3GPP TSG-T (Terminals) Meeting #14
Kyoto, Japan, 12 - 14 December 2001

Tdoc TP-010241

Source: T3

Title: Change Request to Java API specifications (TS 03.19 / 43.019 / 11.13)

Document for: Approval

This document contains several change requests as follows:

T3 Doc	Spec	CR	Rel	Ca	Subject
T3-010790	03.19	A017	R99	F	Clarification of ToolkitException.OUT_OF_TLV_BOUNDARIES in ViewHandler.java
T3-010764	43.019	001	Rel-5	F	API methods and Framework behaviour clarifications regarding ProactiveHandler and EnvelopeResponseHandler
T3-010765	43.019	002	Rel-5	D	Editorial modifications to constant name
T3-010766	43.019	003	Rel-5	B	Addition of the EVENT_FIRST_COMMAND_AFTER_SELECT as a toolkit event
T3-010767	43.019	004	Rel-5	C	extension of list of Simple BER TLV tags in sim.toolkit.ToolkitConstants
T3-010768	43.019	005	Rel-5	C	ToolkitRegistry methods modification
T3-010769	43.019	006	Rel-5	C	ToolkitRegistry methods modification when no TAR is defined
T3-010770	43.019	007	Rel-5	C	Applet triggering on Menu Help Request event
T3-010791	43.019	008	Rel-4	A	Clarification of ToolkitException.OUT_OF_TLV_BOUNDARIES in ViewHandler.java
T3-010792	43.019	009	Rel-5	A	Clarification of ToolkitException.OUT_OF_TLV_BOUNDARIES in ViewHandler.java
T3-010771	11.13	A003	R98	F	Specification for framework part
T3-010772	11.13	A004	R98	F	Alignment with core specifciation

3GPP T3 Meeting #21
Kyoto, Japan, 5 - 7 November, 2001

Tdoc T3-010790
revised T3-010682

CR-Form-v3

CHANGE REQUEST

⌘ **03.19** **CR** **A017** ⌘ rev **-** ⌘ Current version: 8.2.0 ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Clarification of ToolkitException.OUT_OF_TLV_BOUNDARIES in ViewHandler.java
Source:	⌘	T3
Work item code:	⌘	SIM Toolkit API
Date:	⌘	07/11/2001
Category:	⌘	**F**
Release:	⌘	R99

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change: ⌘ Currently the ToolkitException OUT_OF_TLV_BOUNDARIES is thrown if a TLV object with length 0 is accessed.
Clarify the enumeration of the different cases for the ToolkitException OUT_OF_TLV_BOUNDARIES.

Summary of change: ⌘ - Clarify rule when ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown
- Allow access to a zero length substring at the end of a TLV element

The current specification is unclear when exactly to throw a OUT_OF_TLV_BOUNDARIES exception. The term currently used is "Throw OUT_OF_TLV_BOUNDARIES if valueOffset, compareLength or both are out of the current TLV".

This is unclear, especially when it comes to the special case of accessing zero bytes at the end of the TLV element. Is it allowed, for example to copy zero bytes from the end of the element? As an example, with the current spec it is probably not possible to write the following statement, if a secured Envelope with zero length secured data is received.

```
envHandler.copyValue( envHdlr.getSecuredDataOffset(),
                      dest,
                      destOffset,
                      envHdlr.getSecuredDataLength() );
```

The Applet has to test for that special case beforehand.
We propose to make the condition when to throw an exception compliant to the standard array access rules used in javacard.framework.
See Util.arrayCopy(...) as an example

Consequences if not approved:	⌘	In each implementation the case “Length of TLV object = 0” must be checked and processed separately.

Clauses affected:	⌘	Annex A -> ViewHandler.java:
		Methods affected:
		`public short copyValue(short valueOffset, byte destBuffer[], short destOffset, short destLength ) throws [..] ToolkitException`
		
		`public byte compareValue(short valueOffset, byte compareBuffer[], short compareOffset, short compareLength ) throws [..] ToolkitException`
		
		`public short findAndCopyValue(byte tag, byte occurence, short valueOffset, byte destBuffer[], short destOffset, short destLength) throws [..] ToolkitException`
		
		`public byte findAndCompareValue(byte tag, byte occurence, short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength) throws [..] ToolkitException`

Other specs Affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

```
/**
    * Copies a part of the last TLV element which has been found, into a
    * destination buffer.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>dstOffset</code><em> or </em><code>dstLength</code><em> parameter is
negative an </em><code>ArrayIndexOutOfBoundsException</code>
    * <em> exception is thrown and no copy is performed.</em>
    * <li><em>If </em><code>dstOffset+dstLength</code><em> is greater than
</em><code>dstBuffer.length</code><em>, the length
    * of the </em><code>dstBuffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no copy is performed.</em>
    * </ul>
    *
    * @param valueOffset the offset of the first byte in the source TLV element
    * @param dstBuffer a reference to the destination buffer
    * @param dstOffset the position in the destination buffer
    * @param dstLength the data length to be copied
    *
    * @return <code>dstOffset+dstLength</code>
    *
    * @exception NullPointerException if <code>dstBuffer</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if copyValue would cause access of data outside array
bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
    *      <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element
    *      <li><code>OUT_OF_TLV_BOUNDARIES</code> if: ~~<code>valueOffset</code>,
<code>dstLength</code> or both are out of the current TLV </ul>~~
    *      <ul>
    *          <li><code>valueOffset</code> parameter is negative or
    *          <li><code>valueOffset + dstLength</code> is greater than the length of the current TLV
    *      </ul>
    *   </ul>
    */
   public short copyValue( short valueOffset,
                           byte[] dstBuffer,
                           short dstOffset,
                           short dstLength) throws NullPointerException,
                                                   ArrayIndexOutOfBoundsException,
                                                   ToolkitException {
                           return 0;
   }

/**
    * Compares the last found TLV element with a buffer.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>compareOffset</code><em> or </em><code>compareLength</code><em>
parameter is negative an </em><code>ArrayIndexOutOfBoundsException</code>
    * <em> exception is thrown and no compare is performed.</em>
    * <li><em>If </em><code>compareOffset+compareLength</code><em>is greater than
</em><code>compareBuffer.length</code><em>, the length
    * of the </em><code>compareBuffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no compare is performed.</em>
    * </ul>
    *
    * @param valueOffset the offset of the first byte to compare in the TLV element
```

```
    * @param compareBuffer a reference to the comparison buffer
    * @param compareOffset the position in the comparison buffer
    * @param compareLength the length to be compared
    *
    * @return the result of the comparison as follows: <ul>
    *      <li><code>0</code> if identical
    *      <li><code>-1</code> if the first miscomparing byte in simple TLV List is less than that in
<code>compareBuffer</code>,
    *      <li><code>1</code> if the first miscomparing byte in simple TLV List is greater than that in
<code>compareBuffer</code>.</ul>
    *
    * @exception NullPointerException if <code>compareBuffer</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if compareValue would cause access of data outside array
bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
    *      <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element
    *      <li><code>OUT_OF_TLV_BOUNDARIES</code> if:
    *       <ul>
    *           <li><code>valueOffset</code> parameter is negative or
    *           <li><code>valueOffset + dstLength</code> is greater than the length of the current TLV
    *       </ul>
    *   </ul> <code>valueOffset</code>, <code>compareLength</code> or both are out of the current TLV
</ul>
    */
   public byte compareValue(short valueOffset,
                            byte[] compareBuffer,
                            short compareOffset,
                            short compareLength) throws     NullPointerException,
                                                            ArrayIndexOutOfBoundsException,
                                                            ToolkitException {
                        return 0;
   }

   /**
    * Looks for the first occurence of a TLV element from the beginning of a TLV
    * list and copy its value into a destination buffer.
    * If no TLV element is found, the <code>UNAVAILABLE_ELEMENT</code> exception is thrown.
    * If the method is successful then the corresponding TLV becomes current,
    * else no TLV is selected.
    * This search method is Comprehension Required flag independent.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>dstOffset</code><em> parameter is negative or </em><code>dstOffset</code>
    * <em> is greater than </em><code>dstBuffer.length</code><em>, the length of the
</em><code>dstBuffer</code>
    * <em> array an </em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown and no
find is performed.</em>
    * </ul>
    *
    * @param tag the tag of the TLV element to search
    * @param dstBuffer a reference to the destination buffer
    * @param dstOffset the position in the destination buffer
    *
    * @return <code>dstOffset</code> + length of the copied value
    *
    * @exception NullPointerException if <code>dstBuffer</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if findAndCopyValue would cause access of data outside
array bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
```

```
     *     <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element</ul>
     */
    public short findAndCopyValue(byte tag,
                                  byte[] dstBuffer,
                                  short dstOffset) throws       NullPointerException,
                                                                ArrayIndexOutOfBoundsException,
                                                                ToolkitException {
                        return 0;
    }

    /**
     * Looks for the indicated occurence of a TLV element from the beginning of a TLV
     * list and copy its value into a destination buffer.
     * If no TLV element is found, the <code>UNAVAILABLE_ELEMENT</code> exception is thrown.
     * If the method is successful then the corresponding TLV becomes current,
     * else no TLV is selected.
     * This search method is Comprehension Required flag independent.
     *
     * <p>
     * Notes:<ul>
     * <li><em>If </em><code>dstOffset</code><em> or </em><code>dstLength</code><em> parameter is
negative an </em><code>ArrayIndexOutOfBoundsException</code>
     * <em> exception is thrown and no copy is performed.</em>
     * <li><em>If </em><code>dstOffset+dstLength</code><em>is greater than
</em><code>dstBuffer.length</code><em>, the length
     * of the </em><code>dstBuffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
     * and no copy is performed.</em>
     * </ul>
     *
     * @param tag the tag of the TLV element to search
     * @param occurrence the occurrence number of the TLV element (1 for the first, 2 for the second...)
     * @param valueOffset the offset of the first byte in the source TLV element
     * @param dstBuffer a reference to the destination buffer
     * @param dstOffset the position in the destination buffer
     * @param dstLength the data length to be copied
     *
     * @return <code>dstOffset + dstLength</code>
     *
     * @exception NullPointerException if <code>dstBuffer</code> is <code>null</code>
     * @exception ArrayIndexOutOfBoundsException if findAndCopyValue would cause access of data outside
array bounds.
     * @exception ToolkitException with the following reason codes: <ul>
     *     <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
     *     <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element
     *     <li><code>OUT_OF_TLV_BOUNDARIES</code> if:
     *      <ul>
     *          <li><code>valueOffset</code> parameter is negative or
     *          <li><code>valueOffset + dstLength</code> is greater than the length of the current TLV
     *      </ul> <code>valueOffset</code>, <code>dstLength</code> or both are out of the current TLV
     *     <li><code>BAD_INPUT_PARAMETER</code> if an input parameter is not valid (e.g. occurence =
0)</ul>
     */
    public short findAndCopyValue(byte tag,
                                  byte occurence,
                                  short valueOffset,
                                  byte[] dstBuffer,
                                  short dstOffset,
                                  short dstLength) throws       NullPointerException,
                                                                ArrayIndexOutOfBoundsException,
                                                                ToolkitException {
                        return 0;
```

```
    }

    /**
     * Looks for the first occurence of a TLV element from beginning of a TLV
     * list and compare its value with a buffer.
     * If no TLV element is found, the <code>UNAVAILABLE_ELEMENT</code> exception is thrown.
     * If the method is successful then the corresponding TLV becomes current,
     * else no TLV is selected.
     * This search method is Comprehension Required flag independent.
     *
     * <p>
     * Notes:<ul>
     * <li><em>If </em><code>compareOffset</code><em> parameter is negative or
</em><code>compareOffset</code>
     * <em> is greater than </em><code>compareBuffer.length</code><em>, the length of the
</em><code>compareBuffer</code>
     * <em> array an </em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown and no
find is performed.</em>
     * </ul>
     *
     * @param tag the tag of the TLV element to search
     * @param compareBuffer a reference to the comparison buffer
     * @param compareOffset the position in the comparison buffer
     *
     * @return the result of the comparison as follows: <ul>
     *      <li><code>0</code> if identical
     *      <li><code>-1</code> if the first miscomparing byte in simple TLV is less than that in
<code>compareBuffer</code>,
     *      <li><code>1</code> if the first miscomparing byte in simple TLV is greater than that in
<code>compareBuffer</code>.</ul>
     *
     * @exception NullPointerException if <code>compareBuffer</code> is <code>null</code>
     * @exception ArrayIndexOutOfBoundsException if findAndCompareValue would cause access of data
outside array bounds.
     * @exception ToolkitException with the following reason codes: <ul>
     *      <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
     *      <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element</ul>
     */
    public byte findAndCompareValue(byte tag,
                                    byte[] compareBuffer,
                                    short compareOffset) throws   NullPointerException,
                                                                  ArrayIndexOutOfBoundsException,
                                                                  ToolkitException {
                        return 0;
    }

    /**
     * Looks for the indicated occurence of a TLV element from the beginning of a
     * TLV list and compare its value with a buffer.
     * If no TLV element is found, the <code>UNAVAILABLE_ELEMENT</code> exception is thrown.
     * If the method is successful then the corresponding TLV becomes current,
     * else no TLV is selected.
     * This search method is Comprehension Required flag independent.
     *
     * <p>
     * Notes:<ul>
     * <li><em>If </em><code>compareOffset</code><em> or </em><code>compareLength</code><em>
parameter is negative an </em><code>ArrayIndexOutOfBoundsException</code>
     * <em> exception is thrown and no find and compare is performed.</em>
     * <li><em>If </em><code>compareOffset+compareLength</code><em> is greater than
</em><code>compareBuffer.length</code><em>, the length
```

```
    * of the </em><code>compareBuffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no find and compare is performed.</em>
    * </ul>
    *
    * @param tag the tag of the TLV element to search
    * @param occurrence the occurrence number of the TLV element (1 for the first, 2 for the second...)
    * @param valueOffset the offset of the first byte in the source TLV element
    * @param compareBuffer a reference to the comparison buffer
    * @param compareOffset the position in the comparison buffer
    * @param compareLength the length to be compared
    *
    * @return the result of the comparison as follows: <ul>
    *     <li><code>0</code> if identical
    *     <li><code>-1</code> if the first miscomparing byte in simple TLV is less than that in
<code>compareBuffer</code>,
    *     <li><code>1</code> if the first miscomparing byte in simple TLV is greater than that in
<code>compareBuffer</code>.</ul>
    *
    * @exception NullPointerException if <code>compareBuffer</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if findAndCompareValue would cause access of data
outside array bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *     <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
    *     <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element
    *     <li><code>OUT_OF_TLV_BOUNDARIES</code> if:
    *      <ul>
    *          <li><code>valueOffset</code> parameter is negative or
    *          <li><code>valueOffset + dstLength</code> is greater than the length of the current TLV
    *      </ul>
~~<code>valueOffset</code>, <code>compareLength</code> or both are out of the current TLV~~
    *     <li><code>BAD_INPUT_PARAMETER</code> if an input parameter is not valid (e.g. occurence =
0)</ul>
    */
   public byte findAndCompareValue(byte tag,
                                   byte occurence,
                                   short valueOffset,
                                   byte[] compareBuffer,
                                   short compareOffset,
                                   short compareLength) throws NullPointerException,
                                                               ArrayIndexOutOfBoundsException,
                                                               ToolkitException {
                       return 0;
   }
```

