fixed and the access mode is REC\_ACC\_MODE\_PREVIOUS and the current record pointer is set to the first record, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD\_ NOT\_FOUND.
* CRRP5: If the specified offset into the selected record recOffset is less than 0, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT\_OF\_RECORD\_BOUNDARIES.
* CRRP6: If recOffset plus respLength is greater than the record length, an instance of UICCException shall be thrown. The reason code shall be UICC Exception.OUT\_OF\_RECORD\_BOUNDARIES.
* CRRP7: If the access mode is not between 2 and 4 (2 = REC\_ACC\_MODE\_NEXT, etc.), an instance of UICCException shall be thrown. The reason code shall be UICC Exception.INVALID\_MODE.
* CRRP8: If the array resp is null, an instance of NullPointerException shall be thrown.
* CRRP9: If respOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
* CRRP10: If respLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
* CRRP11: If respOffset plus respLength is greater than the length of the array resp.length, or respOffset equals resp.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

5.3.1.10.1.3 Context errors

* CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO\_EF\_SELECTED.
* CRRC2: If the currently selected EF is neither linear fixed nor cyclic, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND\_INCOMPATIBLE.
* CRRC3: If the calling applet does not fulfil the access condition, READ, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY\_STATUS\_NOT\_SATISFIED.
* CRRC4: If the currently selected EF is invalidated and the file status of the EF does not allow for reading an invalidated file, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF\_DATA\_INVALIDATED.
* CRRC5: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL\_ERROR.

##### 5.3.1.10.2 Test area files

Test Source: Test\_Api\_4\_Afv\_Redr.java.

Test Applet: Api\_4\_Afv\_Rredr\_1.java.

Cap File: Api\_4\_Afv\_redr.cap.

##### 5.3.1.10.3 Test coverage

|  |  |
| --- | --- |
| CRR number | Test case number |
| N1 | 2, 3, 4, 5,6, 8, 9, 10, 11, 12, 13 |
| N2 | 3, 9 |
| N3 | 2, 8 |
| N4 | 4, 5, 10, 11 |
| N5 | 4, 11 |
| N6 | 11 |
| N7 | 6, 7, 12, 13 |
| N8 | 6, 13 |
| N9 | 12 |
| N10 |  |
| P1 | 14 |
| P2 | 15 |
| P3 | 5 |
| P4 | 7 |
| P5 | 16 |
| P6 | 17 |
| P7 | 18 |
| P8 | 19 |
| P9 | 20 |
| P10 | 21 |
| P11 | 22 |
| C1 | 1 |
| C2 | 23 |
| C3 | 24 |
| C4 | 25 |
| C5 | Not testable |

##### 5.3.1.10.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 1 | No EF selected   1. select DFTEST,fid=1111 2. recNumber = 1   mode = REC\_ACC\_MODE\_ABSOLUTE  recOffset = 0  byte[] resp = new byte[20]  respOffset = 0  respLength = 10  readRecord() | 2-Shall throw  uicc.access.UICCException with reason code NO\_EF\_SELECTED. |  |
| 2 | Read Absolute from Linear Fixed EF  1 - select EFLARU, fid=6F0c  // Record pointer not set.  2 - recNumber = 2  mode = REC\_ACC\_MODE\_ABSOLUTE  recOffset = 0  respOffset = 0  respLength = 4  readRecord()  3- recNumber = 0  mode = REC\_ACC\_MODE\_NEXT  recOffset = 0  respOffset = 0  respLength = 4  readRecord() | 2 - resp shall be:  resp={0xAA,0xAA,0xAA,0xAA}  3 - resp shall be:  Resp={0x55,0x55,0x55,0x55} |  |
| 3 | Read Current from Linear Fixed EF  //record pointer shall not be changed  1- recNumber = 0  mode = REC\_ACC\_MODE\_CURRENT  recOffset = 0  respOffset = 0  respLength = 4  readRecord() | resp shall be:  resp={0x55,0x55,0x55,0x55} |  |
| 4 | Read Next from Linear Fixed EF  1- select EFLARU, fid=6F0c  //no record selected  recNumber = 0  mode = REC\_ACC\_MODE\_NEXT  recOffset = 0  respOffset = 0  respLength = 4  readRecord()  2- recNumber = 0  mode = REC\_ACC\_MODE\_NEXT  recOffset = 0  respOffset = 0  respLength = 4  readRecord() | 1- resp shall be:  resp={0x55,0x55,0x55,0x55}  2- resp shall be:  resp={0xAA,0xAA,0xAA,0xAA} |  |
| 5 | Read Next from Linear Fixed EF  recNumber = 0  mode = REC\_ACC\_MODE\_NEXT  recOffset = 0  respOffset = 0  respLength = 4  readRecord() | Shall throw uicc.access.UICC Exception with reason code RECORD\_NOT\_FOUND. |  |
| 6 | Read Previous from Linear Fixed EF  1- recNumber = 0  mode = REC\_ACC\_MODE\_PREVIOUS  recOffset = 0  respOffset = 0  respLength = 4  readRecord()  2- select EFLARU, fid=6F0c  //no record selected  recNumber = 0  mode = REC\_ACC\_MODE\_PREVIOUS  recOffset = 0  respOffset = 0  respLength = 4  readRecord()  3- Set the record to the first record  by reading the file  recNumber = 0  mode = REC\_ACC\_MODE\_PREVIOUS  recOffset = 0  respOffset = 0  respLength = 4 | 1- resp shall be:  resp={0x55,0x55,0x55,0x55}  2- resp shall be:  resp={0xAA,0xAA,0xAA,0xAA}  3- resp={0x55,0x55,0x55,0x55} |  |
| 7 | Read Previous from Linear Fixed EF  recNumber = 0  mode = REC\_ACC\_MODE\_PREVIOUS  recOffset = 0  respOffset = 0  respLength = 4  readRecord() | Shall throw uicc.access.UICCException with reason code RECORD \_NOT\_FOUND. |  |
| 8 | Read Absolute from Cyclic EF  1 select EFCARU, fid = 6F09  2- recNumber = 2  mode = REC\_ACC\_MODE\_ABSOLUTE  recOffset = 0  respOffset = 0  respLength = 3  readRecord()  3- recNumber = 1  readRecord()  4- Read the file in next mode to set the record pointer to the first position.  recNumber = 0  mode = REC\_ACC\_MODE\_NEXT  recOffset = 0  respOffset = 0  respLength = 3  readRecord() | 2 - resp shall be:  resp={0xAA,0xAA,0xAA}  3 - resp shall be:  resp={0xAA,0xAA,0xAA}  4 - resp shall be:  resp={0xAA,0xAA,0xAA} |  |
| 9 | Read Current from Cyclic EF  //record pointer shall not be changed //from testcase before  1- recNumber = 0  mode = REC\_ACC\_MODE\_CURRENT  recOffset = 0  respOffset = 0  respLength = 3  readRecord() | 1. resp shall be:   resp={0xAA,0xAA,0xAA} |  |
| 10 | Read Next from Cyclic EF  recNumber = 0  mode = REC\_ACC\_MODE\_NEXT  recOffset = 0  respOffset = 0  respLength = 3  readRecord() | resp shall be:  resp={0xAA,0xAA,0xAA} |  |
| 11 | Read Next from Cyclic EF  1- recNumber = 0  mode = REC\_ACC\_MODE\_NEXT  recOffset = 0  respOffset = 0  respLength = 3  readRecord()   1. select EFCARU, fid = 6F09   //no rec selected  recNumber = 0  mode = REC\_ACC\_MODE\_NEXT  recOffset = 0  respOffset = 0  respLength = 3  readRecord() | 1- resp shall be:  resp={0x55,0x55,0x55}  2- Shall throw  uicc.access.UICCException with reason code RECORD\_NOT\_FOUND. |  |
| 12 | Read Previous from Cyclic EF  1- recNumber = 0  mode = REC\_ACC\_MODE\_PREVIOUS  recOffset = 0  respOffset = 0  respLength = 3  readRecord() | 1- resp shall be:  resp={0xAA,0xAA,0xAA} |  |
| 13 | Read Previous from Cyclic EF  1- recNumber = 0  mode = REC\_ACC\_MODE\_PREVIOUS  recOffset = 0  respOffset = 0  respLength = 3  readRecord()  2- select EFCARU, fid = 6F09  // no rec selected  recNumber = 0  mode = REC\_ACC\_MODE\_PREVIOUS  recOffset = 0  respOffset = 0  respLength = 3  readRecord() | 1- resp shall be:  resp={0x55,0x55,0x55}  2- resp shall be:  resp={0xAA,0xAA,0xAA} |  |
| 14 | **Read Absolute from Linear Fixed EF beyond Records**  1- select EFLARU, fid=6F0C  2- recNumber = -1  mode = REC\_ACC\_MODE\_ABSOLUTE  recOffset = 0  respOffset = 0  respLength = 4  readRecord()  3- recNumber = 3  readRecord() | 2- Shall throw an uicc.access.UICCException with reason code UICCException.RECORD\_NOT\_FOUND.  3- Shall throw an uicc.access.UICCException with reason code UICCException.RECORD\_NOT\_FOUND. |  |
| 15 | **No current record in linear fixed EF, read current**  1- select EFLARU, fid=6F0C // No current record  2- recNumber = 0 // curr rec  mode = REC\_ACC\_MODE\_CURRENT  recOffset = 0  respOffset = 0  respLength = 4  readRecord() | 2 - Shall throw uicc.access.UICC Exception with reason code  RECORD\_NOT\_FOUND. |  |
| 16 | recOffset < 0  1- select EFLARU, fid=6F0C  2- recNumber = 1 // rec 1  mode = REC\_ACC\_MODE\_ABSOLUTE  recOffset = -1  respOffset = 0  respLength = 4  readRecord() | 2 - Shall throw uicc.access.UICCException with reason code OUT\_OF\_RECORD\_BOUNDARIES. |  |
| 17 | recOffset + respLength > Record Length  1- select EFLARU, fid=6F0C  2- recNumber = 1  mode = REC\_ACC\_MODE\_ABSOLUTE  recOffset = 2  respOffset = 0  respLength = 4  readRecord() | 2 - Shall throw sim.access.SIMViewException with reason code OUT\_OF\_RECORD\_BOUNDARIES. |  |
| 18 | Reading with invalid mode  1- select EFLARU, fid=6F0C  2- recNumber = 0  mode = 1  recOffset = 0  respOffset = 0  respLength = 4  readRecord()  3- mode = 5  readRecord() | 2 - Shall throw uicc.access.UICCException with reason code COMMAND\_INCOMPATIBLE.  3 - Shall throw uicc.access.  UICCException with reason code COMMAND\_INCOMPATIBLE. |  |
| 19 | resp is null  resp[] = null  mode = REC\_ACC\_MODE\_CURRENT  respOffset = 0  respLength = 10  readRecord() | Shall throw java.lang.NullPointerException. |  |
| 20 | respOffset < 0  respOffset = -1  respLength = 10  readRecord () | Shall throw  java.lang.ArrayIndexOutOfBoundsException. |  |
| 21 | respLength < 0  respOffset = 0  respLength = -1  readRecord () | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 22 | respOffset + respLength > resp.length  respOffset = 11  respLength = 4  readRecord () | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 23 | EF is neither Cyclic nor Linear Fixed  1- select EFTNU, fid=6F02  2- respOffset = 0  respLength = 4  readRecord() | 2 - Shall throw uicc.access.UICCException with reason code COMMAND\_INCOMPATIBLE. |  |
| 24 | Access condition not fulfilled  1- select EFCNR, fid=6F04  2 - respLength = 3  readRecord() | 2 - Shall throw uicc.access.UICCException with reason code SECURITY\_STATUS\_NOT\_SATISFIED. |  |
| 25 | EF is deactivated  1 - select EFCNU, fid=6F05  deactivateFile()  2 - readRecord()  3 - activateFile | 2 - Shall throw uicc.access.UICC Exception with reason code REF\_DATA\_INVALIDATED |  |

#### 5.3.1.11 Method updateRecord

Test Area Reference: Api\_4\_Afv\_Updr.