3GPP T3 Meeting #21
Kyoto, Japan, 5 - 7 November, 2001

Tdoc T3-010764
revised T3-010624

CR-Form-v3

CHANGE REQUEST

⌘ **43.019** **CR** **001** ⌘ rev **-** ⌘ Current version: **5.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	API methods and Framework behaviour clarifications regarding ProactiveHandler and EnvelopeResponseHandler
Source:	⌘	T3
Work item code:	⌘	SIM API
Date:	⌘	07/11/01
Category:	⌘	**F**
Release:	⌘	REL-5

Use one of the following categories:
- ***F*** *(essential correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(Addition of feature),*
- ***C*** *(Functional modification of feature)*
- ***D*** *(Editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	Behaviour that needs to be explicitely stated in the specification.
Summary of change:	⌘	Define : - The behviour of initDisplayText, initGetInkey and initGetInput methods of ProactiveHandler Java Class when the parameter “length” is equal to 0 - State of the EnvelopeResponseHandler object at the processToolkit method invocation.
Consequences if not approved:	⌘	Can lead to interoperability issues.

Clauses affected:	⌘	$ 6.6 , Annex A		
Other specs Affected:	⌘	[ ] Other core specifications [X] Test specifications [ ] O&M Specifications	⌘	11.13
Other comments:	⌘			

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be

downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

6.6 Handler availability

The system handlers : ProactiveHandler, ProactiveResponseHandler, EnvelopeHandler and EnvelopeResponseHandler are Temporary JCRE Entry Point Object as defined in the Java Card Runtime Environment Specification [8].

The following table describes the minimum availability of the handlers for all the events at the invocation of the processToolkit method of the toolkit applet.

Table 1: Handler availability for each event

EVENT_	Reply busy	ProactiveHandler ProactiveResponseHandler	EnvelopeHandler	EnvelopeResponseHandler	Nb of triggered / registrered Applet
_FORMATTED_SMS_PP_ENV	Y	Y	Y	Y	1 / n (per TAR)
_FORMATTED_SMS_PP_UPD	N	Y	Y	N	1 / n (per TAR)
_UNFORMATTED_SMS_PP_ENV	Y	Y	Y	Y	n / n
_UNFORMATTED_SMS_PP_UPD	N	Y	Y	N	n / n
_FORMATTED_SMS_CB	Y	Y	Y	N	1 / n (per TAR)
_UNFORMATTED_SMS_CB	Y	Y	Y	N	n / n
_MENU_SELECTION	Y	Y	Y	N	1 / n (per Item Id)
_MENU_SELECTION_HELP_REQUEST	Y	Y	Y	N	1 / n (per Item Id)
_CALL_CONTROL	N	Y/N (see Note 2)	Y	Y	1 / 1
_SMS_MO_CONTROL	N	Y/N (see Note 2)	Y	Y	1 / 1
_TIMER_EXPIRATION	Y	Y	Y	N	1/ 8 (per timer) (see Note 1)
_EVENT_DOWNLOAD					
_MT_CALL	Y	Y	Y	N	n / n
_CALL_CONNECTED	Y	Y	Y	N	n / n
_CALL_DISCONNECTED	Y	Y	Y	N	n / n
_LOCATION_STATUS	Y	Y	Y	N	n / n
_USER_ACTIVITY	Y	Y	Y	N	n / n
_IDLE_SCREEN_AVAILABLE	Y	Y	Y	N	n / n
_LANGUAGE_SELECTION	Y	Y	Y	N	n / n
_BROWSER_TERMINATION	Y	Y	Y	N	n / n
_CARD_READER_STATUS	Y	Y	Y	N	n / n
_UNRECOGNISED_ENVELOPE	Y	Y	Y	Y	n / n
_STATUS_COMMAND	N	Y/N (see Note 2)	N	N	n / n
_PROFILE_DOWNLOAD	N	Y/N (see Note 2)	N	N	n / n

NOTE 1: One toolkit applet can register to several timers, but a timer can only be allocated to one toolkit applet.
NOTE 2: Y/N means that handlers may / may not be available depending whether a proactive session is ongoing.

The following rules define the minimum requirement for the availability of the system handlers and the lifetime of their content.

ProactiveHandler:

- The ProactiveHandler is valid from the invocation to the termination of the processToolkit method.
- If a proactive command is pending the ProactiveHandler may not be available.
- At the processToolkit method invocation the TLV-List is cleared.
- At the call of it's init method the content is cleared and then initialised.
- After a call to ProactiveHandler.send method the handler will remain unchanged (i.e. previously send proactive command) until the ProactiveHandler.init or appendTLV methods are called.

ProactiveResponseHandler:

- The ProactiveResponseHandler may not be available before the first call to ProactiveHandler.send method, if available the content is cleared.

- The ProactiveResponseHandler is available after the first call to the ProactiveHandler.send method to the termination of the processToolkit method.
- If a proactive command is pending the ProactiveResponseHandler may not be available.
- The ProactiveResponseHandler content is changed after the call to ProactiveHandler.send method and remains unchanged until next call to the ProactiveHandler.send method.

EnvelopeHandler:

- The EnvelopeHandler and its content are available for all triggered toolkit applets (see Table1), from the invocation to the termination of their processToolkit method.
- The SIM Toolkit Framework guarantees that all registered toolkit applet are triggered and receive the data.

EnvelopeResponseHandler:

- The EnvelopeResponseHandler is available for all triggered toolkit applets, until a toolkit applet has posted an envelope response or sent a proactive command. After a call to the post method the handler is no longer available.
- At the process Toolkit method invocation the TLV-List is cleared.
- The EnvelopeResponseHandler content must be posted before the first invocation of a ProactiveHandler.send method or before the termination of the processToolkit, so that the GSM applet can offer these data to the ME (eg 9Fxx/9Exx). After the first invocation of the ProactiveHandler.send method the EnvelopeResponseHandler is no more available.

The following diagram illustrates these rules.

Applet		Applet 1							Applet 2				
method invocation	*ProcessToolkit*	*Init*		*post* send		*init* send		*Termination* ProcessToolkit	*init*		*init* send		
Envelope Handler													
EnvelopeResponseHandler													
ProactiveHandler													
Proactive ResponseHandler													

Figure 5: Typical handler availability for toolkit applets (see Table 1 for detail)

Annex A (normative): Java Card SIM API

The attached files "Annex_A_java.zip" and "Annex_A_HTML.zip" contains source files for the Java Card SIM API.

[the HTML and JAVA source files will be included]

List of changes to the API html and java source files

Class sim.toolkit. ProactiveHandler

Clarify methods description to match test suite:

```
/**
    * Builds a Display Text Proactive command without sending the command. The Comprehension
    * Required flags are all set to 1.
    * After the method invocation no TLV is selected.
    *
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>offset</code><em> or </em><code>length</code><em> parameter is
negative an </em><code>ArrayIndexOutOfBoundsException</code>
    * <em> exception is thrown and no proactive command is build.</em>
    * <li><em>If </em><code>offset+length</code><em> is greater than
</em><code>buffer.length</code><em>, the length
    * of the </em><code>buffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no proactive command is build.</em>
    * </ul>
    *
    * @param qualifier Display Text command qualifier
    * @param dcs data coding scheme
    * @param buffer reference to the text string source buffer
    * @param offset offset of the text string in the source buffer
    * @param length length of the text string in the source buffer. If length is equal to zero, then the
Text String TLV inserted in the command is a null text string TLV as defined in TS 11.14 [3].
    *
    * @exception NullPointerException if <code>buffer</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if initDisplayText would cause access of data
outside array bounds
    * @exception ToolkitException with the following reason codes: <ul>
    *     <li><code>HANDLER_OVERFLOW</code> if the ProactiveHandler buffer is to small to put
the requested data </ul>
    */
   public void initDisplayText(byte qualifier,
                               byte dcs,
                               byte[] buffer,
                               short offset,
                               short length) throws NullPointerException,
                                                    ArrayIndexOutOfBoundsException,
                                                    ToolkitException {
}

   /**
    * Builds a Get Inkey Proactive command without sending the command. The Comprehension
    * Required flags are all set to 1.
    * After the method invocation no TLV is selected.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>offset</code><em> or </em><code>length</code><em> parameter is
negative an </em><code>ArrayIndexOutOfBoundsException</code>
    * <em> exception is thrown and no proactive command is build.</em>
    * <li><em>If </em><code>offset+length</code><em> is greater than
</em><code>buffer.length</code><em>, the length
```

```
        * of the </em><code>buffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
        * and no proactive command is build.</em>
        * </ul>
        *
        * @param qualifier Get Inkey command qualifier
        * @param dcs data coding scheme
        * @param buffer reference to the displayed text string source buffer
        * @param offset offset of the displayed text string in the source buffer
        * @param length length of the displayed text string in the source buffer. If length is equal to zero,
then the Text String TLV inserted in the command is a null text string TLV as defined in TS 11.14 [3].
        *
        * @exception NullPointerException if <code>buffer</code> is <code>null</code>
        * @exception ArrayIndexOutOfBoundsException if initGetInkey would cause access of data outside
array bounds.
        * @exception ToolkitException with the following reason codes: <ul>
        *     <li><code>HANDLER_OVERFLOW</code> if the ProactiveHandler buffer is to small to put
the requested data</ul>
        */
        public void initGetInkey(byte qualifier,
                                 byte dcs,
                                 byte[] buffer,
                                 short offset,
                                 short length) throws NullPointerException,
                                                      ArrayIndexOutOfBoundsException,
                                                      ToolkitException {
    }

        /**
        * Initialize the building of a Get Input Proactive command. The Comprehension
        * Required flags are all set to 1.
        * The following command parameters (i.e. TLVs) may be appended to the
        * command before sending it: Default Text.
        * After the method invocation no TLV is selected.
        *
        * <p>
        * Notes:<ul>
        * <li><em>If </em><code>offset</code><em> or </em><code>length</code><em> parameter is
negative an </em><code>ArrayIndexOutOfBoundsException</code>
        * <em> exception is thrown and no proactive command is build.</em>
        * <li><em>If </em><code>offset+length</code><em>is greater than
</em><code>buffer.length</code><em>, the length
        * of the </em><code>buffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
        * and no proactive command is build.</em>
        * </ul>
        *
        * @param qualifier Get Input command qualifier
        * @param dcs data coding scheme
        * @param buffer reference to the displayed text string source buffer
        * @param offset offset of the displayed text string in the source buffer
        * @param length length of the displayed text string in the source buffer. If length is equal to zero,
then the Text String TLV inserted in the command is a null text string TLV as defined in TS 11.14 [3].
        * @param minRespLength minimal length of the response text string
        * @param maxRespLength maximal length of the response text string
        *
        * @exception NullPointerException if <code>buffer</code> is <code>null</code>
        * @exception ArrayIndexOutOfBoundsException if initGetInput would cause access of data outside
array bounds.
        * @exception ToolkitException with the following reason codes: <ul>
        *     <li><code>HANDLER_OVERFLOW</code> if the ProactiveHandler buffer is to small to put
the requested data</ul>
```

```
     */
    public void initGetInput(byte qualifier,
                        byte dcs,
                        byte[] buffer,
                        short offset,
                        short length,
                        short minRespLength,
                        short maxRespLength) throws NullPointerException,
                                    ArrayIndexOutOfBoundsException,
                                    ToolkitException {

}
```

3GPP T3 Meeting #21
Kyoto, Japan, 5 - 7 November, 2001

Tdoc T3-010765
revised T3-010706

CR-Form-v3

CHANGE REQUEST

⌘ **43.019 CR 002** ⌘ rev ⌘ Current versio 5.0.0 ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Editorial corrections of constant name
Source:	⌘	T3
Work item code:	⌘	SIM API
Date:	⌘	07/11/2001
Category:	⌘	**D**
Release:	⌘	REL-5

Use one of the following categories:
- ***F*** *(essential correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(Addition of feature),*
- ***C*** *(Functional modification of feature)*
- ***D*** *(Editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	Correct writing errors of a constant name in the specification
Summary of change:	⌘	correction of a constant name, change will to shall, applet to applets
Consequences if not approved:	⌘	Inconsistent constant value names in the specification

Clauses affected:	⌘	6.2; 6.6;		
Other specs affected:	⌘	[ ] Other core specifications [ ] Test specifications [ ] O&M Specifications	⌘	
Other comments:	⌘			

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.
3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

6.2 Applet Triggering

…

The ME shall not be adversely affected by the presence of applets on the SIM card. For instance a syntactically correct Envelope shall not result in an error status word in case of a failure of an applet. The only application as seen by the ME is the SIM application. As a result, a toolkit applet may throw an exception, but this error shall~~will~~ not be sent to the ME.

The difference between a Java Card applet and a Toolkit applet is that the latter does not handle APDUs directly. It will handle higher level messages. Furthermore the execution of a method could span over multiple APDUs, in particular, the proactive protocol commands (Fetch, Terminal Response).

As seen above, when the GSM applet is the selected application and when a toolkit applet is triggered the *select*() method of the toolkit applet shall not be launched since the toolkit applet itself is not really selected.

Here after are the events that can trigger a toolkit applet :

EVENT_PROFILE_DOWNLOAD

Upon reception of the Terminal Profile command by the SIM, the SIM Toolkit Framework stores the ME profile and then triggers the registered toolkit applets which may want to change their registry. A toolkit applet may not be able to issue a proactive command.

…

6.6 Handler availability

The system handlers : ProactiveHandler, ProactiveResponseHandler, EnvelopeHandler and EnvelopeResponseHandler are Temporary JCRE Entry Point Object as defined in the Java Card Runtime Environment Specification [8].

The following table describes the minimum availability of the handlers for all the events at the invocation of the processToolkit method of the toolkit applet.

Table 1: Handler availability for each event

EVENT_	Reply busy	ProactiveHandler ProactiveResponseHandler	EnvelopeHandler	EnvelopeResponseHandler	Nb of triggered / registrered Applet
_FORMATTED_SMS_PP_ENV	Y	Y	Y	Y	1 / n (per TAR)
_FORMATTED_SMS_PP_UPD	N	Y	Y	N	1 / n (per TAR)
_UNFORMATTED_SMS_PP_ENV	Y	Y	Y	Y	n / n
_UNFORMATTED_SMS_PP_UPD	N	Y	Y	N	n / n
_FORMATTED_SMS_CB	Y	Y	Y	N	1 / n (per TAR)
_UNFORMATTED_SMS_CB	Y	Y	Y	N	n / n
_MENU_SELECTION	Y	Y	Y	N	1 / n (per Item Id)
_MENU_SELECTION_HELP_REQUEST	Y	Y	Y	N	1 / n (per Item Id)
_CALL_CONTROL	N	Y/N (see Note 2)	Y	Y	1 / 1
_SMS_MO_CONTROL	N	Y/N (see Note 2)	Y	Y	1 / 1
_TIMER_EXPIRATION	Y	Y	Y	N	1/ 8 (per timer) (see Note 1)
_EVENT_DOWNLOAD					
_MT_CALL	Y	Y	Y	N	n / n
_CALL_CONNECTED	Y	Y	Y	N	n / n
_CALL_DISCONNECTED	Y	Y	Y	N	n / n
_LOCATION_STATUS	Y	Y	Y	N	n / n
_USER_ACTIVITY	Y	Y	Y	N	n / n
_IDLE_SCREEN_AVAILABLE	Y	Y	Y	N	n / n
_LANGUAGE_SELECTION	Y	Y	Y	N	n / n
_BROWSER_TERMINATION	Y	Y	Y	N	n / n
_CARD_READER_STATUS	Y	Y	Y	N	n / n
_UNRECOGNIZSED_ENVELOPE	Y	Y	Y	Y	n / n
_STATUS_COMMAND	N	Y/N (see Note 2)	N	N	n / n
_PROFILE_DOWNLOAD	N	Y/N (see Note 2)	N	N	n / n

NOTE 1: One toolkit applet can register to several timers, but a timer can only be allocated to one toolkit applet.
NOTE 2: Y/N means that handlers may / may not be available depending whether a proactive session is ongoing.

3GPP T3 Meeting #21
Kyoto, Japan, 5 - 7 November, 2001

Tdoc T3-010766
revised T3-010679

CR-Form-v4

CHANGE REQUEST

⌘ **43.019** **CR** **003** ⌘ ev **-** ⌘ Current version: **5.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Addition of the EVENT_FIRST_COMMAND_AFTER_SELECT as a toolkit event
Source:	⌘	T3
Work item code:	⌘	***Date:*** ⌘ 07/11/01
Category:	⌘	**B** ***Release:*** ⌘ REL-5

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	There is no event to allow a toolkit applet to perform operations like file updates before the ME has had the opportunity to access the files. Such an event is required for applications.
Summary of change:	⌘	Definition of the EVENT_FIRST_COMMAND_AFTER_SELECT event.
Consequences if not approved:	⌘	No possibility for a toolkit application to read or update a file before the ME.

Clauses affected:	⌘	§ 6.2, § 6.6, sim.toolkit.ToolkitConstants.java
Other specs affected:	⌘	[ ] Other core specifications ⌘
		[ ] Test specifications
		[ ] O&M Specifications
Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.
3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

6.2 Applet Triggering

The application triggering portion of the SIM Toolkit Framework is responsible for the activation of toolkit applets, based on the APDU received by the GSM application.

Figure 3: toolkit applet triggering diagram

The ME shall not be adversely affected by the presence of applets on the SIM card. For instance a syntactically correct Envelope shall not result in an error status word in case of a failure of an applet. The only application as seen by the ME is the SIM application. As a result, a toolkit applet may throw an exception, but this error will not be sent to the ME.

The difference between a Java Card applet and a Toolkit applet is that the latter does not handle APDUs directly. It will handle higher level messages. Furthermore the execution of a method could span over multiple APDUs, in particular, the proactive protocol commands (Fetch, Terminal Response).

As seen above, when the GSM applet is the selected application and when a toolkit applet is triggered the *select*() method of the toolkit applet shall not be launched since the toolkit applet itself is not really selected.

Here after are the events that can trigger a toolkit applet :

EVENT_FIRST_COMMAND_AFTER_SELECT

Upon reception of the first command received by the GSM~~SIM~~ application after it has been selected or after the ATR if it is the default application, and after the command has been processed by the GSM application, the toolkit framework shall trigger all the toolkit applets registered to this event.

If the first command received by the GSM application is a toolkit applet triggering command (e.g. TERMINAL PROFILE), the toolkit applets registered on the EVENT_FIRST_COMMAND_AFTER_SELECT event shall be triggered first.

EVENT_PROFILE_DOWNLOAD

Upon reception of the Terminal Profile command by the SIM, the SIM Toolkit Framework stores the ME profile and then triggers the registered toolkit applet which may want to change their registry. A toolkit applet may not be able to issue a proactive command.

EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST

A toolkit applet might be activated upon selection in the ME's menu by the user, or request help on this specific menu.

In order to allow the user to choose in a menu, the SIM Toolkit Framework shall have previously issued a SET UP MENU proactive command. When a toolkit applet changes a menu entry of its registry object, the SIM Toolkit Framework shall dynamically update the menu stored in the ME during the current card session. The SIM Toolkit Framework shall use the data of the EFsume file when issuing the SET UP MENU proactive command.

The positions of the toolkit applet menu entries in the item list, the requested item identifiers and the associated limits (e.g. maximum length of item text string) are defined at the loading of the toolkit applet.

If at least one toolkit applet registers to EVENT_MENU_SELECTION_HELP_REQUEST, the SET UP MENU proactive command sent by the SIM Toolkit Framework shall indicate to the ME that help information is available.A toolkit applet registered for one or more menu entries, may be triggered by the event EVENT_MENU_SELECTION_HELP_REQUEST, even if it is not registered to this event. A toolkit applet registered for one or more menu entries should provide help information.

EVENT_FORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_ENV,
EVENT_FORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_PP_UPD

A toolkit applet can be activated upon the reception of a short message.

There are two ways for a card to receive an SMS : via the Envelope SMS-PP Data Download or the Update Record EFsms instruction.

The reception of the SMS by the toolkit applet cannot be guaranteed for the Update Record EFsms instruction.

The received SMS may be :
- formatted according to TS 23.048[4] or an other protocol to identify explicitly the toolkit applet for which the message is sent ;
- unformatted or using a toolkit applet specific protocol the SIM Toolkit Framework will pass this data to all registered toolkit applets.

EVENT_FORMATTED_SMS_PP_ENV

This event is triggered by an envelope APDU containing an SMS_DATADOWNLOAD BER TLV with an SMS_TPDU simple TLV according to TS 23.048[4].

The SIM Toolkit Framework shall:
- verify the TS 23.048[4] security of the SMS TPDU ;
- trigger the toolkit applet registered with the corresponding TAR defined at applet loading;
- take the optional Application Data posted by the triggered toolkit applet if present;
- secure and send the response packet.

The toolkit applet will only be triggered if the TAR is known and the security verified, application data will also be deciphered.

EVENT_UNFORMATTED_SMS_PP_ENV

The registered toolkit applets will be triggered by this event and get the data transmitted in the APDU envelope SMS_DATADOWNLOAD.

But only the first toolkit applet triggered will be able to send back a response as defined by the rules in chapter 6.6.

EVENT_FORMATTED_SMS_PP_UPD

This event is triggered by Update Record EFsms with an SMS TP-UD field formatted according to TS 23.048[4].

The SIM Toolkit Framework shall :
- update the EFsms file with the data received, it is then up to the receiving toolkit applet to change the SMS stored in the file (i.e. the toolkit applet need to have access to the EFsms file)
- verify the TS 23.048[4] security of the SMS TPDU ;
- convert the Update Record EFsms in a TLV List, an EnvelopeHandler ;
- trigger the toolkit applet registered with the corresponding TAR defined at applet loading;

The Update Record EFsms APDU shall be converted in a TLV list as defined below :

UPDATE RECORD APDU	**nb bytes**	**Handler TLV LIST**	**size**
CLA, INS	2	specific event	1
P1,P2	2	device Identity rec-number	1
P3 = 176	1		1
status	1	device Identity rec-status	1
TS-SCA (RP-OA)	<= 12	Address	Y
SMS TPDU	var	SMS TPDU	Y
padding bytes	var		Y

The EnvelopeHandler provided to the applet shall:
- return *BTAG_SMS_PP_DOWNLOAD* to the *getEnvelopeTag()* method call;
- return the Simple TLV list length to the *getLength()* `method call`;
- contain the Simple TLV list :

EnvelopeHandler TLV List
Device identities
Address
SMS TPDU

The applet should use the *findTLV()* methods to get each Simple TLV.

The Device Identity Simple TLV is used to store the information about the absolute record number in the EFsms file and the value of the EFsms record status byte, and formatted as defined below:

Device identities Simple TLV
Device identities tag
length = 02
Absolute Record Number
Record Status

With the absolute record number the toolkit applet can update EFsms in absolute mode to change the received SMS in a readable text.

EVENT_UNFORMATTED_SMS_PP_UPD

The SIM Toolkit Framework will first update the EFsms file, convert the received APDU as described above, and then trigger all the registered toolkit applets. All of them may modify the content of EFsms (i.e. the toolkit applets need to have access to the EFsms file).