##### 5.3.1.11.1 Conformance requirement

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

public void updateRecord(short recNumber,

byte mode,

short recOffset,

byte[] data,

short dataOffset,

short dataLength)

throws java.lang.NullPointerException,

java.lang.ArrayIndexOutOfBoundsException,

UICCException

5.3.1.11.1.1 Normal execution

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

5.3.1.11.1.2 Parameter errors

* CRRP1: If the currently selected EF is linear fixed and the access mode is REC\_ACC\_MODE\_ABSOLUTE and recNumber is less than 0 or greater than records available, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD\_NOT\_FOUND.
* CRRP2: If the currently selected EF is linear fixed and the access mode is REC\_ACC\_MODE \_CURRENT, recNumber is 0 and there is no current record selected, an instance of UICCException shall be thrown. The reason code shall be UICCException. RECORD\_NOT\_FOUND.
* CRRP3: If the currently selected EF is linear fixed and the access mode is REC\_ACC\_MODE\_NEXT and the current record pointer is set to the last record, an instance of UICCException shall be thrown. The reason code shall be UICCException. RECORD\_NOT\_FOUND.
* CRRP4: If the currently selected EF is linear fixed and the access mode is REC\_ACC\_MODE\_PREVIOUS and the current record pointer is set to the first record; an instance of UICCException shall be thrown. The reason code shall be UICCException. RECORD\_NOT\_FOUND.
* CRRP5: If the specified offset into the selected record recOffset is less than 0, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT\_OF\_RECORD\_BOUNDARIES.
* CRRP6: If recOffset plus dataLength is greater than the record length, an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT\_OF\_RECORD\_BOUNDARIES.
* CRRP7: If the access mode is not between 2 and 4 inclusive (2 = REC\_ACC\_MODE\_NEXT, etc.), an instance of UICCException shall be thrown. The reason code shall be UICCException.INVALID\_MODE.
* CRRP8: If the currently selected EF is cyclic and the mode of record access mode is not REC\_ACC\_MODE\_PREVIOUS, an instance of UICCException shall be thrown. The reason code shall be UICCException.INVALID\_MODE.
* CRRP9: If the array data is null, an instance of NullPointerException shall be thrown.
* CRRP10: If dataOffset is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
* CRRP11: If dataLength is less than 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
* CRRP12: If dataOffset plus dataLength, is greater than the length of the array data.length, or dataOffset equals data.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.

5.3.1.11.1.3 Context errors

* CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO\_EF\_SELECTED.
* CRRC2: If the currently selected EF is neither linear fixed nor cyclic, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND\_INCOMPATIBLE.
* CRRC3: If the calling applet does not fulfil the access condition, UPDATE, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException. SECURITY\_STATUS\_NOT\_SATISFIED.
* CRRC4: If the currently selected EF is deactivated and the file status of the EF does not allow for updating an deactivated file, an instance of UICCException shall be thrown. The reason code shall be UICCException. REF\_DATA\_INVALIDATED.
* CRRC5: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException shall be thrown. The reason code shall be UICCException.MEMORY\_PROBLEM.
* CRRC6: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL\_ERROR.

##### 5.3.1.11.2 Test area files

Test Source: Test\_Api\_4\_Afv\_Updr.java.

Test Applet: Api\_4\_Afv\_ Updr\_1.java.

Cap File: Api\_4\_Afv\_updr.cap.

##### 5.3.1.11.3 Test coverage

|  |  |
| --- | --- |
| CRR number | Test case number |
| N1 | 2, 3, 4, 5, 7, 8, 10 |
| N2 | 2 |
| N3 | 3 |
| N4 | 5 |
| N5 | 4 |
| N6 | 7, 8, 9, 10 |
| N7 | 7 |
| N8 | 10 |
| P1 | 11 |
| P2 | 12 |
| P3 | 6 |
| P4 | 9 |
| P5 | 13 |
| P6 | 14 |
| P7 | 15 |
| P8 | 16 |
| P9 | 17 |
| P10 | 18 |
| P11 | 19 |
| P12 | 20 |
| C1 | 1 |
| C2 | 21 |
| C3 | 22 |
| C4 | 23 |
| C5 | Not testable |
| C6 | Not testable |

##### 5.3.1.11.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 0 | Get the UICC AdminFileView  AdminFileViewBuilder.getTheUICCAdminFileView (CLEAR\_ON\_RESET) |  |  |
| 1 | No EF selected  RecNumber = 1  mode = REC\_ACC\_MODE\_ABSOLUTE  recOffset = 0  byte[] data = new byte[20]  dataOffset = 0  dataLength = 10  updateRecord() | Shall throw uicc.access.UICC Exception with reason code NO\_EF\_SELECTED. |  |
| 2 | Update Absolute from Linear Fixed EF  1- select DFTEST, fid = 1111  2- select EFLARU, fid = 6F0C  // Record pointer not set.  3- recNumber = 2  mode = REC\_ACC\_MODE\_ABSOLUTE  data[0:3] = '11'  recOffset = 0  dataOffset = 0  dataLength = 4  updateRecord()  respOffset = 0  respLength = 0  readRecord()  4- // verify result  read respOffset = 0  respLength = 4  recNumber = 0  mode = REC\_ACC\_MODE\_CURRENT  readRecord() | 1- No exception shall be thrown.  2- No exception shall be thrown.  3- No exception shall be thrown.  4- Resp shall be: 11 11 11 11 |  |
| 3 | Update Current from Linear Fixed EF  1- select DFTEST, fid = 1111  2- select EFLARU, fid = 6F0C  // Set record pointer with mode "next".  3- recNumber = 0  mode = REC\_ACC\_MODE\_NEXT  recOffset = 0  data[0:3] = '00'  dataOffset = 0  dataLength = 4  updateRecord()  // write data with mode "current"  4- recNumber = 0  data[0:3] = '22'  mode = REC\_ACC\_MODECURRENT  updateRecord()  5- //verify result  respOffset = 0  respLength = 4  recNumber = 0  mode = REC\_ACC\_MODE\_ CURRENT  readRecord() | 1- No exception shall be thrown.  2- No exception shall be thrown.  3- No exception shall be thrown.  5- No exception shall be thrown.  resp shall be:  resp[0] = '22'  resp[1] = '22'  resp[2] = '22'  resp[3] = '22' |  |
| 4 | Update Next from Linear Fixed EF, no record pointer set  1- select DFTEST, fid = 1111  2- select EFLARU, fid = 6F0C  3- recNumber = 0  mode = REC\_ACC\_MODE\_NEXT  recOffset = 0  data[0:3] = '33'  dataOffset = respOffset = 0  dataLength = respLength = 4  updateRecord()  mode = REC\_ACC\_MODE\_ABSOLUTE\_CURRENT  4-// verify result  readRecord() | 1- No exception shall be thrown.  2- No exception shall be thrown.  4- No exception shall be thrown.  resp shall be:  resp[0] = '33'  resp[1] = '33'  resp[2] = '33'  resp[3] = '33' |  |
| 5 | Update Next from Linear Fixed EF, record pointer set  1- recNumber = 0  mode = REC\_ACC\_MODE\_NEXT  recOffset = 0  data[0:3] = '44'  dataOffset = 0  dataLength = 4  updateRecord()  2- //verify result  readRecord() | 1- No exception shall be thrown.  2- No exception shall be thrown.  resp shall be:  resp[0] = '44'  resp[1] = '44'  resp[2] = '44'  resp[3] = '44' |  |
| 6 | Update Next from Linear Fixed EF, no more records  recNumber = 0  mode = REC\_ACC\_MODE\_NEXT  recOffset = 0  data[0:3] = '55'  dataOffset = 0  dataLength = 4  updateRecord() | Shall throw uicc.access.UICCException with reason code RECORD\_NOT\_FOUND. |  |
| 7 | Update Previous from Linear Fixed EF, no record pointer set  1- select DFTEST, fid = 1111  2- select EFLARU, fid = 6F0C  3- recNumber = 0  mode = REC\_ACC\_MODE\_PREVIOUS  recOffset = 0  data[0:3] = '66'  dataOffset = respOffset = 0  dataLength = respLength = 4  updateRecord()  4- //verify result  readRecord() | 1- No exception shall be thrown.  2- No exception shall be thrown.  3- No exception shall be thrown.  4- No exception shall be thrown.  resp shall be:  resp[0] = '66'  resp[1] = '66'  resp[2] = '66'  resp[3] = '66' |  |
| 8 | Update Previous from Linear Fixed EF, record pointer set  1- recNumber = 0  mode = REC\_ACC\_MODE\_PREVIOUS  recOffset = 0  data[0:3] = '77'  dataOffset = respOffset = 0  dataLength = respLength = 4  updateRecord()  2- //verify result  readRecord() | 1- No exception shall be thrown  2- No exception shall be thrown.  esp shall be:  resp[0] = '77'  resp[1] = '77'  resp[2] = '77'  resp[3] = '77' |  |
| 9 | Update Previous from Linear Fixed EF , no more records  recNumber = 0  mode = REC\_ACC\_MODE\_PREVIOUS  recOffset = 0  data[0:3] = '88'  dataOffset = respOffset = 0  dataLength = respLength = 4 | Shall throw sim.access.SIMViewException with reason code RECORD\_NOT\_FOUND. |  |
| 10 | Update Previous from Cyclic EF  1- select DFTEST, fid = 1111  2- select EFCARU, fid = 6F09  3- recNumber = 2  mode = REC\_ACC\_MODE\_ABSOLUTE  recOffset = 0  respOffset = 0  respLength = 3  readRecord()  4- recNumber = 2  mode = REC\_ACC\_MODE\_PREVIOUS  data[0:2] = 'FF'  dataOffset = 0  dataLength = 3  updateRecord()  5- //verify result  readRecord() | 1- No exception shall be thrown.  2- No exception shall be thrown.  3- No exception shall be thrown.  4- No exception shall be thrown.  5- No exception shall be thrown.  resp shall be:  resp[0] = 'FF'resp[1] = 'FF'  resp[2] = 'FF' |  |
| 11 | Update Absolute from Linear Fixed EF beyond Records  1- select EFLARU, fid = 6F0C  2-recNumber = -1  mode = REC\_ACC\_MODE\_ABSOLUTE  recOffset = 0  dataOffset = 0  dataLength = 4  updateRecord()  3- recNumber = 3  updateRecord() | 1- No exception shall be thrown.  2- Shall throw uicc.access.UICCException with reason code RECORD\_NOT\_FOUND.  3- Shall throw uicc.access.UICCException with reason code RECORD\_NOT\_FOUND. |  |
| 12 | No current record in linear fixed EF, update current  1- select EFLARU, fid = 6F0C  // No curr rec  2- recNumber = 0 // curr rec  mode = REC\_ACC\_MODE \_CURRENT  recOffset = 0  dataOffset = 0  dataLength = 4  updateRecord() | 1 - No exception shall be thrown.  2 - Shall throw uicc.access.UICCException with reason code RECORD\_NOT\_FOUND. |  |
| 13 | recOffset < 0  1- select EFLARU, fid = 6F0C  2- recNumber = 1 // rec 1  mode = REC\_ACC\_MODE\_ABSOLUTE  dataOffset = 0  dataLength = 4  updateRecord() | 1- No exception shall be thrown.  2- Shall throw uicc.access.UICC Exception with reason code OUT\_OF\_RECORD\_BOUNDARIES. |  |
| 14 | recOffset + dataLength > record.length  1- select EFLARU, fid = 6F0C  2- recNumber = 1  mode = REC\_ACC\_MODE\_ABSOLUTE  recOffset = 2  dataOffset = 0  dataLength = 4  updateRecord() | 1- No exception shall be thrown.  2- Shall throw uicc.access.UICC Exception with reason code OUT\_OF\_RECORD\_BOUNDARIES. |  |
| 15 | Updating with invalid mode  1- select EFLARU, fid = 6F0C  2- recNumber = 0  mode = 1  recOffset = 0  dataOffset = 0  dataLength = 4  updateRecord()  3- mode = 5  updateRecord() | 1 - No exception shall be thrown.  2 - Shall throw uicc.access.UICC Exception with reason code INVALID\_MODE.  3 - Shall throw uicc.access. UICC Exception with reason code INVALID\_MODE. |  |
| 16 | Updating Cyclic EF with invalid mode  1- select DFTEST, fid = 1111  2- select EFCARU, fid = 6F09  set record pointer to record nr 1  3- recNumber = 0  mode = REC\_ACC\_MODE\_NEXT  recOffset = 0  data[0:2] = '00'  dataOffset = 0  dataLength = 3  updateRecord()  4- recNumber = 0  mode = REC\_ACC\_MODE\_ABSOLUTE  updateRecord()  5- recNumber = 2  mode = REC\_ACC\_MODE\_ABSOLUTE  updateRecord() | 1 - No exception shall be thrown.  2 - No exception shall be thrown.  3 - Shall throw uicc.access. UICC Exception with reason code INVALID\_MODE.  4 - Shall throw uicc.access.UICC Exception with reason code INVALID\_MODE.  5 - Shall throw uicc.access. UICC Exception with reason code INVALID\_MODE. |  |
| 17 | data[] is null  data[] = null  dataOffset = 0  dataLength = 10  updateRecord() | Shall throw java.lang.NullPointerException. |  |
| 18 | dataOffset < 0  dataOffset = -1  dataLength = 10  updateRecord() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 19 | dataLength < 0  dataOffset = 0  dataLength = -1  updateRecord() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 20 | dataOffset + dataLength > data.length  dataOffset = 10  dataLength = 11  updateRecord() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 21 | EF is neither Cyclic nor Linear Fixed  1- select DFTEST, fid = 1111  2- select EFTARU, fid = 6F03  3- dataOffset = 0  dataLength = 4  updateRecord() | 1- No exception shall be thrown.  2- No exception shall be thrown.  3- Shall throw uii.access.UICC Exception with reason code COMMAND\_INCOMPATIBLE. |  |
| 22 | Access condition not fulfilled  1- select EFCNU, fid = 6F05  2- recOffset = 0  dataOffset = 0  dataLength = 1  mode = REC\_ACC\_MODE\_PREVIOUS  updateRecord()  3- fid = EFLNU  select()  4- recNumber = 1  mode = REC\_ACC\_MODE\_ CURRENT  recOffset = 0  dataOffset = 0  dataLength = 1  updateRecord() | 1- No exception shall be thrown.  2- Shall throw uicc.access.UICC Exception with reason code SECURITY\_STATUS\_NOT\_SATISFIED.  3- No exception shall be thrown.  4- Shall throw uicc.access.UICC Exception with reason code SECURITY\_STATUS\_NOT\_SATISFIED |  |
| 23 | EF is deactivated  1- select EFCNR, fid = 6F04  invalidate()  2- updateRecord()  3- activateFile()  4- restore the file content EFLARU, EFCARU | 1- No exception shall be thrown.  2- Shall throw uicc.access.UICCException with reason codeREF\_DATA\_INVALIDATED  3- No exception shall be thrown. |  |
|  | Restore the file content  1- restore the file content of EFLARU:  record 1 = 0x55,0x55,0x55,0x55  record 2 = 0xAA,0xAA,0xAA,0xAA  2- restore the file content of EFCARU:  record 1 = 0x55,0x55,0x55  record 2 = 0xAA,0xAA,0xAA |  |  |