EVENT_FORMATTED_SMS_CB, EVENT_UNFORMATTED_SMS_CB

When the ME receives a new cell broadcast message, the cell broadcast page may be passed to the SIM using the envelope command according to the content of the EF_{CBMID} file. E.g. the application may then read the message and extract a meaningful piece of information which could be displayed to the user, for instance.

The received cell broadcast page can be either:
- formatted according to TS 23.048 [4] or an other protocol to identify explicitly the toolkit applet for which the message is sent ;
- unformatted or using a toolkit applet specific protocol the SIM Toolkit Framework will pass this data to all registered toolkit applets.

EVENT_FORMATTED_SMS_CB

This event is triggered by an envelope APDU containing an CELL_BROADCAST_DATADOWNLOAD BER TLV with a Cell Broadcast Page simple TLV according to TS 23.048 [4].

The SIM Toolkit Framework shall:
- verify the TS 23.048[4] security of the Cell Broadcast Page;
- trigger the toolkit applet registered with the corresponding TAR defined at applet loading.

The toolkit applet will only be triggered if the TAR is known and the security verified, application data will also be deciphered.

The TAR value is the same as the one used in the events *EVENT_FORMATTED_SMS_PP_ENV* and *EVENT_FORMATTED_SMS_PP_UPD.*

EVENT_UNFORMATTED_SMS_CB

The registered toolkit applets will be triggered by this event and get the data transmitted in the APDU envelope CELL_BROADCAST_DATADOWNLOAD.

EVENT_CALL_CONTROL_BY_SIM

When the SIM is in call control mode and when the user dials a number, this number is passed to the SIM. Only one toolkit applet can handle the answer to this command: call barred, modified or accepted.

EVENT_EVENT_DOWNLOAD_MT_CALL, EVENT_EVENT_DOWNLOAD_CALL_CONNECTED,
EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED, EVENT_EVENT_DOWNLOAD_LOCATION_STATUS,
EVENT_EVENT_DOWNLOAD_USER_ACTIVITY, EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE,
EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS,
EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION,
EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION

The toolkit applet will be triggered by the registered event download trigger, upon reception of the corresponding Envelope command.

In order to allow the toolkit applet to be triggered by these events, the SIM Toolkit Framework shall have previously issued a SET UP EVENT LIST proactive command. When a toolkit applet changes one or more of these requested events of its registry object, the SIM Toolkit Framework shall dynamically update the event list stored in the ME during the current card session.

EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM

Before sending an SMS MO entered by the user, the SMS is submitted to the SIM. Only one toolkit applet can register to this event

EVENT_TIMER_EXPIRATION

At the registration to this event the toolkit applet gets the reference to its timer. The toolkit applet can then manage the timer, it will be triggered at the reception of the APDU Envelope TIMER EXPIRATION.

The SIM Toolkit Framework shall reply busy to this Envelope APDU if it cannot guaranty to trigger the corresponding toolkit applet.

EVENT_UNRECOGNIZED_ENVELOPE

The applet registered to this event shall be triggered by the framework if the BER-TLV tag contained in the ENVELOPE APDU is not defined in the associated release of TS 11.14 [3] and if no corresponding constant is defined in the list of the ToolkitConstants interface. The unrecognized Envelope event will allow a toolkit applet to handle the evolution of the TS 11.14 specification.

EVENT_STATUS_COMMAND

At reception of a STATUS APDU command, the SIM Toolkit Framework shall trigger the registered toolkit applet.

A range of events is reserved for proprietary usage (from –128 to –1). The use of these events will make the toolkit applet incompatible.

The toolkit applet shall be triggered for the registered events upon reception, and shall be able to access to the data associated to the event using the methods provided by the *sim.toolkit.ViewHandler.EnvelopeHandler* class.

The order of triggering the toolkit applet shall follow the priority level of each toolkit applet defined at its loading. If several toolkit applets have the same priority level, the last loaded toolkit applet takes precedence.

6.6 Handler availability

The system handlers : ProactiveHandler, ProactiveResponseHandler, EnvelopeHandler and EnvelopeResponseHandler are Temporary JCRE Entry Point Object as defined in the Java Card Runtime Environment Specification [8].

The following table describes the minimum availability of the handlers for all the events at the invocation of the processToolkit method of the toolkit applet.

Table 1: Handler availability for each event

EVENT_	Reply busy	ProactiveHandler ProactiveResponseHandler	EnvelopeHandler	EnvelopeResponseHandler	Nb of triggered / registrered Applet
_FORMATTED_SMS_PP_ENV	Y	Y	Y	Y	1 / n (per TAR)
_FORMATTED_SMS_PP_UPD	N	Y	Y	N	1 / n (per TAR)
_UNFORMATTED_SMS_PP_ENV	Y	Y	Y	Y	n / n
_UNFORMATTED_SMS_PP_UPD	N	Y	Y	N	n / n
_FORMATTED_SMS_CB	Y	Y	Y	N	1 / n (per TAR)
_UNFORMATTED_SMS_CB	Y	Y	Y	N	n / n
_MENU_SELECTION	Y	Y	Y	N	1 / n (per Item Id)
_MENU_SELECTION_HELP_REQUEST	Y	Y	Y	N	1 / n (per Item Id)
_CALL_CONTROL	N	Y/N (see Note 2)	Y	Y	1 / 1
_SMS_MO_CONTROL	N	Y/N (see Note 2)	Y	Y	1 / 1
_TIMER_EXPIRATION	Y	Y	Y	N	1/ 8 (per timer) (see Note 1)
_EVENT_DOWNLOAD					
_MT_CALL	Y	Y	Y	N	n / n
_CALL_CONNECTED	Y	Y	Y	N	n / n
_CALL_DISCONNECTED	Y	Y	Y	N	n / n
_LOCATION_STATUS	Y	Y	Y	N	n / n
_USER_ACTIVITY	Y	Y	Y	N	n / n
_IDLE_SCREEN_AVAILABLE	Y	Y	Y	N	n / n
_LANGUAGE_SELECTION	Y	Y	Y	N	n / n
_BROWSER_TERMINATION	Y	Y	Y	N	n / n
_CARD_READER_STATUS	Y	Y	Y	N	n / n
_UNRECOGNISED_ENVELOPE	Y	Y	Y	Y	n / n
_STATUS_COMMAND	N	Y/N (see Note 2)	N	N	n / n
_PROFILE_DOWNLOAD	N	Y/N (see Note 2)	N	N	n / n
_FIRST_COMMAND_AFTER_SELECT	N	N	N	N	n / n

NOTE 1: One toolkit applet can register to several timers, but a timer can only be allocated to one toolkit applet.
NOTE 2: Y/N means that handlers may / may not be available depending whether a proactive session is ongoing.

The following rules define the minimum requirement for the availability of the system handlers and the lifetime of their content.

ProactiveHandler:

- The ProactiveHandler is valid from the invocation to the termination of the processToolkit method.
- If a proactive command is pending the ProactiveHandler may not be available.
- At the processToolkit method invocation the TLV-List is cleared.
- At the call of it's init method the content is cleared and then initialised.
- After a call to ProactiveHandler.send method the handler will remain unchanged (i.e. previously send proactive command) until the ProactiveHandler.init or appendTLV methods are called.

ProactiveResponseHandler:

- The ProactiveResponseHandler may not be available before the first call to ProactiveHandler.send method, if available the content is cleared.
- The ProactiveResponseHandler is available after the first call to the ProactiveHandler.send method to the termination of the processToolkit method.
- If a proactive command is pending the ProactiveResponseHandler may not be available.
- The ProactiveResponseHandler content is changed after the call to ProactiveHandler.send method and remains unchanged until next call to the ProactiveHandler.send method.

EnvelopeHandler:

- The EnvelopeHandler and its content are available for all triggered toolkit applets (see Table1), from the invocation to the termination of their processToolkit method.

- The SIM Toolkit Framework guarantees that all registered toolkit applet are triggered and receive the data.

EnvelopeResponseHandler:

- The EnvelopeResponseHandler is available for all triggered toolkit applets, until a toolkit applet has posted an envelope response or sent a proactive command. After a call to the post method the handler is no longer available.

- The EnvelopeResponseHandler content must be posted before the first invocation of a ProactiveHandler.send method or before the termination of the processToolkit, so that the GSM applet can offer these data to the ME (eg 9Fxx/9Exx). After the first invocation of the ProactiveHandler.send method the EnvelopeResponseHandler is no more available.

The following diagram illustrates these rules.

Applet		Applet 1							Applet 2			
method invocation	*processToolkit*	*init*		*post* *send*		*init* *send*		*termination* *processToolkit*	*init*	*send*		*init*
Envelope Handler												
EnvelopeResponseHandler												
ProactiveHandler												
Proactive ResponseHandler												

Figure 5: Typical handler availability for toolkit applets (see Table 1 for detail)

```
//-------------------------------------------------------------------------------
// PACKAGE DEFINITION
//-------------------------------------------------------------------------------
package sim.toolkit;

//-------------------------------------------------------------------------------
// IMPORTS
//-------------------------------------------------------------------------------
import javacard.framework.*;

/**
 *
 * <code>ToolkitConstants</code> encapsulates constants related to the Toolkit applets.
 *
 * @version 4.0.0
 * @author 3GPP TSG-T WG3, JCF SIM API TF
 */
 public interface ToolkitConstants
{
    // --------------------------- Events Constants -------------------------
    /** Event : Profile Download  = 1                                       */
    public static final byte EVENT_PROFILE_DOWNLOAD                    = (byte)1;
    /** Event : Envelope SMS-PP Data Download (03.48 formatted)  = 2        */
    public static final byte EVENT_FORMATTED_SMS_PP_ENV                = (byte)2;
    /** Event : Update Record EF sms APDU     (03.48 formatted)  = 3        */
    public static final byte EVENT_FORMATTED_SMS_PP_UPD                = (byte)3;
    /** Event : Envelope SMS-PP Data Download unformatted sms  = 4          */
    public static final byte EVENT_UNFORMATTED_SMS_PP_ENV              = (byte)4;
    /** Event : Update Record EFsms APDU      unformatted sms   = 5         */
    public static final byte EVENT_UNFORMATTED_SMS_PP_UPD              = (byte)5;
    /** Event : Cell Broadcast Data Download  = 6                           */
    public static final byte EVENT_UNFORMATTED_SMS_CB                  = (byte)6;
    /** Event : Menu Selection  = 7                                         */
    public static final byte EVENT_MENU_SELECTION                      = (byte)7;
    /** Event : Menu Selection Help Request = 8                             */
```

```
    public static final byte EVENT_MENU_SELECTION_HELP_REQUEST          = (byte)8;
    /** Event : Call Control by SIM = 9                                    */
    public static final byte EVENT_CALL_CONTROL_BY_SIM                  = (byte)9;
    /** Event : MO Short Message Control by SIM = 10                       */
    public static final byte EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM      = (byte)10;
    /** Event : Timer Expiration = 11                                      */
    public static final byte EVENT_TIMER_EXPIRATION                     = (byte)11;
    /** Event : Event Download - MT call type = 12                         */
    public static final byte EVENT_EVENT_DOWNLOAD_MT_CALL               = (byte)12;
    /** Event : Event Download - Call connected type = 13                  */
    public static final byte EVENT_EVENT_DOWNLOAD_CALL_CONNECTED        = (byte)13;
    /** Event : Event Download - Call disconnected type = 14               */
    public static final byte EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED     = (byte)14;
    /** Event : Event Download - Location status type = 15                 */
    public static final byte EVENT_EVENT_DOWNLOAD_LOCATION_STATUS       = (byte)15;
    /** Event : Event Download - User activity type = 16                   */
    public static final byte EVENT_EVENT_DOWNLOAD_USER_ACTIVITY         = (byte)16;
    /** Event : Event Download - Idle screen available type = 17           */
    public static final byte EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE = (byte)17;
    /** Event : Event Download - Card Reader Status = 18                   */
    public static final byte EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS    = (byte)18;
    /** Event : Status APDU command = 19                                   */
    public static final byte EVENT_STATUS_COMMAND                       = (byte)19;
    /** Event : Event Download - Language Selection = 20                   */
    public static final byte EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION    = (byte)20;
    /** Event : Event Download - Browser Termination = 21                  */
    public static final byte EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION   = (byte)21;
    /** Event : Cell Broadcast Data Download Formatted = 24                */
    public static final byte EVENT_FORMATTED_SMS_CB                     = (byte)24;
    /** Event : First Command after GSM application selection = 127     */
    public static final byte EVENT_FIRST_COMMAND_AFTER_SELECT           = (byte)127 ;

    /** Event : Proprietary events reserved range = [-128, -1]             */
    /** Event : Unrecognized Envelope = -1                                 */
    public static final byte EVENT_UNRECOGNIZED_ENVELOPE                = (byte)-1;
```

```
    /* ... */
}
```

3GPP T3 Meeting #21
Kyoto, Japan, 5 - 7 November, 2001

Tdoc T3-010767
was T3-010683

CR-Form-v3

CHANGE REQUEST

⌘ **43.019** **CR** **004** ⌘ rev **-** ⌘ Current version: 5.0.0 ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Update the list of Simple TLV tags in sim.toolkit.ToolkitConstants
Source:	⌘	T3
Work item code:	⌘	SIM API
Date:	⌘	07/11/2001
Category:	⌘	**C**
Release:	⌘	REL-5

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	List of constant values in ToolkitConstants is not up to date with 11.14
Summary of change:	⌘	Introduction of 2 new constant values into ToolkitConstant interface
Consequences if not approved:	⌘	No standard constant values for the new TLV tags

Clauses affected:	⌘	Annex A		
Other specs affected:	⌘	[ ] Other core specifications [ ] Test specifications [ ] O&M Specifications	⌘	
Other comments:	⌘			

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.
3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Add to sim.toolkit.ToolkitConstants

```
/** Simple-TLV : Other address (data destination address)        = 0x3E */
public static final byte TAG_OTHER_ADDRESS                       = (byte)0x3E;
/** Simple-TLV : Network Access Name                             = 0x47 */
public static final byte TAG_NETWORK_ACCESS_NAME                 = (byte)0x47;
```

3GPP T3 Meeting #21
Kyoto, Japan, 5 - 7 November, 2001

Tdoc T3-010768
revised T3-010684

CR-Form-v3

CHANGE REQUEST

⌘ **43.019** **CR** **005** ⌘ rev **-** ⌘ Current version: **5.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	ToolkitRegistry methods modification
Source:	⌘	T3
Work item code:	⌘	***Date:*** ⌘ 07/11/01
Category:	⌘	**C** ***Release:*** ⌘ REL-5

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	Mandate atomicity on operations performed on ToolkitRegistry object Enhancement : allow method of ToolkitRegistry class to be part of a transaction
Summary of change:	⌘	- A note is added in the ToolkitRegistry description to add atomicity. - All methods of the ToolkitRegistry can now throw TransactionException exception.
Consequences if not approved:	⌘	

Clauses affected:	⌘	Annex_A_java.zip, Annex_A_HTML.zip
Other specs Affected:	⌘	[ ] Other core specifications ⌘ [ ] Test specifications [ ] O&M Specifications
Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

```
/**
 *
 * The <code>ToolkitRegistry</code> class offers basic services and methods to allow
 * any Toolkit applet to register its configuration (supported events) during
 * the install phase and possibly to change it during all the applet life time.
 * Each toolkit applet will get a reference to its registry entry with the static
 * method <code>getEntry</code>. The initial state of all the events is cleared.<p>
 * All updates performed on the toolkit registry are atomic.
 * The toolkit registry update operations are subject to atomic commit capacity limitations.
 * If the commit capacity is exceeded, no update is performed and a TransactionException exception is thrown. <p>
 * Note: the constants related to the events are defined in the <code>ToolkitConstants
 * </code> interface.<p>
 *
 * Example of use:<pre><code>
[…]

   /**
    * Sets an event in the Toolkit Registry entry of the applet.
    * No exception shall be thrown if the applet registers more than once to the same event.
    *
    * @param event value of the new event to register (between -128 and 127)
    *
    * @exception ToolkitException with the following reason codes: <ul>
    *     <li>EVENT_NOT_SUPPORTED if the event is not supported
    *     <li>EVENT_ALREADY_REGISTERED if the event has already been registered
    *       (for limited event like Call Control)
    *        <li>EVENT_NOT_ALLOWED if event is EVENT_MENU_SELECTION,
    *            EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION,
EVENT_STATUS_COMMAND </ul>
    */
   public void setEvent(byte event) throws ToolkitException {
   }

    /**
    * Sets an event list in the Toolkit Registry entry of the applet.
    * In case of any exception the state of the registry is undefined. The toolkit applet has to include this
    * call within a transaction if necessary.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>offset</code><em> or </em><code>length</code><em> parameter is negative an
</em><code>ArrayIndexOutOfBoundsException</code>
    * <em> exception is thrown and no event list is set.</em>
    * <li><em>If </em><code>offset+length</code><em> is greater than </em><code>eventList.length</code><em>,
the length
    * of the </em><code>eventList</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no event list is set.</em>
    * </ul>
    *
    * @param eventList buffer containing the list of the new events to register
    * @param offset offset in the eventlist buffer for event registration
    * @param length length in the eventlist buffer for event registration
    *
    * @exception NullPointerException if <code>eventlist</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if setEventList would cause access of data outside array bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *     <li>EVENT_NOT_SUPPORTED if one event is not supported
    *     <li>EVENT_ALREADY_REGISTERED if one event has already been registered
    *     (for limited event like Call Control)
    *       <li>EVENT_NOT_ALLOWED if eventList contains EVENT_MENU_SELECTION,
```

```
    *       EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION,
EVENT_STATUS_COMMAND </ul>
    */
   public void setEventList(byte[] eventList,
                            short offset,
                            short length) throws NullPointerException,
                                    ArrayIndexOutOfBoundsException,
                                    ToolkitException,
                                    TransactionException {
   }

   /**
    * Clears an event in the Toolkit Registry entry of the applet.
    *
    * @param event the value of the event to unregister (between -128 and 127)
    * @exception ToolkitException with the following reason codes: <ul>
    *       <li>EVENT_NOT_ALLOWED if event is EVENT_MENU_SELECTION,
    *       EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION,
EVENT_STATUS_COMMAND</ul>
    * @exception TransactionException - if the operation would cause the commit capacity to be exceeded.
    */
   public void clearEvent(byte event) throws ToolkitException,
                                    TransactionException {
   }

[…]

   /**
    * Disables a menu entry.
    * This method doesn't modify the registration state to the EVENT_MENU_SELECTION
    * and EVENT_MENU_SELECTION_HELP_REQUEST. After invocation of this method, during the current card
session, the SIM
    * Toolkit Framework shall dynamically update the menu stored in the ME.
    *
    * @param id the menu entry identifier supplied by the <code>initMenuEntry()</code> method
    *
    * @exception ToolkitException with the following reason codes: <ul>
    *     <li>ENTRY_NOT_FOUND if the menu entry does not exist for this applet</ul>
    * @exception TransactionException - if the operation would cause the commit capacity to be exceeded.
    */
   public void disableMenuEntry(byte id) throws ToolkitException,
                                    TransactionException {
   }

   /**
    * Enables a menu entry.
    * This method doesn't modify the registration state to the EVENT_MENU_SELECTION
    * and EVENT_MENU_SELECTION_HELP_REQUEST. After invocation of this method, during the current card
session, the SIM
    * Toolkit Framework shall dynamically update the menu stored in the ME.
    *
    * @param id the menu entry identifier supplied by the <code>initMenuEntry()</code> method
    *
    * @exception ToolkitException with the following reason codes: <ul>
    *     <li>MENU_ENTRY_NOT_FOUND if the menu entry does not exist for this applet</ul>
    * @exception TransactionException - if the operation would cause the commit capacity to be exceeded.
    */
   public void enableMenuEntry(byte id) throws ToolkitException,
                                    TransactionException {
   }

   /**
```

```
    * Initialises the next menu entry allocated at loading. The default state of the menu entry is
    * 'enabled'. The value of the <code>helpSupported</code> boolean parameter
    * defines the registration status of the applet to the event
    * EVENT_MENU_SELECTION_HELP_REQUEST. The applet is registered to
    * the EVENT_MENU_SELECTION. The icon identifier provided will be added to
    * the icon identifier list of the item icon identifier list Simple TLV if
    * all the applets registered to the EVENT_MENU_SELECTION provide it.
    * The Icon list qualifier transmitted to the ME will be 'icon is not self
    * explanatory' if one of the applet registered prefers this qualifier.
    * This method shall be called by the applet in the same order than the
    * order of the item parameters defined at the applet loading if the applet
    * has several menu entries. The applet shall initialise all its loaded
    * menu entries during its installation.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>offset</code><em> or </em><code>length</code><em> parameter is negative an
</em><code>ArrayIndexOutOfBoundsException</code>
    * <em> exception is thrown and no menu entry is initialised.</em>
    * <li><em>If </em><code>offset+length</code><em> is greater than
</em><code>menuEntry.length</code><em>, the length
    * of the </em><code>menuEntry</code><em> array a
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no menu entry is initialised.</em>
    * </ul>
    *
    * @param menuEntry a reference on a byte array, containing the menu entry string
    * @param offset offset of the menu entry string in the buffer
    * @param length length of the menu entry string
    * @param nextAction a byte coding the next action indicator for the menu entry (or 0)
    * @param helpSupported equals true if help is available for the menu entry
    * @param iconQualifier the preferred value for the icon list qualifier
    * @param iconIdentifier the icon identifier for the menu entry  (0 means no icon)
    *
    * @return the identifier attached to the initialised menu entry
    *
    * @exception NullPointerException if <code>menuEntry</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if initMenuEntry would cause access of data outside array bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *     <li>REGISTRY_ERROR if the menu entry cannot be initialised (eg no more item data in applet loading
parameter)
    *       <li>ALLOWED_LENGTH_EXCEEDED if the menu entry string is bigger than the alloacted space</ul>
    * @exception TransactionException - if the operation would cause the commit capacity to be exceeded.
    */
   public byte initMenuEntry(byte[] menuEntry,
                             short offset,
                             short length,
                             byte nextAction,
                             boolean helpSupported,
                             byte iconQualifier,
                             short iconIdentifier) throws NullPointerException,
                                                          ArrayIndexOutOfBoundsException,
                                                          ToolkitException,
                                                          TransactionException {
       return 0;
   }

   /**
    * Changes the value of a menu entry. The default state of the changed menu
    * entry is 'enabled'. The value of the <code>helpSupported</code> boolean
    * parameter defines the registration status of the EVENT_MENU_SELECTION_HELP_REQUEST
    * event. The icon identifier provided will be added to the icon identifier list
```

```
    * of the item icon identifier list Simple TLV if all the applets registered
    * to the EVENT_MENU_SELECTION provide it.
    * The Icon list qualifier transmitted to the ME will be 'icon is not self
    * explanatory' if one of the applet registered prefers this qualifier.
    * After the invocation of this method, during the current card session, the SIM Toolkit Framework
    * shall dynamically update the menu stored in the ME.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>offset</code><em> or </em><code>length</code><em> parameter is negative an
</em><code>ArrayIndexOutOfBoundsException</code>
    * <em> exception is thrown and no menu entry is changed.</em>
    * <li><em>If </em><code>offset+length</code><em> is greater than
</em><code>menuEntry.length</code><em>, the length
    * of the </em><code>menuEntry</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no menu entry is changed.</em>
    * </ul>
    *
    * @param id the menu entry identifier supplied by the <code>initMenuEntry()</code> method
    * @param menuEntry a reference on a byte array, containing the menu entry string
    * @param offset the position of the menu entry string in the buffer
    * @param length the length of the menu entry string
    * @param nextAction a byte coding the next action indicator for the menu entry (or 0)
    * @param helpSupported equals true if help is available for the menu entry
    * @param iconQualifier the preferred value for the icon list qualifier
    * @param iconIdentifier the icon identifier for the menu entry  (0 means no icon)
    *
    * @exception NullPointerException if <code>menuEntry</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if changeMenuEntry would cause access of data outside array
bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li>MENU_ENTRY_NOT_FOUND if the menu entry does not exist for this applet
    *       <li>ALLOWED_LENGTH_EXCEEDED if the menu entry string is bigger than the alloacted space</ul>
    * @exception TransactionException - if the operation would cause the commit capacity to be exceeded.
    */
   public void changeMenuEntry(byte id,
                               byte[] menuEntry,
                               short offset,
                               short length,
                               byte nextAction,
                               boolean helpSupported,
                               byte iconQualifier,
                               short iconIdentifier) throws NullPointerException,
                                                            ArrayIndexOutOfBoundsException,
                                                            ToolkitException,
                                                            TransactionException {
   }

   /**
    * Asks the Toolkit framework to allocate a Timer that the applet can manage.
    * By calling this method the applet is registered to the
    * EVENT_TIMER_EXPIRATION of the allocated timer.
    * The timer is allocated by the applet until it explicitly releases it.
    * So it can then issue the Timer Management proactive command to start,
    * stop or get the value of its allocated timer.
    *
    * @return the identifier of the Timer allocated to the applet
    *
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li>NO_TIMER_AVAILABLE if all the timers are allocated or the maximum number
    *          of timers have been allocated to this applet</ul>
```

```
     * @exception TransactionException - if the operation would cause the commit capacity to be exceeded.
     */
    public byte allocateTimer() throws ToolkitException,
                                       TransactionException {
        return 0;
    }

    /**
     * Release a Timer that has been allocated to the calling applet.
     * The applet is deregistered of the EVENT_TIMER_EXPIRATION
     * for the indicated Timer Identifier.
     *
     * @param timerIdentifier the identifier of the Timer to be released
     *
     * @exception ToolkitException with the following reason codes: <ul>
     *      <li>INVALID_TIMER_ID if the timerIdentifierd is not allocated to this applet.</ul>
     * @exception TransactionException - if the operation would cause the commit capacity to be exceeded.
     */
    public void releaseTimer(byte timerIdentifier) throws ToolkitException,
                                       TransactionException {
    }


    /**
     * Requests a duration for the EVENT_STATUS_COMMAND event of the registering
     * toolkit applet. Due to different duration requested by other toolkit
     * applets or due to restriction of the ME, the SIM Toolkit Framework may
     * adjust another duration.
     * This method can be used at every time to request a new duration.
     *
     * @param duration specifies the number of seconds requested for proactive polling.
     *          The maximum value of <code>duration</code> is <code>15300</code> (255 minutes).
     *          If <code>duration<code> is equal to <code>POLL_NO_DURATION</code>,
     *          the calling applet deregisters from EVENT_STATUS_COMMAND, and
     *          the SIM Toolkit Framework may issue the POLLING OFF proactive command.
     *          If <code>duration<code> is equal to <code>POLL_SYSTEM_DURATION</code>,
     *          the calling applet registers to the EVENT_STATUS_COMMAND and let
     *          the SIM Toolkit Framework define the duration.
     *
     * @exception ToolkitException with the following reason codes: <ul>
     *      <li>REGISTRY_ERROR   if <code>duration</code> is greater than the maximum value.</ul>
     * @exception TransactionException - if the operation would cause the commit capacity to be exceeded.
     */
    public void requestPollInterval(short duration) throws ToolkitException,
                                       TransactionException {
    }
```