#### 5.3.1.12 Method searchRecord

Test Area Reference: Api\_4\_Afv\_Sear.

##### 5.3.1.12.1 Conformance requirement

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

public short searchRecord(byte mode,

short recordNum,

short searchIndication,

byte[] patt,

short pattOffset,

short pattLength,

short[] response,

short respOffset,

short respLength)

throws java.lang.NullPointerException,

java.lang.ArrayIndexOutOfBoundsException,

UICCException

5.3.1.12.1.1 Normal execution

* CRRN1: Search a given pattern in byte array patt[] of a current linear fixed or cyclic EF.
* CRRN2: If the pattern is found, the number of each record is stored in byte array response[] and the total number of updated bytes in the array response[] buffer is returned.
* CRRN3: If the value of respLength is greater than the number of records found, the whole response is copied into the response buffer and the number of elements copied is returned by the method.
* CRRN4: If the value of respLength is smaller than the number of found patterns, the first record numbers are copied into the response array and the value of respLength is returned.
* CRRN5: If mode is SIMPLE\_SEARCH\_START\_FORWARD, the search starts at the given record number forward towards the end of the file.
* CRRN6: If mode is SIMPLE\_SEARCH \_START\_BACKWARD, the search starts at a given record number backward towards to the beginning of the file.
* CRRN7: If mode is ENHANCED\_SEARCH and SEARCH\_INDICATION\_START\_BACKWARD\_FROM\_PREVIOUS is set in searchIndication, the search is backward starting from previous record towards to the beginning of the file.
* CRRN8: If the mode is ENHANCED\_SEARCH and SEARCH\_INDICATION\_START\_BACKWARD\_FROM\_PREVIOUS\_GR is set in searchIndication, the search is backward starting at a given record from previous record towards to the beginning of the file.
* CRRN9: If the mode is ENHANCED\_SEARCH and SEARCH\_INDICATION\_START\_FORWARD\_FROM\_NEXT is set in searchIndication, the search is forward starting at the next record towards the end of the file.
* CRRN10: If the mode is ENHANCED\_SEARCH and SEARCH\_INDICATION\_FORWARD\_FROM\_NEXT\_GR is set in searchIndication, the search is forward starting at a given record number towards to the end of the file.
* CRRN11: If the mode is ENHANCED\_SEARCH, and bit 4 of the most significant byte of the searchIndication is not set, the search starts in the record from the offset (absolute position) given in the less significant byte of searchIndication.
* CRRN12: If the mode is ENHANCED\_SEARCH, and bit 4 of the most significant byte of the searchIndication is set, the search starts in the record after the first occurrence of the value contained in the less significant byte of searchIndication.
* CRRN13: If pattern given in patt[] is not found, the method returns 0.
* CRRN14: If one or more matches are found the record pointer shall be set to the first record where the search pattern was found.

5.3.1.12.1.2 Parameter errors

* CRRP1: If mode is not 4, 5 or 6, an instance of UICCException shall be thrown. The reason code shall be UICCException.INVALID\_MODE.
* CRRP2: If the pattern array patt is null, an instance of java.lang.NullPointerException shall be thrown.
* CRRP3: If the response array response is null, an instance of java.lang.NullPointerExceptino shall be thrown.
* CRRP4: If parameter pattOffset is negative, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
* CRRP5: If parameter pattLength is negative, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
* CRRP6: If parameter respOffset is negative, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
* CRRP7: If parameter respLength negative, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
* CRRP8: If parameter pattOffset plus pattLength are greater than the length of array patt, an instance of java.lang.ArrayIndexOutOfBoundsException shall be thrown.
* CRRP9: If parameter respOffset plus respLength are greater than the length of array response a ArrayIndexOutOfBoundsException shall be thrown.
* CRRP10: If parameter recordNum is negative, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD\_NOT\_FOUND.
* CRRP11: If parameter recordNum is greather than, the total number of records from the currently selected EF, an instance of UICCException shall be thrown. The reason code shall be UICCException.RECORD\_NOT\_FOUND.
* CRRP12: If pattLength is greater than record size of the currently selected EF an instance of UICCException shall be thrown. The reason code shall be UICCException.OUT\_OF\_FILE\_BOUNDARIES.

5.3.1.12.1.3 Context errors

* CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO\_EF\_SELECTED.
* CRRC2: If the currently selected EF is not linear fixed or cyclic, an instance of UICCException shall be thrown. The reason code shall be UICCException.COMMAND\_INCOMPATIBLE.
* CRRC3: If the calling applet does not fulfil the access condition, READ, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY\_STATUS\_NOT\_SATISFIED.
* CRRC4: If the currently selected EF is deactivated and the file status of the EF does not allow for reading a deactivated file, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF\_DATA\_INVALIDATED.
* CRRC5: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL\_ERROR.

##### 5.3.1.12.2 Test area files

Test Source: Test\_Api\_4\_Afv\_Sear.java.

Test Applet: Api\_4\_Afv\_Sear\_1.java.

Cap File: Api\_4\_Afv\_sear.cap.