[…]

3GPP T3 Meeting #21
Kyoto, Japan, 5 - 7 November, 2001

Tdoc T3-010769
revised T3-010685

CR-Form-v3

CHANGE REQUEST

⌘ **43.019** **CR** **006** ⌘ rev **-** ⌘ Current version: **5.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	ToolkitRegistry methods modification when no TAR is defined
Source:	⌘	T3
Work item code:	⌘	***Date:*** ⌘ 07/11/01
Category:	⌘	C ***Release:*** ⌘ REL-5

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	Define standard behaviour if TAR is not defined.
Summary of change:	⌘	- Add a reason in the ToolkitException class and setEvent and setEventList throws exception if there is no TAR defined for the applet and if the applet requests an event which needs a TAR.
Consequences if not approved:	⌘	

Clauses affected:	⌘	Annex_A_java.zip, Annex_A_HTML.zip
Other specs Affected:	⌘	[ ] Other core specifications ⌘ [ ] Test specifications [ ] O&M Specifications
Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

ToolkitRegistry Class

```
[…]
   /**
    * Sets an event in the Toolkit Registry entry of the applet.
    * No exception shall be thrown if the applet registers more than once to the same event.
    *
    * @param event value of the new event to register (between -128 and 127)
    *
    * @exception ToolkitException with the following reason codes: <ul>
    *     <li>EVENT_NOT_SUPPORTED if the event is not supported
    *     <li>EVENT_ALREADY_REGISTERED if the event has already been registered
    *       (for limited event like Call Control)
    *       <li>EVENT_NOT_ALLOWED if event is EVENT_MENU_SELECTION,
    *           EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION,
EVENT_STATUS_COMMAND </ul>
    *     <li>TAR_NOT_DEFINED if event is FORMATTED_SMS_PP_ENV, FORMATTED_SMS_PP_UPD or
FORMATTED_SMS_CB_ENV and the applet has no TAR defined
    */
   public void setEvent(byte event) throws ToolkitException {
   }

    /**
    * Sets an event list in the Toolkit Registry entry of the applet.
    * In case of any exception the state of the registry is undefined. The toolkit applet has to include this
    * call within a transaction if necessary.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>offset</code><em> or </em><code>length</code><em> parameter is negative an
</em><code>ArrayIndexOutOfBoundsException</code>
    * <em> exception is thrown and no event list is set.</em>
    * <li><em>If </em><code>offset+length</code><em> is greater than </em><code>eventList.length</code><em>,
the length
    * of the </em><code>eventList</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no event list is set.</em>
    * </ul>
    *
    * @param eventList buffer containing the list of the new events to register
    * @param offset offset in the eventlist buffer for event registration
    * @param length length in the eventlist buffer for event registration
    *
    * @exception NullPointerException if <code>eventlist</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if setEventList would cause access of data outside array bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *     <li>EVENT_NOT_SUPPORTED if one event is not supported
    *     <li>EVENT_ALREADY_REGISTERED if one event has already been registered
    *     (for limited event like Call Control)
    *       <li>EVENT_NOT_ALLOWED if eventList contains EVENT_MENU_SELECTION,
    *       EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION,
EVENT_STATUS_COMMAND </ul>
    *     <li>TAR_NOT_DEFINED if eventList contains FORMATTED_SMS_PP_ENV,
FORMATTED_SMS_PP_UPD or FORMATTED_SMS_CB_ENV and the applet has no TAR defined
    */
   public void setEventList(byte[] eventList,
                            short offset,
                            short length) throws NullPointerException,
                                    ArrayIndexOutOfBoundsException,
                                    ToolkitException {
   }
```

[…]

ToolkitException Class

```
    /** This reason code (=14) is used to indicate that an input parameter of
     * the method is not valid
     */
    public static final short BAD_INPUT_PARAMETER       = (short)14;
    /** This reason code (=15) is used to indicate that there is no TAR
     * defined for the applet.
     */
    public static final short TAR_NOT_DEFINED       = (short)15;
```

3GPP T3 SWG API Meeting #9
Madrid, Spain, 29 - 31 October, 2001

Tdoc T3-010770

CR-Form-v3

CHANGE REQUEST

⌘ **43.019** **CR** **007** ⌘ rev **-** ⌘ Current version: **5.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Change of the behavior of the framework in case of Event Menu Selection Help Request
Source:	⌘	T3
Work item code:	⌘	***Date:*** ⌘ 19/09/01
Category:	⌘	**C** ***Release:*** ⌘ REL-5

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	Interoperability issue
Summary of change:	⌘	Clarify the activation of Toolkit applet on the EVENT_MENU_SELECTION_HELP_REQUEST event.
Consequences if not approved:	⌘	No standard solution to provide such a service.

Clauses affected:	⌘	§ 6.2		
Other specs Affected:	⌘	[ ] Other core specifications [ ] Test specifications [ ] O&M Specifications	⌘	
Other comments:	⌘			

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.
3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

6.2 Applet Triggering

The application triggering portion of the SIM Toolkit Framework is responsible for the activation of toolkit applets, based on the APDU received by the GSM application.

Figure 3: toolkit applet triggering diagram

The ME shall not be adversely affected by the presence of applets on the SIM card. For instance a syntactically correct Envelope shall not result in an error status word in case of a failure of an applet. The only application as seen by the ME is the SIM application. As a result, a toolkit applet may throw an exception, but this error will not be sent to the ME.

The difference between a Java Card applet and a Toolkit applet is that the latter does not handle APDUs directly. It will handle higher level messages. Furthermore the execution of a method could span over multiple APDUs, in particular, the proactive protocol commands (Fetch, Terminal Response).

As seen above, when the GSM applet is the selected application and when a toolkit applet is triggered the *select*() method of the toolkit applet shall not be launched since the toolkit applet itself is not really selected.

Here after are the events that can trigger a toolkit applet :

EVENT_PROFILE_DOWNLOAD

Upon reception of the Terminal Profile command by the SIM, the SIM Toolkit Framework stores the ME profile and then triggers the registered toolkit applet which may want to change their registry. A toolkit applet may not be able to issue a proactive command.

EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST

A toolkit applet might be activated upon selection in the ME's menu by the user, or request help on this specific menu.

In order to allow the user to choose in a menu, the SIM Toolkit Framework shall have previously issued a SET UP MENU proactive command. When a toolkit applet changes a menu entry of its registry object, the SIM Toolkit Framework shall dynamically update the menu stored in the ME during the current card session. The SIM Toolkit Framework shall use the data of the EFsume file when issuing the SET UP MENU proactive command.

The positions of the toolkit applet menu entries in the item list, the requested item identifiers and the associated limits (e.g. maximum length of item text string) are defined at the loading of the toolkit applet.

If at least one Menu Id of a toolkit applet registers to EVENT_MENU_SELECTION_HELP_REQUEST, the SET UP MENU proactive command sent by the SIM Toolkit Framework shall indicate to the ME that help information is available.~~A toolkit applet registered for one or more menu entries, may be triggered by the event EVENT_MENU_SELECTION_HELP_REQUEST, even if it is not registered to this event. A toolkit applet registered for one or more menu entries should provide help information.~~

A toolkit applet shall be triggered by the EVENT_MENU_SELECTION_HELP_REQUEST event only if the Menu Id corresponding to the Envelope Menu Selection Help Request received by the SIM Tookit framework was registered with the *helpSupported* value set to true.

EVENT_FORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_PP_UPD

A toolkit applet can be activated upon the reception of a short message.

There are two ways for a card to receive an SMS : via the Envelope SMS-PP Data Download or the Update Record EFsms instruction.

The reception of the SMS by the toolkit applet cannot be guaranteed for the Update Record EFsms instruction.

The received SMS may be :
- formatted according to TS 23.048[4] or an other protocol to identify explicitly the toolkit applet for which the message is sent ;
- unformatted or using a toolkit applet specific protocol the SIM Toolkit Framework will pass this data to all registered toolkit applets.

EVENT_FORMATTED_SMS_PP_ENV

This event is triggered by an envelope APDU containing an SMS_DATADOWNLOAD BER TLV with an SMS_TPDU simple TLV according to TS 23.048[4].

The SIM Toolkit Framework shall:
- verify the TS 23.048[4] security of the SMS TPDU ;
- trigger the toolkit applet registered with the corresponding TAR defined at applet loading;
- take the optional Application Data posted by the triggered toolkit applet if present;
- secure and send the response packet.

The toolkit applet will only be triggered if the TAR is known and the security verified, application data will also be deciphered.

EVENT_UNFORMATTED_SMS_PP_ENV

The registered toolkit applets will be triggered by this event and get the data transmitted in the APDU envelope SMS_DATADOWNLOAD.

But only the first toolkit applet triggered will be able to send back a response as defined by the rules in chapter 6.6.

EVENT_FORMATTED_SMS_PP_UPD

This event is triggered by Update Record EFsms with an SMS TP-UD field formatted according to TS 23.048[4].

The SIM Toolkit Framework shall :
- update the EFsms file with the data received, it is then up to the receiving toolkit applet to change the SMS stored in the file (i.e. the toolkit applet need to have access to the EFsms file)
- verify the TS 23.048[4] security of the SMS TPDU ;
- convert the Update Record EFsms in a TLV List, an EnvelopeHandler ;
- trigger the toolkit applet registered with the corresponding TAR defined at applet loading;

The Update Record EFsms APDU shall be converted in a TLV list as defined below :

UPDATE RECORD APDU	nb bytes	Handler TLV LIST	size
CLA, INS	2	specific event	1
P1,P2	2	device Identity rec-number	1
P3 = 176	1		1
status	1	device Identity rec-status	1
TS-SCA (RP-OA)	<= 12	Address	Y
SMS TPDU	var	SMS TPDU	Y
padding bytes	var		Y

The EnvelopeHandler provided to the applet shall:
- return *BTAG_SMS_PP_DOWNLOAD* to the *getEnvelopeTag()* method call;
- return the Simple TLV list length to the *getLength()* `method call`;
- contain the Simple TLV list :

EnvelopeHandler TLV List
Device identities
Address
SMS TPDU

The applet should use the *findTLV()* methods to get each Simple TLV.

The Device Identity Simple TLV is used to store the information about the absolute record number in the EFsms file and the value of the EFsms record status byte, and formatted as defined below:

Device identities Simple TLV
Device identities tag
length = 02
Absolute Record Number
Record Status

With the absolute record number the toolkit applet can update EFsms in absolute mode to change the received SMS in a readable text.

EVENT_UNFORMATTED_SMS_PP_UPD

The SIM Toolkit Framework will first update the EFsms file, convert the received APDU as described above, and then trigger all the registered toolkit applets. All of them may modify the content of EFsms (i.e. the toolkit applets need to have access to the EFsms file).

EVENT_FORMATTED_SMS_CB, EVENT_UNFORMATTED_SMS_CB

When the ME receives a new cell broadcast message, the cell broadcast page may be passed to the SIM using the envelope command according to the content of the EF_{CBMID} file. E.g. the application may then read the message and extract a meaningful piece of information which could be displayed to the user, for instance.

The received cell broadcast page can be either:
- formatted according to TS 23.048 [4] or an other protocol to identify explicitly the toolkit applet for which the message is sent ;
- unformatted or using a toolkit applet specific protocol the SIM Toolkit Framework will pass this data to all registered toolkit applets.

EVENT_FORMATTED_SMS_CB

This event is triggered by an envelope APDU containing an CELL_BROADCAST_DATADOWNLOAD BER TLV with a Cell Broadcast Page simple TLV according to TS 23.048 [4].

The SIM Toolkit Framework shall:
- verify the TS 23.048[4] security of the Cell Broadcast Page;
- trigger the toolkit applet registered with the corresponding TAR defined at applet loading.

The toolkit applet will only be triggered if the TAR is known and the security verified, application data will also be deciphered.

The TAR value is the same as the one used in the events *EVENT_FORMATTED_SMS_PP_ENV* and *EVENT_FORMATTED_SMS_PP_UPD.*

EVENT_UNFORMATTED_SMS_CB

The registered toolkit applets will be triggered by this event and get the data transmitted in the APDU envelope CELL_BROADCAST_DATADOWNLOAD.

EVENT_CALL_CONTROL_BY_SIM

When the SIM is in call control mode and when the user dials a number, this number is passed to the SIM. Only one toolkit applet can handle the answer to this command: call barred, modified or accepted.

EVENT_EVENT_DOWNLOAD_MT_CALL, EVENT_EVENT_DOWNLOAD_CALL_CONNECTED, EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED, EVENT_EVENT_DOWNLOAD_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_USER_ACTIVITY, EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE, EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS, EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION, EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION

The toolkit applet will be triggered by the registered event download trigger, upon reception of the corresponding Envelope command.

In order to allow the toolkit applet to be triggered by these events, the SIM Toolkit Framework shall have previously issued a SET UP EVENT LIST proactive command. When a toolkit applet changes one or more of these requested events of its registry object, the SIM Toolkit Framework shall dynamically update the event list stored in the ME during the current card session.

EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM

Before sending an SMS MO entered by the user, the SMS is submitted to the SIM. Only one toolkit applet can register to this event

EVENT_TIMER_EXPIRATION

At the registration to this event the toolkit applet gets the reference to its timer. The toolkit applet can then manage the timer, it will be triggered at the reception of the APDU Envelope TIMER EXPIRATION.

The SIM Toolkit Framework shall reply busy to this Envelope APDU if it cannot guaranty to trigger the corresponding toolkit applet.

EVENT_UNRECOGNIZED_ENVELOPE

The applet registered to this event shall be triggered by the framework if the BER-TLV tag contained in the ENVELOPE APDU is not defined in the associated release of TS 11.14 [3] and if no corresponding constant is defined in the list of the ToolkitConstants interface. The unrecognized Envelope event will allow a toolkit applet to handle the evolution of the TS 11.14 specification.

EVENT_STATUS_COMMAND

At reception of a STATUS APDU command, the SIM Toolkit Framework shall trigger the registered toolkit applet.

A range of events is reserved for proprietary usage (from –128 to –1). The use of these events will make the toolkit applet incompatible.

The toolkit applet shall be triggered for the registered events upon reception, and shall be able to access to the data associated to the event using the methods provided by the *sim.toolkit.ViewHandler.EnvelopeHandler* class.

The order of triggering the toolkit applet shall follow the priority level of each toolkit applet defined at its loading. If several toolkit applets have the same priority level, the last loaded toolkit applet takes precedence.

3GPP T3 Meeting #21
Kyoto, Japan, 5 - 7 November, 2001

Tdoc T3-010791

CR-Form-v3

CHANGE REQUEST

⌘ **43.019** **CR** 008 ⌘ rev **-** ⌘ Current version: 8.2.0 ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Clarification of ToolkitException.OUT_OF_TLV_BOUNDARIES in ViewHandler.java
Source:	⌘	T3
Work item code:	⌘	SIM Toolkit API
Date:	⌘	07/11/2001
Category:	⌘	**F**
Release:	⌘	REL-4

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
REL-4 (Release 4)
REL-5 (Release 5)

Reason for change: ⌘ Currently the ToolkitException OUT_OF_TLV_BOUNDARIES is thrown if a TLV object with length 0 is accessed.
Clarify the enumeration of the different cases for the ToolkitException OUT_OF_TLV_BOUNDARIES.

Summary of change: ⌘ - Clarify rule when ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown
- Allow access to a zero length substring at the end of a TLV element

The current specification is unclear when exactly to throw a OUT_OF_TLV_BOUNDARIES exception. The term currently used is "Throw OUT_OF_TLV_BOUNDARIES if valueOffset, compareLength or both are out of the current TLV".

This is unclear, especially when it comes to the special case of accessing zero bytes at the end of the TLV element. Is it allowed, for example to copy zero bytes from the end of the element? As an example, with the current spec it is probably not possible to write the following statement, if a secured Envelope with zero length secured data is received.

```
envHandler.copyValue( envHdlr.getSecuredDataOffset(),
                      dest,
                      destOffset,
                      envHdlr.getSecuredDataLength() );
```

The Applet has to test for that special case beforehand.
We propose to make the condition when to throw an exception compliant to the standard array access rules used in javacard.framework.

		See Util.arrayCopy(...) as an example
Consequences if not approved:	⌘	In each implementation the case “Length of TLV object = 0” must be checked and processed separately.

Clauses affected:	⌘	Annex A -> ViewHandler.java:
		Methods affected:
		`public short copyValue(short valueOffset, byte destBuffer[], short destOffset, short destLength ) throws [..] ToolkitException`
		
		`public byte compareValue(short valueOffset, byte compareBuffer[], short compareOffset, short compareLength ) throws [..] ToolkitException`
		
		`public short findAndCopyValue(byte tag, byte occurence, short valueOffset, byte destBuffer[], short destOffset, short destLength) throws [..] ToolkitException`
		
		`public byte findAndCompareValue(byte tag, byte occurence, short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength) throws [..] ToolkitException`

Other specs Affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

```
/**
    * Copies a part of the last TLV element which has been found, into a
    * destination buffer.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>dstOffset</code><em> or </em><code>dstLength</code><em> parameter is
negative an </em><code>ArrayIndexOutOfBoundsException</code>
    * <em> exception is thrown and no copy is performed.</em>
    * <li><em>If </em><code>dstOffset+dstLength</code><em> is greater than
</em><code>dstBuffer.length</code><em>, the length
    * of the </em><code>dstBuffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no copy is performed.</em>
    * </ul>
    *
    * @param valueOffset the offset of the first byte in the source TLV element
    * @param dstBuffer a reference to the destination buffer
    * @param dstOffset the position in the destination buffer
    * @param dstLength the data length to be copied
    *
    * @return <code>dstOffset+dstLength</code>
    *
    * @exception NullPointerException if <code>dstBuffer</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if copyValue would cause access of data outside array
bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
    *      <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element
    *      <li><code>OUT_OF_TLV_BOUNDARIES</code> if: ~~<code>valueOffset</code>,
<code>dstLength</code> or both are out of the current TLV </ul>~~
    *      <ul>
    *          <li><code>valueOffset</code> parameter is negative or
    *          <li><code>valueOffset + dstLength</code> is greater than the length of the current TLV
    *      </ul>
    *   </ul>
    */
   public short copyValue( short valueOffset,
                           byte[] dstBuffer,
                           short dstOffset,
                           short dstLength) throws NullPointerException,
                                                   ArrayIndexOutOfBoundsException,
                                                   ToolkitException {
                           return 0;
   }

/**
    * Compares the last found TLV element with a buffer.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>compareOffset</code><em> or </em><code>compareLength</code><em>
parameter is negative an </em><code>ArrayIndexOutOfBoundsException</code>
    * <em> exception is thrown and no compare is performed.</em>
    * <li><em>If </em><code>compareOffset+compareLength</code><em>is greater than
</em><code>compareBuffer.length</code><em>, the length
    * of the </em><code>compareBuffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no compare is performed.</em>
    * </ul>
    *
    * @param valueOffset the offset of the first byte to compare in the TLV element
```

```
    * @param compareBuffer a reference to the comparison buffer
    * @param compareOffset the position in the comparison buffer
    * @param compareLength the length to be compared
    *
    * @return the result of the comparison as follows: <ul>
    *      <li><code>0</code> if identical
    *      <li><code>-1</code> if the first miscomparing byte in simple TLV List is less than that in
<code>compareBuffer</code>,
    *      <li><code>1</code> if the first miscomparing byte in simple TLV List is greater than that in
<code>compareBuffer</code>.</ul>
    *
    * @exception NullPointerException if <code>compareBuffer</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if compareValue would cause access of data outside array
bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
    *      <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element
    *      <li><code>OUT_OF_TLV_BOUNDARIES</code> if:
    *       <ul>
    *           <li><code>valueOffset</code> parameter is negative or
    *           <li><code>valueOffset + dstLength</code> is greater than the length of the current TLV
    *       </ul>
    *   </ul> <code>valueOffset</code>, <code>compareLength</code> or both are out of the current TLV
</ul>
    */
   public byte compareValue(short valueOffset,
                            byte[] compareBuffer,
                            short compareOffset,
                            short compareLength) throws     NullPointerException,
                                                            ArrayIndexOutOfBoundsException,
                                                            ToolkitException {
                        return 0;
   }

   /**
    * Looks for the first occurence of a TLV element from the beginning of a TLV
    * list and copy its value into a destination buffer.
    * If no TLV element is found, the <code>UNAVAILABLE_ELEMENT</code> exception is thrown.
    * If the method is successful then the corresponding TLV becomes current,
    * else no TLV is selected.
    * This search method is Comprehension Required flag independent.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>dstOffset</code><em> parameter is negative or </em><code>dstOffset</code>
    * <em> is greater than </em><code>dstBuffer.length</code><em>, the length of the
</em><code>dstBuffer</code>
    * <em> array an </em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown and no
find is performed.</em>
    * </ul>
    *
    * @param tag the tag of the TLV element to search
    * @param dstBuffer a reference to the destination buffer
    * @param dstOffset the position in the destination buffer
    *
    * @return <code>dstOffset</code> + length of the copied value
    *
    * @exception NullPointerException if <code>dstBuffer</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if findAndCopyValue would cause access of data outside
array bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
```

```
     *     <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element</ul>
     */
    public short findAndCopyValue(byte tag,
                                  byte[] dstBuffer,
                                  short dstOffset) throws       NullPointerException,
                                                                ArrayIndexOutOfBoundsException,
                                                                ToolkitException {
                        return 0;
    }

    /**
     * Looks for the indicated occurence of a TLV element from the beginning of a TLV
     * list and copy its value into a destination buffer.
     * If no TLV element is found, the <code>UNAVAILABLE_ELEMENT</code> exception is thrown.
     * If the method is successful then the corresponding TLV becomes current,
     * else no TLV is selected.
     * This search method is Comprehension Required flag independent.
     *
     * <p>
     * Notes:<ul>
     * <li><em>If </em><code>dstOffset</code><em> or </em><code>dstLength</code><em> parameter is
negative an </em><code>ArrayIndexOutOfBoundsException</code>
     * <em> exception is thrown and no copy is performed.</em>
     * <li><em>If </em><code>dstOffset+dstLength</code><em>is greater than
</em><code>dstBuffer.length</code><em>, the length
     * of the </em><code>dstBuffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
     * and no copy is performed.</em>
     * </ul>
     *
     * @param tag the tag of the TLV element to search
     * @param occurrence the occurrence number of the TLV element (1 for the first, 2 for the second...)
     * @param valueOffset the offset of the first byte in the source TLV element
     * @param dstBuffer a reference to the destination buffer
     * @param dstOffset the position in the destination buffer
     * @param dstLength the data length to be copied
     *
     * @return <code>dstOffset + dstLength</code>
     *
     * @exception NullPointerException if <code>dstBuffer</code> is <code>null</code>
     * @exception ArrayIndexOutOfBoundsException if findAndCopyValue would cause access of data outside
array bounds.
     * @exception ToolkitException with the following reason codes: <ul>
     *     <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
     *     <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element
     *     <li><code>OUT_OF_TLV_BOUNDARIES</code> if:
     *      <ul>
     *          <li><code>valueOffset</code> parameter is negative or
     *          <li><code>valueOffset + dstLength</code> is greater than the length of the current TLV
     *      </ul> <code>valueOffset</code>, <code>dstLength</code> or both are out of the current TLV
     *     <li><code>BAD_INPUT_PARAMETER</code> if an input parameter is not valid (e.g. occurence =
0)</ul>
     */
    public short findAndCopyValue(byte tag,
                                  byte occurence,
                                  short valueOffset,
                                  byte[] dstBuffer,
                                  short dstOffset,
                                  short dstLength) throws       NullPointerException,
                                                                ArrayIndexOutOfBoundsException,
                                                                ToolkitException {
                        return 0;
```

```
    }

    /**
     * Looks for the first occurence of a TLV element from beginning of a TLV
     * list and compare its value with a buffer.
     * If no TLV element is found, the <code>UNAVAILABLE_ELEMENT</code> exception is thrown.
     * If the method is successful then the corresponding TLV becomes current,
     * else no TLV is selected.
     * This search method is Comprehension Required flag independent.
     *
     * <p>
     * Notes:<ul>
     * <li><em>If </em><code>compareOffset</code><em> parameter is negative or
</em><code>compareOffset</code>
     * <em> is greater than </em><code>compareBuffer.length</code><em>, the length of the
</em><code>compareBuffer</code>
     * <em> array an </em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown and no
find is performed.</em>
     * </ul>
     *
     * @param tag the tag of the TLV element to search
     * @param compareBuffer a reference to the comparison buffer
     * @param compareOffset the position in the comparison buffer
     *
     * @return the result of the comparison as follows: <ul>
     *      <li><code>0</code> if identical
     *      <li><code>-1</code> if the first miscomparing byte in simple TLV is less than that in
<code>compareBuffer</code>,
     *      <li><code>1</code> if the first miscomparing byte in simple TLV is greater than that in
<code>compareBuffer</code>.</ul>
     *
     * @exception NullPointerException if <code>compareBuffer</code> is <code>null</code>
     * @exception ArrayIndexOutOfBoundsException if findAndCompareValue would cause access of data
outside array bounds.
     * @exception ToolkitException with the following reason codes: <ul>
     *      <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
     *      <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element</ul>
     */
    public byte findAndCompareValue(byte tag,
                                    byte[] compareBuffer,
                                    short compareOffset) throws   NullPointerException,
                                                                  ArrayIndexOutOfBoundsException,
                                                                  ToolkitException {
                        return 0;
    }

    /**
     * Looks for the indicated occurence of a TLV element from the beginning of a
     * TLV list and compare its value with a buffer.
     * If no TLV element is found, the <code>UNAVAILABLE_ELEMENT</code> exception is thrown.
     * If the method is successful then the corresponding TLV becomes current,
     * else no TLV is selected.
     * This search method is Comprehension Required flag independent.
     *
     * <p>
     * Notes:<ul>
     * <li><em>If </em><code>compareOffset</code><em> or </em><code>compareLength</code><em>
parameter is negative an </em><code>ArrayIndexOutOfBoundsException</code>
     * <em> exception is thrown and no find and compare is performed.</em>
     * <li><em>If </em><code>compareOffset+compareLength</code><em> is greater than
</em><code>compareBuffer.length</code><em>, the length
```

```
    * of the </em><code>compareBuffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no find and compare is performed.</em>
    * </ul>
    *
    * @param tag the tag of the TLV element to search
    * @param occurrence the occurrence number of the TLV element (1 for the first, 2 for the second...)
    * @param valueOffset the offset of the first byte in the source TLV element
    * @param compareBuffer a reference to the comparison buffer
    * @param compareOffset the position in the comparison buffer
    * @param compareLength the length to be compared
    *
    * @return the result of the comparison as follows: <ul>
    *     <li><code>0</code> if identical
    *     <li><code>-1</code> if the first miscomparing byte in simple TLV is less than that in
<code>compareBuffer</code>,
    *     <li><code>1</code> if the first miscomparing byte in simple TLV is greater than that in
<code>compareBuffer</code>.</ul>
    *
    * @exception NullPointerException if <code>compareBuffer</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if findAndCompareValue would cause access of data
outside array bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *     <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
    *     <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element
    *     <li><code>OUT_OF_TLV_BOUNDARIES</code> if:
    *      <ul>
    *          <li><code>valueOffset</code> parameter is negative or
    *          <li><code>valueOffset + dstLength</code> is greater than the length of the current TLV
    *      </ul>
~~<code>valueOffset</code>, <code>compareLength</code> or both are out of the current TLV~~
    *     <li><code>BAD_INPUT_PARAMETER</code> if an input parameter is not valid (e.g. occurence =
0)</ul>
    */
   public byte findAndCompareValue(byte tag,
                                   byte occurence,
                                   short valueOffset,
                                   byte[] compareBuffer,
                                   short compareOffset,
                                   short compareLength) throws NullPointerException,
                                                               ArrayIndexOutOfBoundsException,
                                                               ToolkitException {
                       return 0;
   }
```