##### 5.3.1.12.3 Test coverage

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

##### 5.3.1.12.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 1 | No EF selected  1- select DFTEST, fid=1111  2- searchRecord() | 2-shall throw uicc.access.UICC Exception with reason code NO\_EF\_SELECTED. |  |
| 2 | Fixed linear EF,  Simple mode search forward  1- select EFLSEA, fid=6F1A  2- mode = SIMPLE\_SEARCH\_START\_FORWARD  recordNum = 1  patt[]={0x10,0x03,0x04}  pattOffset = 0  pattLength = 1  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord()  3- Simple mode search forward  mode = SIMPLE\_SEARCH\_START\_FORWARD  recordNum = 2  patt[]={0x10,0x03,0x04}  pattOffset = 1  pattLength = 2  resp.length = 4  respOffset = 1  respLength = 3  searchRecord() | 2- no exception shall be thrown  Shall return 0.  response shall be:  response={0,0,0,0}  3- Shall return 2.  response shall be:  response={0,2,4,0} |  |
| 3 | Simple mode, search backward  1- mode = SIMPLE\_SEARCH\_START\_BACKWARD  recordNum = 1  patt[]={0x08,0x0A,0x0B}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0}  respOffset = 2  respLength = 2  searchRecord()  2-mode = SIMPLE\_SEARCH\_START\_BACKWARD  recordNum = 6  patt[]={0x08,0x09,0x0A,0x0B }  pattOffset = 1  pattLength = 2  response[] = {0,0,0,0}  respOffset = 1  respLength = 3  searchRecord() | 1- shall return 0.  response shall be:  response={0,0,0,0}  2- shall return 3.  response shall be:  response={0,4,3,1} |  |
| 4 | Enhanced Mode, search backward from previous record, start from an offset in record.  1- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_BACKWARD\_FROM\_PREVIOUS + 0x0009  recordNum = 0  patt[]={0x01,0x02,0x03,0x04}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord()  2- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_BACKWARD\_FROM\_PREVIOUS+0x0000  recordNum = 0  patt[]={0x0C,0x0D,0x0E,0x0F,0x01,0x02}  pattOffset = 0  pattLength = 5  response[] = {0,0,0,0}  respOffset = 2  respLength = 2  searchRecord() | 1- shall return 1,  response shall be:  resp={3,0,0,0}  2- shall return 1  response shall be:  response={0,0,2,0} |  |
| 5 | Enhanced Mode, search backward from previous record, start from a value in record.  1- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_BACKWARD\_FROM\_PREVIOUS + 0x0810  recordNum = 0  patt[]={0x01,0x02,0x03,0x04}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord()  2- perform 3 readRecord() in next mode to set current pointer to pointer 5  3- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_BACKWARD\_FROM\_PREVIOUS+0x080E  recordNum = 0  patt[]={0x01,0x02,0x03,0x04}  pattOffset = 3  pattLength = 1  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord() | 1- shall return 0,  response shall be:  resp={0,0,0,0}  3- shall return 2  response shall be:  response={4,2,0,0} |  |
| 6 | Enhanced Mode, search backward from previous given record, start from an offset in record.  1- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_BACKWARD\_FROM\_PREVIOUS\_GR + 0x0000  recordNum = 1  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 1  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord()  2- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_BACKWARD\_FROM\_PREVIOUS\_GR + 0x0004  recordNum = 6  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord() | 1- shall return 1,  response shall be:  resp={1,0,0,0}  2- shall return 4  response shall be:  response={5,4,3,2} |  |
| 7 | Enhanced Mode, search backward from previous given record, start from a value in record.  1- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_BACKWARD\_FROM\_PREVIOUS\_GR + 0x080D  recordNum = 1  patt[]={0x0E,0x0E,0x0E}  pattOffset = 1  pattLength = 1  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord()  2- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_BACKWARD\_FROM\_PREVIOUS\_GR + 0x0800  recordNum = 6  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord() | 1- shall return 1,  response shall be:  resp={1,0,0,0}  2- shall return 0  response shall be:  response={0,0,0,0} |  |
| 8 | Enhanced Mode, search forward from next record, start from an offset in record.  1- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_FORWARD\_FROM\_NEXT + 0x0003  recordNum = 0  patt[]={0x00,0x0A,0x0B}  pattOffset = 1  pattLength = 2  response[] = {0,0,0}  respOffset = 2  respLength = 2  searchRecord()  2- Perform readRecord() in previous mode  3- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_FORWARD\_FROM\_NEXT + 0x0003  recordNum = 0  patt[]={0x00,0x0A,0x0B}  pattOffset = 1  pattLength = 2  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord() | 1- shall return 2  response shall be:  resp={0,0,3,4}  3- shall return 1  response shall be:  response={4,0,0,0} |  |
| 9 | Enhanced Mode, search forward from next record, start from a value in record.  1- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_FORWARD\_FROM\_NEXT + 0x0804  recordNum = 0  patt[]={0x01,0x02,0x03}  pattOffset = 1  pattLength = 2  response[] = {0,0,0,0}  respOffset = 2  respLength = 2  searchRecord()  2- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_FORWARD\_FROM\_NEXT + 0x0801  recordNum = 0  patt[]={0x01,0x02,0x03}  pattOffset = 2  pattLength = 1  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord() | 1- shall return 0,  response shall be:  resp={0,0,0,0}  2- shall return 2  response shall be:  response={5,6,0,0} |  |
| 10 | Enhanced Mode, search forward from next given record, start from an offset in record.  1- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_FORWARD\_FROM\_NEXT\_GR + 0x0007  recordNum = 1  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0}  respOffset = 1  respLength = 3  searchRecord()  2- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_FORWARD\_FROM\_NEXT\_GR + 0x000C  recordNum = 3  patt[]={0x03,0x02,0x01}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord() | 1- shall return 3,  response shall be:  resp={0,3,4,5}  2- shall return 1  response shall be:  response={6,0,0,0} |  |
| 11 | Enhanced Mode, search forward from next given record, start from a value in record.  1- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_FORWARD\_FROM\_NEXT\_GR + 0x080D  recordNum = 5  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord()  2- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_FORWARD\_FROM\_NEXT\_GR + 0x080C  recordNum = 5  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord() | 1- shall return 0,  response shall be:  resp={0,0,0,0}  2- shall return 1  response shall be:  response={5,0,0,0} |  |
| 12 | Simple mode, total number of found patterns exceed response[]  1- mode = SIMPLE\_SEARCH\_START\_FORWARD  recordNum = 1  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0}  respOffset = 0  respLength = 4  searchRecord()  2- mode = SIMPLE\_SEARCH\_START\_FORWARD  recordNum = 1  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0,0}  respOffset = 0  respLength = 4  searchRecord() | 1- shall return 4  response shall be:  response={1,2,3,4}  2- shall return 4  response shall be:  response={1,2,3,4,0} |  |
| 13 | Invalid mode  mode = 0x14 (simple search forward with SFI)  searchIndication= 0  recordNum = 2  patt[]={0x01,0x02,0x03}  pattOffset = 1  pattLength = 2  response[] = {0,0,0,0,0}  respOffset = 2  respLength = 2  searchRecord() | shall throw an uicc.access.UICC Exception with reason code INVALID\_MODE. |  |
| 14 | Pattern array is null  mode = SIMPLE\_SEARCH\_START\_FORWARD  searchIndication= 0  recordNum = 0  patt[]= null  pattOffset = 1  pattLength = 2  response[] = {0,0,0,0,0}  respOffset = 2  respLength = 2  searchRecord() | shall throw an java.lang.NullPointerException. |  |
| 15 | Response array is null  mode = SIMPLE\_SEARCH\_START\_FORWARD  searchIndication= 0  recordNum = 0  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 1  response[] = null  respOffset = 0  respLength = 5  searchRecord() | shall throw an instance of java.lang.NullPointerException. |  |
| 16 | pattOffset<0  mode = SIMPLE\_SEARCH\_START\_FORWARD  searchIndication= 0  recordNum = 0  patt[]={0x01,0x02,0x03}  pattOffset = -1  pattLength = 1  response[] = {0,0,0,0,0}  respOffset = 0  respLength = 5  searchRecord() | shall throw an instance of java.lang.ArrayIndexOutOfBoundsException. |  |
| 17 | pattLength<0  mode = SIMPLE\_SEARCH\_START\_FORWARD  searchIndication= 0  recordNum = 0  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = -1  response[] = {0,0,0,0,0}  respOffset = 0  respLength = 5  searchRecord() | shall throw an instance of java.lang.ArrayIndexOutOfBoundsException. |  |
| 18 | respOffset <0  mode = SIMPLE\_SEARCH\_START\_FORWARD  searchIndication= 0  recordNum = 0  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 1  response[] = {0,0,0,0,0}  respOffset = -1  respLength = 5  searchRecord() | shall throw an instance of java.lang.ArrayIndexOutOfBoundsException. |  |
| 29 | respLength <0  mode = SIMPLE\_SEARCH\_START\_FORWARD  searchIndication= 0  recordNum = 0  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 1  response[] = {0,0,0,0,0}  respOffset = 0  respLength = -1  searchRecord() | shall throw an instance of java.lang.ArrayIndexOutOfBoundsException. |  |
| 20 | PattOffset + pattLength > patt[]  mode = SIMPLE\_SEARCH\_START\_FORWARD  searchIndication= 0  recordNum = 0  patt[]={0x01,0x02,0x03}  pattOffset = 2  pattLength = 2  response[] = {0,0,0,0,0}  respOffset = 1  respLength = 5  searchRecord() | shall throw an instance of java.lang.ArrayIndexOutOfBoundsException. |  |
| 21 | RespOffset + respLength > response[]  mode = SIMPLE\_SEARCH\_START\_FORWARD  searchIndication= 0  recordNum = 0  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 1  response[] = {0,0,0,0,0}  respOffset = 3  respLength = 3  searchRecord() | shall throw an instance of java.lang.ArrayIndexOutOfBoundsException. |  |
| 22 | recordNum < 0  mode = SIMPLE\_SEARCH\_START\_FORWARD  searchIndication= 0  recordNum = -1  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 1  response[] = {0,0,0,0,0}  respOffset = 0  respLength = 5  searchRecord() | shall throw an uicc.access.UICC Exception with reason code RECORD\_NOT\_FOUND |  |
| 23 | RecordNum > total number of file records  1- mode = SIMPLE\_SEARCH\_START\_FORWARD  searchIndication= 0  recordNum = 7  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 1  response[] = {0,0,0,0,0}  respOffset = 0  respLength = 5  searchRecord()  2- mode = SIMPLE\_SEARCH\_START\_FORWARD  searchIndication= 0  recordNum = 0  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 1  response[] = {0,0,0,0,0}  respOffset = 0  respLength = 5  searchRecord() | 1- shall throw an uicc.access.UICC Exception with reason code RECORD\_NOT\_FOUND  2- shall throw an uicc.access.UICC Exception with reason code RECORD\_NOT\_FOUND |  |
| 24 | pattlength > record length  1- mode = SIMPLE\_SEARCH\_START\_FORWARD  searchIndication= 0  recordNum = 3  patt[16]={0x55,0x55,…,0x55}  pattOffset = 0  pattLength = 16  response[] = {0,0,0,0,0}  respOffset = 0  respLength = 5  searchRecord()  2- mode = ENHANCED\_SEARCH  searchIndication= SEARCH\_INDICATION\_FORWARD\_FROM\_NEXT\_GR + 0x000E  recordNum = 3  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0,0}  respOffset = 0  respLength = 5  searchRecord() | 1- shall throw an uicc.access.UICCException with reason code OUT\_OF\_FILE\_BOUNDARIES.  2- shall throw an uicc.access.UICCException with reason code OUT\_OF\_FILE\_BOUNDARIES. |  |
| 25 | Wrong file structure  1- select EFTDAC, fid=6F0F  2- searchRecord() | 2- shall throw an uicc.access.UICCException with reason code COMMAND\_INCOMPATIBLE |  |
| 26 | Security status not satisfied  1- select EFLNR, fid=6F0A  2- searchRecord() | 2- shall throw an uicc.access.UICCException with reason code SECURITY\_STATUS\_NOT\_SATISFIED |  |
| 27 | File deactivated  1- select EFLARU, fid=6F10  2- deactivateFile EFLARU  3- searchRecord()  4- activateFile() | 3- shall throw an uicc.access.UICCException with reason code DATA\_INVALIDATED |  |
| 28 | Cyclic EF, Simple mode search forward  1- select EFCSEA, fid=6F1B  2- mode = SIMPLE\_SEARCH\_START\_FORWARD  recordNum = 1  patt[]={0x10,0x03,0x04}  pattOffset = 0  pattLength = 1  response[] = {0,0,0,0,0}  respOffset = 0  respLength = 5  searchRecord()  3- mode = SIMPLE\_SEARCH\_START\_FORWARD  recordNum = 2  patt[]={0x10,0x03,0x04}  pattOffset = 1  pattLength = 2  response[] = {0,0,0,0,0}  respOffset = 2  respLength = 3  searchRecord()  4- updateRecord() in previous mode with value {0x03,0x02,0x01,0x03,0x02,0x01,0x03,0x02,0x01,0x03,0x02,0x01,0x03,0x02,0x01} (new record 1 is set to previous record 6)  5- mode = SIMPLE\_SEARCH\_START\_FORWARD  recordNum = 2  patt[]={0x10,0x03,0x04}  pattOffset = 1  pattLength = 2  response[] = {0,0,0,0,0}  respOffset = 2  respLength = 3  searchRecord() | 2- shall return 0  response shall be:  response={0,0,0,0,0}  3- Shall return 3.  response shall be:  response={0,0,2,4,1}  5- Shall return 3.  response shall be:  response={0,0,2,3,5} |  |
| 29 | Cyclic EF, Simple mode search backward  mode = SIMPLE\_SEARCH\_START\_BACKWARD  recordNum = 3  patt[]={0x10,0x03,0x04}  pattOffset = 1  pattLength = 2  response[] = {0,0,0,0,0}  respOffset = 1  respLength = 4  searchRecord() | shall return 3  response shall be:  response={0,3,2,5,0} |  |
| 30 | Cyclic EF, Enhanced mode, search forward from next record, start from an offset in record  mode = ENHANCED\_MODE  searchIndication= SEARCH\_INDICATION\_START\_FORWARD\_FROM\_NEXT + 0x0009  recordNum = 0  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0,0}  respOffset = 2  respLength = 3  searchRecord() | shall return 3  response shall be:  response={0,0,4,5,6} |  |
| 31 | Cyclic EF, Enhanced mode, search forward from next record, start from a value in record  mode = ENHANCED\_MODE  searchIndication= SEARCH\_INDICATION\_START\_FORWARD\_FROM\_NEXT + 0x0810  recordNum = 0  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 3  response[] = {0,0,0,0,0}  respOffset = 2  respLength = 3  searchRecord() | shall return 0  response shall be:  response={0,0,0,0,0} |  |
| 32 | Cyclic EF, Enhanced mode, search forward from next given record, start from an offset in record  mode = ENHANCED\_MODE  searchIndication= SEARCH\_INDICATION\_START\_FORWARD\_FROM\_NEXT\_GR + 0x0005  recordNum = 3  patt[]={0x01,0x02,0x03}  pattOffset = 0  pattLength = 1  response[] = {0,0,0,0,0}  respOffset = 0  respLength = 5  searchRecord() | shall return 5  response shall be:  response={3,4,5,6,1} |  |
| 33 | Cyclic EF, Enhanced mode, search forward from next given record, start from a value in record  1- mode = ENHANCED\_MODE  searchIndication= SEARCH\_INDICATION\_START\_FORWARD\_FROM\_NEXT\_GR + 0x0805  recordNum = 6  patt[]={0x0E,0x0F,0x00}  pattOffset = 0  pattLength = 2  response[] = {0,0,0,0,0}  respOffset = 0  respLength = 5  searchRecord()  2- Restore EF initial state (record 1 shall be assigned to the record that content is {0x01,0x02,0x03,0x04,0x05,0x06,0x07,0x08,0x09,0x0A, 0x0B,0x0C,0x0D,0xE,0x0F}) using 5 updateRecord() in previous mode | 1- shall return 2  response shall be:  response={2,4,0,0,0} |  |
| 34 | Cyclic EF, Enhanced mode, search backward from previous record, start from an offset in record  1- Set current record pointer to record 6 using 5 readRecord() in next mode  2- mode = ENHANCED\_MODE  searchIndication= SEARCH\_INDICATION\_START\_BACKWARD\_FROM\_PREVIOUS + 0x0003  recordNum = 0  patt[]={0x02,0x01,0x00}  pattOffset = 0  pattLength = 2  response[] = {0,0,0,0,0}  respOffset = 3  respLength = 2  searchRecord() | 2- shall return 1  response shall be:  response={0,0,0,6,0} |  |
| 35 | Cyclic EF, Enhanced mode, search backward from previous record, start from a value in record  mode = ENHANCED\_MODE  searchIndication= SEARCH\_INDICATION\_START\_BACKWARD\_FROM\_PREVIOUS + 0x0801  recordNum = 0  patt[]={0x01,0x02,0x03}  pattOffset = 1  pattLength = 2  response[] = {0,0,0,0,0}  respOffset = 0  respLength = 5  searchRecord() | shall return 5  response shall be:  response={5,4,3,2,1} |  |
| 36 | Cyclic EF, Enhanced mode, search backward from given record, start from an offset in record  mode = ENHANCED\_MODE  searchIndication= SEARCH\_INDICATION\_START\_BACKWARD\_FROM\_PREVIOUS\_GR + 0x0003  recordNum = 5  patt[]={0x02,0x01,0x00}  pattOffset = 0  pattLength = 2  response[] = {0,0,0,0,0}  respOffset = 3  respLength = 2  searchRecord() | shall return 1  response shall be:  response={0,0,0,6,0} |  |
| 37 | Cyclic EF, Enhanced mode, search backward from given record, start from a value in record  mode = ENHANCED\_MODE  searchIndication= SEARCH\_INDICATION\_START\_BACKWARD\_FROM\_PREVIOUS + 0x0801  recordNum = 3  patt[]={0x01,0x02,0x03}  pattOffset = 1  pattLength = 2  response[] = {0,0,0,0,0}  respOffset = 0  respLength = 5  searchRecord() | shall return 5  response shall be:  response={3,2,1,5,4} |  |

#### 5.3.1.13 Method increase

Test Area Reference: Api\_4\_Afv\_Incr.

##### 5.3.1.13.1 Conformance requirement

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

public short increase(byte[] incr,

short incrOffset,

short incrLength,

byte[] resp,

short respOffset)

throws java.lang.NullPointerException,

java.lang.ArrayIndexOutOfBoundsException,

UICCException

5.3.1.13.1.1 Normal execution

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

5.3.1.13.1.2 Parameter errors

* CRRP1: If the array incr is null, an instance of NullPointerException shall be thrown.
* CRRP2: If the array resp is null, an instance of NullPointerException shall be thrown.
* CRRP3: If incrOffset is negative, an instance of ArrayIndexOutOfBoundsException shall be thrown.
* CRRP4: If incrLength is negative, an instance of ArrayIndexOutOfBoundsException shall be thrown.
* CRRP5: If respOffset is negative, an instance of ArrayIndexOutOfBoundsExceptoin shall be thrown.
* CRRP6: If incrOffset plus incrLength, is greater than the length of array incr, an instance of ArrayIndexOutOfBoundsException shall be thrown and no increase is performed.
* CRRP7: If respOffset is greater than the length of array resp, an instance of ArrayIndexOutOfboundsException shall be thrown.
* CRRP8: If the result of the addition is greater than the maximum value of the record (represented by all bytes set to 'FF'), an instance of UICCException shall be thrown. The reason code shall be UICCException.MAX\_VALUE\_REACHED.
* CRRP9: If incrLength is greater than 127, and exception shall be thrown.

5.3.1.13.1.3 Context errors

* CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO\_EF\_SELECTED.
* CRRC2: If the method call causes a memory problem (e.g. memory access error), an instance of UICCException. shall be thrown. The reason code shall be UICCException.MEMORY\_PROBLEM.
* CRRC3: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL\_ERROR.
* CRRC4: If file is not a cyclic one, an instance of the UICCException shall be thrown. The reason code shall be UICCException.COMMAND\_INCOMPATIBLE.
* CRRC5: If the calling applet does not fulfil the access condition, INCREASE, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY\_STATUS\_NOTSATISFIED.
* CRRC6: If the currently selected EF is invalidated, an instance of UICCException shall be thrown. The reason code shall be UICCException.REF\_DATA\_INVALIDATED.
* CRRC7: If the currently selected cyclic EF has no record, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO\_RECORD\_FOUND.

##### 5.3.1.13.2 Test areafiles

Test Source: Test\_Api\_4\_Afv\_Incr.java.

Test Applet: Api\_4\_Afv\_Incr\_1.java.

Cap File: Api\_4\_Afv\_incr.cap.

##### 5.3.1.13.3 Test coverage

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

##### 5.3.1.13.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 1 | No EF selected  1- select DFTEST fid=1111  2- byte[] incr = new byte[4]  byte[] resp = new byte[4]  incrOffset = 0  incrLength = 2  respOffset = 0  increase() | 2- An UICCException.NO\_EF\_SELECTED should be thrown |  |
| 2 | increase , verify response  1- select EFCARU, fid=6F09  set the record pointer with readRecord() in PREVIOUS mode  2-//Set both record to 00 00 00  mode = REC\_ACC\_MODE\_PREVIOUS  data[] = {0x00,0x00,0x00}  recOffset = 0  dataOffset = 0  dataLength = 3  updateRecord() //update Record 1  updateRecord() //update Record 2  3- incr[] = {0x00,0x00,0x01}  incrOffset = 0  incrLength = 3  resp.length = 4  respOffset = 0  ret = 3  increase() | 3- resp[] = {0x00,0x00,0x01,0x00} |  |
| 3 | increase, verify file  1- incr[]={0x00,0x00,0x00,0x02}  incrOffset = 1  incrLength = 3  resp.length = 4  respOffset = 1  increase()  2- resp[] = {0x00,0x00,0x00,0x00}  recNumber = 0  mode = REC\_ACC\_MODE\_ABSOLUTE\_CURRENT  recOffset = 0  resp.length =4  respOffset = 0  respLength = 3  readRecord() | 1- resp[] = {0x00,0x00,0x00,0x03}  2- resp[] = {0x00,0x00,0x03,0x00} |  |
| 4 | incr[] is null  incr[] = null  incrOffset = 0  incrLength = 1  resp.length = 4  respOffset = 0  increase() | Shall throw java.lang.NullPointerException. |  |
| 5 | incrLength< 0  incr.length = 4  incrOffset = 0  incrLength = -1  resp.length = 4  respOffset = 0  increase() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 6 | incrOffset < 0  incr.length = 4  incrOffset = -1  incrLength = 1  resp.length = 4  respOffset = 0  increase() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 7 | IncrOffset + incrLength > incr.length  incr.length = 4  incrOffset = 1  incLength = 4resp.length = 4  respOffset = 0  increase() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 8 | Reach Maximum Value  1- incr[0:3] = 0xFF  incrOffset = 0  incrLength = 3  resp.length = 4  respOffset = 0  increase()  2- //Set both record to FF FF FF  mode = REC\_ACC\_MODE\_PREVIOUS  data[] = {0xFF,0xFF,0xFF}  recOffset = 0  dataOffset = 0  dataLength = 3  updateRecord() //update Record 1  updateRecord() //update Record 2  3- incr[] = {0x00,0x00,0x01}  incrOffset = 0  incrLength = 3  resp.length = 4  respOffset = 0  increase() | 1- Shall throw uicc.access.UICCException with reason code MAX\_VALUE\_REACHED.  2- Shall throw uicc.access.UICCException with reason code MAX\_VALUE\_REACHED. |  |
| 9 | resp[] is null  incr.length = 4  incrOffset = 0  incrLength = 1  resp[] = null  respOffset = 0  increase() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 10 | respOffset < 0  incr.length = 4  incrOffset = 0  incrLength = 1  resp.length = 4  respOffset = -1  increase() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 11 | respOffset + recordLength > resp.length  incr.length = 4  incrOffset = 0  incrLength = 3  resp.length = 3  respOffset = 2  increase() | Shall throw  java.lang. ArrayIndexOutOfBoundsException. |  |
| 12 | EF is not Cyclic  1- select EFTARU fid= 6F03  2- incr.length= 3  incrOffset = 0  incrLength = 3  resp.length = 3  respOffset = 0  increase()  3 - select EFLARU, fid=6F0C  4 - incr.length= 3  incrOffset = 0  incrLength = 3  resp.length = 3  respOffset = 0  increase() | 2 - Shall throw uicc.access.UICCException with reason code COMMAND\_INCOMPATIBLE.  4 - Shall throw uicc.access.UICCException with reason code COMMAND\_INCOMPATIBLE. |  |
| 13 | Access condition not fulfilled  1- select EFCNIC, fid=6F06  2- incr.length= 3  incrOffset = 0  incrLength = 3  resp.length = 3  respOffset = 0  increase() | 2 - Shall throw uicc.access.UICCException with reason code SECURITY\_STATUS\_NOT\_SATISFIED. |  |
| 14 | EF is invalidated  1-select EFCARU, fid=6F09  2 - invalidate()  3 - incr.length= 3  incrOffset = 0  incrLength = 3  resp.length = 3  respOffset = 0  increase()  4 - rehabilitate()  5- Restore initial content of EFCARU | 3 - Shall throw uicc.access.UICCException with reason code REF\_DATA\_INVALIDATED |  |
| 15 | incrLength out of range  1- Create an EF Cyclic with 1 record of 0x7F length, fid=0x2C7F  2- Select EF Cyclic, fid=0x2C7F  3- Set record to following value rec[0] = 0; rec[1..126] = 0xFF with an update record.  4- incr.length=128  incrOffset = 1  incrLength = 127  resp.length = 255  respOffset = 0  Incr[] initialized to = {0x00,… ,0x00,0x01}  respOffset = 0  ret = 0x7F  increase()  5- incr.length=128  incrOffset = 0  incrLength = 128  resp.length = 255  respOffset = 0  Incr[] initialized to 0  respOffset = 0  increase()  6- Delete EF Cyclic with fid=0x2C7F | 4- resp[0..126] = {0x01,0x00,0x00,…,0x00}  5- Shall throw an exception |  |

#### 5.3.1.14 Method deactivateFile

Test Area Reference: Api\_4\_Afv\_Dacf.

##### 5.3.1.14.1 Conformance requirement

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

public void deactivateFile()

throws UICCException

5.3.1.14.1.1 Normal execution

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

5.3.1.14.1.2 Parameter errors

No requirements.

5.3.1.14.1.3 Context errors

* CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO\_EF\_SELECTED.
* CRRC2: If the calling applet does not fulfil the access condition, activate, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY\_STATUS\_NOT\_SATISFIED.
* CCRC3: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL\_ERROR.

##### 5.3.1.14.2 Test area files

Test Sourec: Test\_Api\_4\_Afv\_Dacf.java.

Test Applet: Api\_4\_Afv\_Dacf\_1.java.

Cap File: Api\_4\_Afv\_dacf.cap.

##### 5.3.1.14.3 Test coverage

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

##### 5.3.1.14.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 0 | Test applet is installed with no access right on Application Pin2 |  |  |
| 1 | No EF is selected  1- select DFTEST fid=1111  2- call deactivateFile() | 2- An UICCException NO\_EF\_SELECTED is thrown |  |
| 2 | Deactivate activated File  0- Select root directory  1- Select EFUICC fid=2FF0  2- ReadBinary EFUICC  3- Deactivate EFUICC  4- ReadBinary EFUICC | 2- No Exception shall be thrown  4- UICCException.REF\_DATA\_INVALIDATED is thrown |  |
| 3 | Deactivate deactivated File  1- deactivateFile EFUICC  2- activateFile EFUICC | 1- No Exception shall be thrown |  |
| 4 | Access condition not fulfilled  1- select DFTEST fid=1111  2- select EFLADA fid=6F15  3- deactivateFile EFLADA | 3- An UICCException SECURITY\_STATUS\_NOT\_SATISFIED is thrown |  |

#### 5.3.1.15 Method activateFile

Test Area Reference: Api\_4\_Afv\_Actf.

##### 5.3.1.15.1 Conformance requirement

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

public void activateFile()

throws UICCException

5.3.1.15.1.1 Normal execution

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

5.3.1.15.1.2 Parameter errors

No requirements.

5.3.1.15.1.3 Context errors

* CRRC1: If the calling applet has currently no EF selected, an instance of UICCException shall be thrown. The reason code shall be UICCException.NO\_EF\_SELECTED.
* CRRC2: If the calling applet does not fulfil the access condition, activate, to perform this function, an instance of UICCException shall be thrown. The reason code shall be UICCException.SECURITY\_STATUS\_NOT\_SATISFIED.
* CRRC3: If the method call causes an error to occur that is not expected and thus not handled, an instance of UICCException shall be thrown. The reason code shall be UICCException.INTERNAL\_ERROR.

##### 5.3.1.15.2 Test area files

Test Source: Test\_Api\_4\_Afv\_Actf.java.

Test Applet: Api\_4\_Afv\_Actf\_1.java.

Cap File: Api\_4\_Afv\_actf.cap.

##### 5.3.1.15.3 Test coverage

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

##### 5.3.1.15.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 0 | Test applet is installed with no access right on Application Pin2 |  |  |
| 1 | No EF is selected  1- Select DFTEST fid=1111  2- Call activateFile() | 2- A UICCException NO\_EF\_SELECTED is thrown |  |
| 2 | Activate deactivated File  0- Select Root directory  1- Select EFUICC fid=2FF0  2- ReadBinary EFUICC  3- Deactivate EFUICC  4- ReadBinary EFUICC  5 -ActivateFile EFUICC  6- ReadBinary EFUICC | 2- No Exception shall be thrown  4- UICCException.REF\_DATA\_INVALIDATED is thrown  6- No Exception shall be thrown |  |
| 3 | Activate activated File  ActiveFile EFUICC | No Exception shall be thrown |  |
| 4 | Access condition not fulfilled  1- Select DFTEST fid=1111  2- Select EFLADA fid=6F15  3- ActivateFile EFLADA | 3- A UICCException SECURITY\_STATUS\_NOT\_SATISFIED is thrown |  |

### 5.3.2 Class AdminFileViewBuilder

#### 5.3.2.1 Method getTheUICCAdminFileView

Test Area Reference: Api\_4\_Afb\_Gtafb.

##### 5.3.2.1.1 Conformance requirement

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

public static AdminFileView getTheUICCAdminFileView(byte event)

throws javacard.framework.SystemException

5.3.2.1.1.1 Normal execution

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

5.3.2.1.1.2 Parameter errors

* CRRP1: If event is not one of the following values JCSystem.NOT\_A\_TRANSIENT\_OBJECT, JCSystem.CLEAR\_ON\_DESELECT, or JCSystem.CLEAR\_ON\_RESET a SystemException is thrown. The value of the SystemException shall be SystemException.ILLEGAL\_VALUE.

5.3.2.1.1.3 Context errors

* CRRC1: If event is JCSystem.CLEAR\_ON\_RESET or JCSystem.CLEAR\_ON\_DESELECT and not enough transient memory space is available a SystemException shall be thrown with reason code SystemException.NO\_TRANSIENT\_SPACE.
* CRRC2: If event is JCSystem.CLEAR\_ON\_DESELECT and the applet is not a currently selected applet a SystemException with reason code SystemException.ILLEGAL\_TRANSIENT shall be thrown.

##### 5.3.2.1.2 Test area files

Test Source: Test\_ Api\_4\_Afb\_Gtafb.java.

Test Applet: Api\_4\_Afb\_Gtafb\_1.java.

Cap File: Api\_4\_Afb\_Gtafb.cap.

##### 5.3.2.1.3 Test coverage

|  |  |
| --- | --- |
| CRR number | Test case number |
| N1 | 2 |
| N2 | 1 |
| N3 | 2 |
| N4 | 2 |
| N5 | 3 |
| P1 | 7 |
| C1 | 5, 6  Testable only if available transient space is lower than 32767 |
| C2 | 4 |

##### 5.3.2.1.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 1 | Method returns null  Install Applet1 with full access rights on the UICC file system  Invoke the method getTheUICCView before the javacard.framework.Applet.register(..) method invocation | returns null |  |
| 2 | Normal execution  1- Envelope menu selection is sent to the UICC  Invoke the method getTheUICCView() with the event JCSystem.NOT\_A\_TRANSIENT\_OBJECT  and stores the result in a class variable FV1  Applet1 calls status() command  Select DFTest using FV1  Select EFTARU using FV1  Read first 3 bytes using FV1  Reset  Terminal profile  2- Envelope menu selection is sent to the UICC  Applet1 calls FV1.status() command  Applet1 calls FV1.select(0x7FFF)  Invoke the method getTheUICCView() with the event JCSystem.CLEAR\_ON\_RESET and stores the result in a class variable FV2  Applet1 calls FV2.status() command  Select DFTest using FV2  Select EFTARU using FV2  Read first 3 bytes using FV2  Reset  Terminal profile  4 - Envelope menu selection is sent to the UICC  Applet1 calls status() command  Applet1 calls select(0x7FFF)  5- Select the Applet by AID  Invoke the method in the method process() getTheUICCView() with the event  JCSystem.CLEAR\_ON\_DESELECT and stores the result in a class variable  Applet1 calls status() command  Select DFTest using FV3  Select EFTARU using FV3  Read first 3 bytes using FV3  Select ADF2 by AID  6- Select the Applet by AID  Applet1 calls status() command  Applet1 calls select(0x7FFF) | 1- Applet1 is triggered  No exception shall be thrown  Current selected DF is the MF  Expected value is {FF FF FF}  2- Applet1 is triggered  Current selected DF is DFTest  [UICCException](file:///e:\Documenti\Standard%20&%20Specifications\ETSI\TSStandard\ts_102241v060700p0\102241_Annex_A_HTML\uicc\access\UICCException.html).FILE\_NOT\_FOUND is thrown  No exception shall be thrown  Current selected DF the MF  Expected value is {FF FF FF}  4- Applet1 is triggered  Current selected DF is the MF  [UICCException](file:///e:\Documenti\Standard%20&%20Specifications\ETSI\TSStandard\ts_102241v060700p0\102241_Annex_A_HTML\uicc\access\UICCException.html).FILE\_NOT\_FOUND is thrown  5- Applet1 is selected  No exception shall be thrown  Current selected DF the MF  Expected value is {FF FF FF}  6- Applet1 is selected  Current selected DF is the MF  [UICCException](file:///e:\Documenti\Standard%20&%20Specifications\ETSI\TSStandard\ts_102241v060700p0\102241_Annex_A_HTML\uicc\access\UICCException.html).FILE\_NOT\_FOUND is thrown |  |
| 3 | Fileview context independency  1- Envelope menu selection is sent to the UICC  2- Check that previous fileviews are different (FV1 != FV2 != FV3)  3- Select DFTest/EFLARU using FV1  4- Select DFTest/EFCARU using FV2  5- Select DFTest/EFCARU using FV3  6- Read record number 1 using FV1 (in absolute mode)  7- Read record number 2 using FV2 (in absolute mode) | 1- Applet1 is triggered  3- No exception shall be thrown  4- No exception shall be thrown  5- An exception is thrown  6- Expected value is “55 55 55 55”  7- Expected value is “AA AA AA” |  |
| 4 | ILLEGAL\_TRANSIENT SystemException  1- Envelope menu selection is sent to the UICC  2- Applet1 calls getTheUICCView() method with the event JCSystem. CLEAR\_ON\_DESELECT | 1- Applet1 is triggered  2- SystemException. ILLEGAL\_TRANSIENT is thrown |  |
| 5 | NO\_TRANSIENT\_SPACESystemException with CLEAR\_ON\_RESET Fileview object  1- Get the available transient memory space using method length=JCSystem.[getAvailableMemory](file:///E:\Documenti\Standard%20%26%20Specifications\JavaCard_221\specs\api\html\javacard\framework\JCSystem.html#getAvailableMemory%28byte%29)(MEMORY\_TYPE\_TRANSIENT\_RESET)  2- If length < 32767, (test case could be performed)  2.1- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR\_ON\_RESET)  2.2- Applet calls method getTheUICCView() with event  JCSystem.CLEAR\_ON\_RESET  4- Reset | 1- No exception shall be thrown  2.1- No exception shall be thrown  2.2- SystemException. NO\_TRANSIENT\_SPACE is thrown |  |
| 6 | NO\_TRANSIENT\_SPACE SystemException with CLEAR\_ON\_DESELECT Fileview object  1- Select the Applet by AID  2- Get the available transient memory space using method length=JCSystem.[getAvailableMemory](file:///E:\Documenti\Standard%20%26%20Specifications\JavaCard_221\specs\api\html\javacard\framework\JCSystem.html#getAvailableMemory%28byte%29)(MEMORY\_TYPE\_TRANSIENT\_DESELECT)  3- If length < 32767, (test case could be performed)  3.1- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR\_ON\_DESELECT)  3.2- Applet calls method getTheUICCView() with event  JCSystem.CLEAR\_ON\_DESELECT  4- Reset | 1- Applet1 is selected  2- No exception shall be thrown  3.1- No exception shall be thrown  3.2- SystemException. NO\_TRANSIENT\_SPACE is thrown |  |
| 7 | ILLEGAL\_VALUE SystemException  Invoke the method getTheUICCView() with every event codes except 0,1,2 | 1- SystemException.ILLEGAL\_VALUE is thrown |  |

#### 5.3.2.2 Method getTheAdminFileView(javacard.framework.AID aid, byte event)

Test Area Reference: Api\_4\_Afb\_Gtafob.

##### 5.3.2.2.1 Conformance requirement:

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

public static AdminFileView getTheAdminFileView(javacard.framework.AID aid, byte event)

throws NullPointerException,

javacard.framework.SystemException

5.3.2.2.1.1 Normal execution

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

5.3.2.2.1.2 Parameter errors

* CRRP1: If event is not one of the following values JCSystem.NOT\_A\_TRANSIENT\_OBJECT, JCSystem.CLEAR\_ON\_DESELECT, or JCSystem.CLEAR\_ON\_RESET a SystemException is thrown. The value of the SystemException shall be SystemException.ILLEGAL\_VALUE.
* CRRP2: If the AID is null a NullPointerException shall be thrown.

5.3.2.2.1.3 Context errors

* CRRC1: If event is JCSystem.CLEAR\_ON\_RESET or JCSystem.CLEAR\_ON\_DESELECT and not enough transient memory space is available a SystemException shall be thrown with reason code SystemException.NO\_TRANSIENT\_SPACE.
* CRRC2: If event is JCSystem.CLEAR\_ON\_DESELECT and the applet is not a currently selected applet a SystemException with reason code SystemException.ILLEGAL\_TRANSIENT shall be thrown.

##### 5.3.2.2.2 Test area files

Test Source: Test\_ Api\_4\_Afb\_Gtafob.java.

Test Applet: Api\_4\_Afb\_Gtafob.java.

Cap File: Api\_4\_Afb\_Gtafob.cap.

##### 5.3.2.2.3 Test coverage

|  |  |
| --- | --- |
| CRR number | Test case number |
| N1 | 1 to 3 |
| N2 | 1 |
| N3 | 1 |
| N4 | 2 |
| N5 | 3 |
| N6 | 2 |
| P1 | 7 |
| P2 | 8 |
| C1 | 5, 6  Testable only if available transient space is lower than 32767 |
| C2 | 4 |

##### 5.3.2.2.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 1 | Method returns null  1- Install Applet1 with full access rights on the UICC file system  2- Invoke the method getTheAdminFileView before the javacard.framework.Applet.register(..) method invocation  3- Envelope menu selection is sent to the UICC  4- Invoke the method getTheAdminFileView() with AID = unknown ADF AID | 2- returns null  3- applet is triggered  4- returns null |  |
| 2 | Normal execution  1- Envelope menu selection is sent to the UICC  Invoke the method getTheAdminFileView() with AID = ADF1 with the event JCSystem.NOT\_A\_TRANSIENT\_OBJECT  and stores the result in a class variable FV1  Applet1 calls FV1.status() command  Select DFTest using FV1 Select EFTARU using FV1Read first 3 bytes using FV1  Reset  Terminal profile  2 - Envelope menu selection is sent to the UICC  Applet1 calls FV1.status() command  Read first 3 bytes using FV1  Applet1 calls FV1.select(EFRFU1)  Invoke the method getTheAdminFileView() with the event JCSystem.CLEAR\_ON\_RESET and stores the result in a class variable FV2  Applet1 calls FV2.status() command  Select DFTest using FV2  Select EFTARU using FV2  Read first 3 bytes using FV2  Reset  Terminal profile  4 - Envelope menu selection is sent to the UICC  Applet1 calls FV2.status() command  Read first 3 bytes using FV2  Applet1 calls FV2.select(EFRFU1)  5- Select the Applet by AID  Invoke the method getTheAdminFileView() with AID = ADF1 with the event: JCSystem.CLEAR\_ON\_DESELECT and stores the result in a class variable FV3  Applet1 calls FV3.status() command  Select DFTest using FV3 Select EFTARU using FV3  Read first 3 bytes using FV3  6- Select the Applet by AID  Applet1 calls FV3.status() command  Read first 3 bytes using FV3  Applet1 calls FV3.select(EFRFU1) | 1- Applet1 is triggered  No Exception shall be thrown  Current selected DF is ADF1  Expected value is {FF FF FF}  2- Applet1 is triggered  Current selected DF is DFTest  Expected value is {FF FF FF}  UICCExceptionUICCException.FILE\_NOT\_FOUND is thrown  No exception shall be thrown  Current selected DF is the ADF1  Expected value is {FF FF FF}  4- Applet1 is triggered  Current selected DF is the ADF1  [UICCException](file:///e:\Documenti\Standard%20&%20Specifications\ETSI\TSStandard\ts_102241v060700p0\102241_Annex_A_HTML\uicc\access\UICCException.html). NO\_EF\_SELECTED  [UICCException](file:///e:\Documenti\Standard%20&%20Specifications\ETSI\TSStandard\ts_102241v060700p0\102241_Annex_A_HTML\uicc\access\UICCException.html).FILE\_NOT\_FOUND is thrown  5- Applet1 is selected  No Exception shall be thrown  Current selected DF is ADF1  Expected value is {FF FF FF}  6- Applet1 is selected  Current selected DF is ADF1  UICCException.NO\_EF\_SELECTED shall be thrown  UICCException.FILE\_NOT\_FOUND shall be thrown |  |
| 3 | FileView context independency  1- Envelope menu selection is sent to the UICC  2- Check that previous fileviews are different (FV1 != FV2 != FV3)  3- Select DFTest/EFLARU using FV1  4- Select DFTest/EFCARU using FV2  5- Select DFTest/EFCARU using FV3  6- Read record number 1 using FV1 (in absolute mode)  7- Read record number 2 using FV2 (in absolute mode) | 1- Applet1 is triggered  3- No exception shall be thrown  4- No exception shall be thrown  5- An exception shall be thrown  6- Expected value is “55 55 55 55”  7- Expected value is “AA AA AA” |  |
| 4 | ILLEGAL\_TRANSIENT SystemException  1- Envelope menu selection is sent to the UICC  2- Applet1 calls getTheAdminFileView() method with the event JCSystem.CLEAR\_ON\_DESELECT | 1- Applet1 is triggered  2- SystemException. ILLEGAL\_TRANSIENT is thrown |  |
| 5 | NO\_TRANSIENT\_SPACE SystemException  with CLEAR\_ON\_RESET FileView object  1 Get the available transient memory space using method length = JCSystem.getAvailableMemory(MEMORY\_TYPE\_TRANSIENT\_RESET)  2- If length < 32767, (test case could be performed)  2.1- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR\_ON\_RESET)  2.2- Applet calls method getTheAdminFileView() with AID = ADF1  with event  JCSystem.CLEAR\_ON\_RESET | 1- No Exception shall be thrown  2.1- No Exception shall be thrown  2.2-SystemException.NO\_TRANSIENT\_SPACE is thrown |  |
| 6 | NO\_TRANSIENT\_SPACE SystemException  with CLEAR\_ON\_DESELECT FileView object  1 - Select the Applet by AID  2- Get the available transient memory space using method length = JCSystem.getAvailableMemory(MEMORY\_TYPE\_TRANSIENT\_DESELECT)  3- If length < 32767, (test case could be performed)  3.1- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR\_ON\_DESELECT)  }  3.2- Applet calls method getTheAdminFileView() with AID = ADF1  with event:  JCSystem.CLEAR\_ON\_DESELECT  4- Reset | 1- Applet1 is triggered  2- No Exception shall be thrown  3.1- No Exception shall be thrown  3.2- SystemException. NO\_TRANSIENT\_SPACE is thrown |  |
| 7 | ILLEGAL\_VALUE SystemException  1- Invoke the method getTheAdminFileView() with every event codes except: 0, 1, 2 | 1- SystemException.ILLEGAL\_VALUE is thrown |  |
| 8 | NullPointerException  Invoke the method getTheAdminFileView() with AID = NULL  with event:  1 - JCSystem.CLEAR\_ON\_RESET | 1- Shall be thrown  java.lang.NullPointerException |  |

#### 5.3.2.3 Method getTheAdminFileView(byte[] buffer, short bOffset, short bLength, byte event)

Test Area Reference: Api\_4\_Afb\_Gtaf\_Bsbb.

##### 5.3.2.3.1 Conformance requirement:

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

public static AdminFileView getTheAdminFileView(byte[] buffer, short bOffset, short bLength, byte event)

throws NullPointerException,

javacard.framework.SystemException,

ArrayIndexOutOfBoundException

5.3.2.3.1.1 Normal execution

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

5.3.2.3.1.2 Parameters error

* CRRP1: If event is not one of the following values JCSystem.NOT\_A\_TRANSIENT\_OBJECT, JCSystem.CLEAR\_ON\_DESELECT, or JCSystem.CLEAR\_ON\_RESET a SystemException is thrown. The value of the SystemException shall be SystemException.ILLEGAL\_VALUE.
* CRRP2: If the buffer is null a NullPointerException shall be thrown.
* CRRP3: if bLength is less then 0, an instance of ArrayIndexOutOfBoundsException shall be thrown.
* CRRP4: if bOffset plus bLength is greater than the length of the array buffer.length, an instance of ArrayIndexOutOfBoundsException shall be thrown.
* CRRP5: if bLength is not in the range of 5 - 16 bytes a SystemException.ILLEGAL\_VALUE shall be thrown.

5.3.2.3.1.3 Context errors

* CRRC1: If event is JCSystem.CLEAR\_ON\_RESET or JCSystem.CLEAR\_ON\_DESELECT and not enough transient memory space is available a SystemException shall be thrown with reason code SystemException.NO\_TRANSIENT\_SPACE.
* CRRC2: If event is JCSystem.CLEAR\_ON\_DESELECT and the applet is not a currently selected applet a SystemException with reason code SystemException.ILLEGAL\_TRANSIENT shall be thrown.

##### 5.3.2.3.2 Test area files

Test Source: Test\_ Api\_4\_Afb\_Gtaf\_Bsbb.java.

Test Applet: Api\_4\_Afb\_Gtaf\_Bsbb.java.

Cap File: Api\_4\_Afb\_Gtaf\_Bsbb.cap.

##### 5.3.2.3.3 Test coverage

|  |  |
| --- | --- |
| CRR number | Test case number |
| N1 | 2 |
| N2 | 3 |
| N3 | 1 |
| N4 | 2 |
| N5 | 11, 12 |
| N6 | 5 |
| P1 | 2 |
| P2 | 5 |
| P3 | 7 |
| P4 | 8 |
| P5 | 9 |
| C1 | 11, 12  Testable only if available transient space is lower than 32767 |
| C2 | 6 |

##### 5.3.2.3.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 1 | Method returns null  1- Install Applet1 with full access rights on the UICC file system  2- Invoke the method getTheFileView before the javacard.framework.Applet.register(..) method invocation  Invoke the method getTheAdminFileView() with buffer[20] = {ADF1,…}  bOffset= 0  bLength= 16  3- Envelope menu selection is sent to the UICC  4- Invoke the method getTheAdminFileView before the javacard.framework.Applet.register(..) method invocation  Invoke the method getTheAdminFileView() with buffer[] = unknown aid | 2- returns null  3- Applet is triggered  4- returns null |  |
| 2 | Normal execution  1- Envelope menu selection is sent to the UICC  Invoke the method getTheAdminFileView()  with buffer[20] = {ADF1,…}  bOffset= 0  bLength= 16  JCSystem.NOT\_A\_TRANSIENT\_OBJECT  and stores the result in a class variable FV1  Applet1 calls FV1.status() command  Select DFTest using FV1  Select EFTARU using FV1  Read first 3 bytes using FV1  Reset  Terminal profile  2 - Envelope menu selection is sent to the UICC  Applet1 calls FV1.status() command  Read first 3 bytes using FV1  Applet1 calls FV1.select(EFRFU1)  Invoke the method getTheAdminFileView() with the event JCSystem.CLEAR\_ON\_RESET and stores the result in a class variable FV2  Applet1 calls FV2.status() command  Select DFTest using FV2  Select EFTARU using FV2  Read first 3 bytes using FV2  Reset  Terminal profile  4 - Envelope menu selection is sent to the UICC  Applet1 calls FV2.status() command  Read first 3 bytes using FV2  Applet1 calls FV2.select(EFRFU1)  5- Select the Applet by AID  Invoke the method getTheAdminFileView() with AID = ADF1 with  buffer[20] = {ADF1,…}  bOffset= 0  bLength= 16  the event:JCSystem.CLEAR\_ON\_DESELECT and stores the result in a class variable FV3  Applet1 calls FV3.status() command  Select DFTest using FV3  Select EFTARU using FV3  Read first 3 bytes using FV3  6- Select the Applet by AID  Applet1 calls FV3.status() command  Read first 3 bytes using FV3  Applet1 calls FV3.select(EFRFU1) | 1- Applet1 is triggered  No Exception shall be thrown  Current selected DF is ADF1  Expected value is {FF FF FF}  2- Applet1 is triggered  Current selected DF is DFTest  Expected value is {FF FF FF}  UICCExceptionUICCException.FILE\_NOT\_FOUND is thrown  No exception shall be thrown  Current selected DF is ADF1  Expected value is {FF FF FF}  4- Applet1 is triggered  Current selected DF is ADF1  UICCException.NO\_EF\_SELECTED.  [UICCException](file:///e:\Documenti\Standard%20&%20Specifications\ETSI\TSStandard\ts_102241v060700p0\102241_Annex_A_HTML\uicc\access\UICCException.html).FILE\_NOT\_FOUND is thrown  5- Applet1 is selected  No Exception shall be thrown  Current selected DF is ADF1  Expected value is {FF FF FF}  6- Applet1 is selected  Current selected DF is ADF1  UICCException.NO\_EF\_SELECTED.  [UICCException](file:///e:\Documenti\Standard%20&%20Specifications\ETSI\TSStandard\ts_102241v060700p0\102241_Annex_A_HTML\uicc\access\UICCException.html).FILE\_NOT\_FOUND is thrown |  |
| 3 | FileView context independency  1- Envelope menu selection is sent to the UICC  2- Check that previous fileviews are different (FV1 != FV2 != FV3)  3- Select DFTest/EFLARU using FV1  4- Select DFTest/EFCARU using FV2  5- Select DFTest/EFCARU using FV3  6- Read record number 1 using FV1 (in absolute mode)  7- Read record number 2 using FV2 (in absolute mode) | 1- Applet1 is triggered  3- No exception shall be thrown  4- No exception shall be thrown  5- An exception shall be thrown  6- Expected value is “55 55 55 55”  7- Expected value is “AA AA AA” |  |
| 4 | ILLEGAL\_TRANSIENT SystemException  1- Envelope menu selection is sent to the UICC  2- Applet1 calls getTheAdminFileView() method with  buffer[20] = {ADF1,…}  bOffset= 0  bLength= 16  with  the event JCSystem.CLEAR\_ON\_DESELECT | 1- Applet1 is triggered  2- SystemException. ILLEGAL\_TRANSIENT is thrown |  |
| 5 | NO\_TRANSIENT\_SPACE SystemException  with CLEAR\_ON\_RESET FileView object  1- Get the available transient memory space using method length = JCSystem.getAvailableMemory(MEMORY\_TYP\_TRANSIENT\_RESET)  2- If length < 32767, (test case could be performed)  2.1- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR\_ON\_RESET)  2.2- Applet calls method getTheAdminFileView() with buffer[20] = {ADF1,…}  bOffset= 0  bLength= 16  with the event  JCSystem.CLEAR\_ON\_RESET | 1- No Exception shall be thrown  2.1- No Exception shall be thrown  2.2- SystemException. NO\_TRANSIENT\_SPACE is thrown |  |
| 6 | NO\_TRANSIENT\_SPACE SystemException  with CLEAR\_ON\_DESELECT FileView object  1 - Select the Applet by AID  2- Get the available transient memory space using method length = JCSystem.getAvailableMemory(MEMORY\_TYPE\_TRANSIENT\_DESELECT)  3- If length < 32767, (test case could be performed)  3.1- Fill the available transient memory space by creating array, using method JCSystem.makeTransientByteArray(length, JCSystem.CLEAR\_ON\_DESELECT)  }  3.2- Applet calls method getTheAdminFileView() with buffer[20] = {ADF1,…}  bOffset= 0  bLength= 16  with event:  JCSystem.CLEAR\_ON\_DESELECT  4- Reset | 1- Applet1 is triggered  2- No Exception shall be thrown  3.1- No Exception shall be thrown  3.2- SystemException. NO\_TRANSIENT\_SPACE is thrown |  |
| 7 | ILLEGAL\_VALUE SystemException  1- Invoke the method getTheAdminFileView() with every event codes except: 0, 1, 2 | 1- SystemException.ILLEGAL\_VALUE is thrown |  |
| 8 | NullPointerException  Invoke the method getTheAdminFileView() with buffer[20] = null  bOffset= 0  bLength= 16  with event:  1 - JCSystem.CLEAR\_ON\_RESET | 1- Shall be thrown  java.lang.NullPointerException |  |
| 9 | ArrayIndexOutOfBoundsException  1-Envelope menu selection is sent to the UICC  Invoke the method getTheAdminFileView() with buffer[20] = {ADF1,…}  bOffset= 5  bLength= 16  event =JCSystem. CLEAR\_ON\_RESET  Invoke the method getTheAdminFileView() with buffer[20] = {ADF1,…}  bOffset= -1  bLength= 16  event =JCSystem. CLEAR\_ON\_RESET | 1- Applet1 is triggered  Shall be thrown ArrayIndexOutOfBoundsException  Shall be thrown ArrayIndexOutOfBoundsException |  |
| 10 | SystemException.ILLEGAL\_VALUE  1-Envelope menu selection is sent to the UICC  Invoke the method getTheAdminFileView() with buffer[20] = {ADF1,…}  bOffset= 0  bLength= 4  event =JCSystem. CLEAR\_ON\_RESET  Invoke the method getTheAdminFileView() with buffer[20] = {ADF1,…}  bOffset= 0  bLength= 17  event =JCSystem. CLEAR\_ON\_RESET | 1- Applet1 is triggered  SystemException.ILLEGAL\_VALUE shall be thrown  SystemException.ILLEGAL\_VALUE shall be thrown |  |

### 5.3.3 Class AdminException

#### 5.3.3.1 Constructor

Test Area Reference: Api\_4\_Aex\_Coor.

##### 5.3.3.1.1 Conformance requirement

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

public AdminException(short reason)

5.3.3.1.1.1 Normal execution

* CRRN1: Constructs an AdminException with the specified reason.

5.3.3.1.1.2 Parameter errors

No requirements.

5.3.3.1.1.3 Context errors

No requirements.

##### 5.3.3.1.2 Test area files

Test Source: Test\_Api\_4\_Aex\_Coor.java.

Test Applet: Api\_4\_Aex\_Coor\_1.java.

Cap File: Api\_4\_Aex\_Coor.cap.

##### 5.3.3.1.3 Test coverage

|  |  |
| --- | --- |
| CRR number | Test case number |
| N1 | 1 |

##### 5.3.3.1.4 Test procedure

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

#### 5.3.3.2 Method throwIt

Test Area Reference: Api\_4\_Aex\_Thit.

##### 5.3.3.2.1 Conformance requirement

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

public static void throwIt(short reason)

throws AdminException

5.3.3.2.1.1 Normal execution

* CRRN1: Throws the JCRE instance of AdminException with the specified reason.
* CRRN2: Etends javacard.framework.CardRuntimeException.

5.3.3.2.1.2 Parameter errors

No requirements.

5.3.3.2.1.3 Context errors

No requirements.

##### 5.3.3.2.2 Test area files

Test Source: Test\_Api\_4\_Aex\_Thit.java.

Test Applet: Api\_4\_Aex\_Thit\_1.java.

Cap File: Api\_4\_Aex\_Thit.cap.

##### 5.3.3.2.3 Test coverage

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

##### 5.3.3.2.4 Test procedure

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

#### 5.3.3.3 Reason Codes

Test Area Reference: Api\_4\_Aex\_Cons.

##### 5.3.3.3.1 Conformance Requirement:

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

5.3.3.3.1.1 Normal execution

* CRRN1: The Constants of the class AdminException shall all have the same name and value defined in the ETSI TS 102 241 [9].
* CRRN2: Constructs AdminException an Exception with the specified reason.

5.3.3.3.1.2 Parameter errors

No requirements.

5.3.3.3.1.3 Context errors

No requirements.

##### 5.3.3.3.2 Test area files

None.

##### 5.3.3.3.3 Test Coverage

|  |  |
| --- | --- |
| CRR number | Test case number |
| N1 & N2 | The constants in Java are resolved at compilation time, therefore a runtime test is not useful. No test of constants will be performed |

##### 5.3.3.3.4 Test Procedure

None.

## 5.4 Package uicc.system

### 5.4.1 Class HandlerBuilder

#### 5.4.1.1 Method buildTLVHandler(byte type, short capacity)

Test Area Reference: Api\_3\_Hdb\_Bthdbs.

##### 5.4.1.1.1 Conformance requirement

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

public static ViewHandler buildTLVHandler(byte type,

short capacity)

throws java.lang.ArrayIndexOutOfBoundsException,

javacard.framework.SystemException

5.4.1.1.1.1 Normal execution

* CRRN1: Allocates a TLVHandler with an internal buffer of length capacity.

5.4.1.1.1.2 Parameter errors

* CRRP1: If the type parameter does not match with the predefined values, a javacard.framework.SystemException is thrown with ILLEGAL\_VALUE reason code.
* CRRP2: If capacity is negative, a javacard.framework.SystemException is thrown with ILLEGAL\_VALUE reason code.

5.4.1.1.1.3 Context errors

* CRRC1: If there are not enough resources in the card to allocate the handler, a javacard.framework.SystemException is thrown with NO\_RESOURCE reason code.

##### 5.4.1.1.2 Test area files

Test Source: Test\_Api\_3\_Hdb\_Bthdbs.java.

Test Applet: Api\_3\_Hdb\_Bthdbs\_1.java.

Cap File: Api\_3\_hdb\_bthdbs.cap.

##### 5.4.1.1.3 Test coverage

|  |  |
| --- | --- |
| CRR number | Test case number |
| N1 | 1, 2 |
| P1 | 4, 5 |
| P2 | 3 |
| C1 | Not testable |

##### 5.4.1.1.4 Test procedure

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Description | API Expectation | APDU Expectation |
| 1 | Call buildTLVHandler() method with EDIT\_HANDLER type  Type = EDIT\_HANDLER  Capacity = (short)10  Check the created object is not null | No exception shall be thrown |  |
| 2 | Call buildTLVHandler() method with BER\_EDIT\_HANDLER type  Type = BER\_EDIT\_HANDLER  Capacity = (short)10  Check the created object is not null | No exception shall be thrown |  |
| 3 | Negative capacity  Type = EDIT\_HANDLER  Capacity = (short)-10 | A javacard.framework.SystemException is thrown with ILLEGAL\_VALUE reason code.. |  |
| 4 | Type does not match with predefined values  Type = (byte)3  Capacity = (short)10 | javacard.framework.SystemException shall be thrown with ILLEGAL\_VALUE reason code. |  |
| 5 | Type does not match with predefined values  Type = (byte)0  Capacity = (short)10 | javacard.framework.SystemException shall be thrown with ILLEGAL\_VALUE reason code. |  |

#### 5.4.1.2 Method buildTLVHandler(byte type, short capacity, byte[] buffer , short offset, short length)

Test Area Reference: Api\_3\_Hdb\_Bthdbs\_Bss.

##### 5.4.1.2.1 Conformance requirement

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

public static ViewHandler buildTLVHandler(byte type,

short capacity,

byte[] buffer,

short offset,

short length)

throws java.lang.ArrayIndexOutOfBoundsException,

javacard.framework.SystemException,

java.lang.NullPointerException

5.4.1.2.1.1 Normal execution

* CRRN1: Allocates a TLVHandler with an internal buffer of length capacity.
* CRRN2: Copies the buffer content to an internal buffer of the TLVHandler starting at bOffset.
* CRRN3: The internal buffer shall be at least bLength long.

5.4.1.2.1.2 Parameter errors

* CRRP1: If the type does not match with the predefined values, a javacard.framework.SystemException is thrown with ILLEGAL\_VALUE reason code.
* CRRP2: If buffer is null, a java.lang.NullPointerException is thrown.
* CRRP3: If bOffset would cause access outside array bounds, an java.lang.ArrayIndexOutOfBoundsException is thrown.
* CRRP4: if bLength is negative, a java.lang.ArrayIndexOutOfBoundsException is thrown.
* CRRP5: If capacity is negative, a javacard.framework.SystemException is thrown with ILLEGAL\_VALUE reason code.
* CRRP6: If bOffset+bLength is greater than the length of the buffer, a java.lang.ArrayIndexOutOfBoundsException is thrown.

5.4.1.2.1.3 Context errors

* CRRC1: If there are not enough resources in the card to allocate the handler, a javacard.framework.SystemException is thrown with NO\_RESOURCE reason code.

##### 5.4.1.2.2 Test area files

Test Source: Test\_Api\_3\_Hdb\_Bthdbs\_Bss.java.

Test Applet: Api\_3\_Hdb\_Bthdbs\_Bss\_1.java.

Cap File: Api\_3\_hdb\_bthdbs\_bss.cap.

##### 5.4.1.2.3 Test coverage

|  |  |
| --- | --- |
| CRR number | Test case number |
| N1 | 1, 2 |
| N2 | 6 |
| N3 | 1, 2, 6 |
| P1 | 4, 5 |
| P2 | 7 |
| P3 | 8, 9 |
| P4 | 10 |
| P5 | 3 |
| P6 | 11 |
| C1 | Not testable |

##### 5.4.1.2.4 Test procedure

| Id | Description | API Expectation | APDU Expectation |
| --- | --- | --- | --- |
| 1 | Call buildTLVHandler() method with EDIT\_HANDLER type  Type = EDIT\_HANDLER  Capacity = (short)10  Buffer[10]  Offset = (short)0  Length = (short)0  Check the created object is not null | No exception shall be thrown |  |
| 2 | Call buildTLVHandler() method with BER\_EDIT\_HANDLER type  Type = BER\_EDIT\_HANDLER  Capacity = (short)10  Buffer[10]  Offset = (short)0  Length = (short)0  Check the created object is not null | No exception shall be thrown |  |
| 3 | Negative capacity  Type = EDIT\_HANDLER  Capacity = (short)-10  Buffer[10]  Offset = (short)0  Length = (short)5 | A javacard.framework.SystemException is thrown with ILLEGAL\_VALUE reason code.. |  |
| 4 | Type does not match with predefined values  Type = (byte)0  Capacity = (short)10  Buffer[10]  Offset = (short)0  Length = (short)5 | A javacard.framework.SystemException shall be thrown with ILLEGAL\_VALUE reason code |  |
| 5 | Type does not match with predefined values  Type = (byte)3  Capacity = (short)10  Buffer[10]  Offset = (short)0  Length = (short)5 | A javacard.framework.SystemException shall be thrown with ILLEGAL\_VALUE reason code |  |
| 6 | Internal Buffer starts at bOffset  Type = EDIT\_HANDLER  Capacity = (short)10  Buffer[10]  Offset = (short)4  Length = (short)5  Check the internal buffer of the TLVHandler starts with bOffset data. | No exception shall be thrown |  |
| 7 | Buffer is null  Type = EDIT\_HANDLER  Capacity = (short)10  Buffer[] = null  Offset = (short)0  Length = (short)5 | A java.lang.NullPointerException shall be thrown. |  |
| 8 | bOffset > Buffer Length  Type = EDIT\_HANDLER  Capacity = (short)10  Buffer[10]  Offset = (short)11  Length = (short)0 | A java.lang.ArrayIndexOutOfBoundsException shall be thrown |  |
| 9 | bOffset < 0  Type = EDIT\_HANDLER  Capacity: (short)10  Buffer[10]  Offset = (short)-1  Length = (short)0 | A java.lang.ArrayIndexOutOfBoundsException shall be thrown |  |
| 10 | bLength < 0  Type = EDIT\_HANDLER  Capacity = (short)10  Buffer[10]  Offset = (short)0  Length = (short)-1 | A java.lang.ArrayIndexOutOfBoundsException shall be thrown |  |
| 11 | bOffset+bLength > buffer length  Type = EDIT\_HANDLER  Capacity = (short)10  Buffer[10]  Offset = (short)7  Length = (short)8 | A java.lang.ArrayIndexOutOfBoundsException shall be thrown |  |

### 5.4.2 Interface UICCPlatform

#### 5.4.2.1 Method getTheVolatileByteArray

Test Area Reference: Api\_3\_Upf\_Gvba.

##### 5.4.2.1.1 Conformance requirement

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

public static byte[] getTheVolatileByteArray()

5.4.2.1.1.1 Normal execution

* CRRN1: Returns the instance of the volatile byte array designated by the JCRE as global array. The byte array length shall be at least equal to 256 bytes.

5.4.2.1.1.2 Parameter errors

No requirement.

5.4.2.1.1.3 Context errors

* CRRC1: If the method is invoked from a context which is not the currently selected applet or the currently triggered applet i.e. the context of the applet that treats the current APDU or the context of the applet that has been triggered by the current APDU, a java.lang.SecurityException is thrown.
* CRRC2: A reference to this byte array cannot be stored in class variables or instance variables or array components.

##### 5.4.2.1.2 Test area files

Test Source: Test\_Api\_3\_Upf\_Gvba.java.

Test Applet: Api\_3\_Upf\_Gvba\_1.java.

Api\_ShareableInterface.java.

Api\_GetShareableClientApplet.java.

Cap File: Api\_3\_upf\_gvba.cap.

Api\_3\_upf\_gvba2.cap.

##### 5.4.2.1.3 Test coverage

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

##### 5.4.2.1.4 Test procedure

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Description | API Expectation | APDU Expectation |
| 1 | Call getTheVolatileByteArray() method and store it in a local variable  1- Trigger the applet and check the returned byte array length is at least equal to 256 bytes.  2- Select the applet and check the returned byte array length is at least equal to 256 bytes. | 1- No exception shall be thrown.  2- No exception shall be thrown. |  |
| 2 | Method invoked from a different context  By the way of the Shareable Interface, call the getTheVolatileByteArray() method through another applet. | A java.lang.SecurityException shall be thrown. |  |
| 3 | Store the instance in a class variable | A java.lang.SecurityException shall be thrown. |  |
| 4 | Store the instance in an instance variable | A java.lang.SecurityException shall be thrown. |  |
| 5 | Store the instance in an array component | A java.lang.SecurityException shall be thrown. |  |