3GPP T3 Meeting #21
Kyoto, Japan, 5 - 7 November, 2001

Tdoc T3-010792

CR-Form-v3

CHANGE REQUEST

⌘ **43.019** **CR** 009 ⌘ rev **-** ⌘ Current version: 8.2.0 ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Clarification of ToolkitException.OUT_OF_TLV_BOUNDARIES in ViewHandler.java
Source:	⌘	T3
Work item code:	⌘	SIM Toolkit API
Date:	⌘	07/11/2001
Category:	⌘	**F**
Release:	⌘	REL-5

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change: ⌘ Currently the ToolkitException OUT_OF_TLV_BOUNDARIES is thrown if a TLV object with length 0 is accessed.
Clarify the enumeration of the different cases for the ToolkitException OUT_OF_TLV_BOUNDARIES.

Summary of change: ⌘ - Clarify rule when ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown
- Allow access to a zero length substring at the end of a TLV element

The current specification is unclear when exactly to throw a OUT_OF_TLV_BOUNDARIES exception. The term currently used is "Throw OUT_OF_TLV_BOUNDARIES if valueOffset, compareLength or both are out of the current TLV".

This is unclear, especially when it comes to the special case of accessing zero bytes at the end of the TLV element. Is it allowed, for example to copy zero bytes from the end of the element? As an example, with the current spec it is probably not possible to write the following statement, if a secured Envelope with zero length secured data is received.

```
envHandler.copyValue( envHdlr.getSecuredDataOffset(),
                      dest,
                      destOffset,
                      envHdlr.getSecuredDataLength() );
```

The Applet has to test for that special case beforehand.
We propose to make the condition when to throw an exception compliant to the standard array access rules used in javacard.framework.

		See Util.arrayCopy(...) as an example
Consequences if not approved:	⌘	In each implementation the case “Length of TLV object = 0” must be checked and processed separately.

Clauses affected:	⌘	Annex A -> ViewHandler.java:
		Methods affected:
		`public short copyValue(short valueOffset, byte destBuffer[], short destOffset, short destLength ) throws [..] ToolkitException`
		
		`public byte compareValue(short valueOffset, byte compareBuffer[], short compareOffset, short compareLength ) throws [..] ToolkitException`
		
		`public short findAndCopyValue(byte tag, byte occurence, short valueOffset, byte destBuffer[], short destOffset, short destLength) throws [..] ToolkitException`
		
		`public byte findAndCompareValue(byte tag, byte occurence, short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength) throws [..] ToolkitException`

Other specs Affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

```
/**
    * Copies a part of the last TLV element which has been found, into a
    * destination buffer.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>dstOffset</code><em> or </em><code>dstLength</code><em> parameter is
negative an </em><code>ArrayIndexOutOfBoundsException</code>
    * <em> exception is thrown and no copy is performed.</em>
    * <li><em>If </em><code>dstOffset+dstLength</code><em> is greater than
</em><code>dstBuffer.length</code><em>, the length
    * of the </em><code>dstBuffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no copy is performed.</em>
    * </ul>
    *
    * @param valueOffset the offset of the first byte in the source TLV element
    * @param dstBuffer a reference to the destination buffer
    * @param dstOffset the position in the destination buffer
    * @param dstLength the data length to be copied
    *
    * @return <code>dstOffset+dstLength</code>
    *
    * @exception NullPointerException if <code>dstBuffer</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if copyValue would cause access of data outside array
bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
    *      <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element
    *      <li><code>OUT_OF_TLV_BOUNDARIES</code> if: ~~<code>valueOffset</code>,
<code>dstLength</code> or both are out of the current TLV </ul>~~
    *      <ul>
    *          <li><code>valueOffset</code> parameter is negative or
    *          <li><code>valueOffset + dstLength</code> is greater than the length of the current TLV
    *      </ul>
    *   </ul>
    */
   public short copyValue( short valueOffset,
                           byte[] dstBuffer,
                           short dstOffset,
                           short dstLength) throws NullPointerException,
                                                   ArrayIndexOutOfBoundsException,
                                                   ToolkitException {
                           return 0;
   }

/**
    * Compares the last found TLV element with a buffer.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>compareOffset</code><em> or </em><code>compareLength</code><em>
parameter is negative an </em><code>ArrayIndexOutOfBoundsException</code>
    * <em> exception is thrown and no compare is performed.</em>
    * <li><em>If </em><code>compareOffset+compareLength</code><em>is greater than
</em><code>compareBuffer.length</code><em>, the length
    * of the </em><code>compareBuffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no compare is performed.</em>
    * </ul>
    *
    * @param valueOffset the offset of the first byte to compare in the TLV element
```

```
    * @param compareBuffer a reference to the comparison buffer
    * @param compareOffset the position in the comparison buffer
    * @param compareLength the length to be compared
    *
    * @return the result of the comparison as follows: <ul>
    *      <li><code>0</code> if identical
    *      <li><code>-1</code> if the first miscomparing byte in simple TLV List is less than that in
<code>compareBuffer</code>,
    *      <li><code>1</code> if the first miscomparing byte in simple TLV List is greater than that in
<code>compareBuffer</code>.</ul>
    *
    * @exception NullPointerException if <code>compareBuffer</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if compareValue would cause access of data outside array
bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
    *      <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element
    *      <li><code>OUT_OF_TLV_BOUNDARIES</code> if:
    *       <ul>
    *           <li><code>valueOffset</code> parameter is negative or
    *           <li><code>valueOffset + dstLength</code> is greater than the length of the current TLV
    *       </ul>
    *   </ul> <code>valueOffset</code>, <code>compareLength</code> or both are out of the current TLV
</ul>
    */
   public byte compareValue(short valueOffset,
                            byte[] compareBuffer,
                            short compareOffset,
                            short compareLength) throws     NullPointerException,
                                                            ArrayIndexOutOfBoundsException,
                                                            ToolkitException {
                        return 0;
   }

   /**
    * Looks for the first occurence of a TLV element from the beginning of a TLV
    * list and copy its value into a destination buffer.
    * If no TLV element is found, the <code>UNAVAILABLE_ELEMENT</code> exception is thrown.
    * If the method is successful then the corresponding TLV becomes current,
    * else no TLV is selected.
    * This search method is Comprehension Required flag independent.
    *
    * <p>
    * Notes:<ul>
    * <li><em>If </em><code>dstOffset</code><em> parameter is negative or </em><code>dstOffset</code>
    * <em> is greater than </em><code>dstBuffer.length</code><em>, the length of the
</em><code>dstBuffer</code>
    * <em> array an </em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown and no
find is performed.</em>
    * </ul>
    *
    * @param tag the tag of the TLV element to search
    * @param dstBuffer a reference to the destination buffer
    * @param dstOffset the position in the destination buffer
    *
    * @return <code>dstOffset</code> + length of the copied value
    *
    * @exception NullPointerException if <code>dstBuffer</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if findAndCopyValue would cause access of data outside
array bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *      <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
```

```
     *     <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element</ul>
     */
    public short findAndCopyValue(byte tag,
                                  byte[] dstBuffer,
                                  short dstOffset) throws       NullPointerException,
                                                                ArrayIndexOutOfBoundsException,
                                                                ToolkitException {
                        return 0;
    }

    /**
     * Looks for the indicated occurence of a TLV element from the beginning of a TLV
     * list and copy its value into a destination buffer.
     * If no TLV element is found, the <code>UNAVAILABLE_ELEMENT</code> exception is thrown.
     * If the method is successful then the corresponding TLV becomes current,
     * else no TLV is selected.
     * This search method is Comprehension Required flag independent.
     *
     * <p>
     * Notes:<ul>
     * <li><em>If </em><code>dstOffset</code><em> or </em><code>dstLength</code><em> parameter is
negative an </em><code>ArrayIndexOutOfBoundsException</code>
     * <em> exception is thrown and no copy is performed.</em>
     * <li><em>If </em><code>dstOffset+dstLength</code><em>is greater than
</em><code>dstBuffer.length</code><em>, the length
     * of the </em><code>dstBuffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
     * and no copy is performed.</em>
     * </ul>
     *
     * @param tag the tag of the TLV element to search
     * @param occurrence the occurrence number of the TLV element (1 for the first, 2 for the second...)
     * @param valueOffset the offset of the first byte in the source TLV element
     * @param dstBuffer a reference to the destination buffer
     * @param dstOffset the position in the destination buffer
     * @param dstLength the data length to be copied
     *
     * @return <code>dstOffset + dstLength</code>
     *
     * @exception NullPointerException if <code>dstBuffer</code> is <code>null</code>
     * @exception ArrayIndexOutOfBoundsException if findAndCopyValue would cause access of data outside
array bounds.
     * @exception ToolkitException with the following reason codes: <ul>
     *     <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
     *     <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element
     *     <li><code>OUT_OF_TLV_BOUNDARIES</code> if:
     *      <ul>
     *          <li><code>valueOffset</code> parameter is negative or
     *          <li><code>valueOffset + dstLength</code> is greater than the length of the current TLV
     *      </ul> <code>valueOffset</code>, <code>dstLength</code> or both are out of the current TLV
     *     <li><code>BAD_INPUT_PARAMETER</code> if an input parameter is not valid (e.g. occurence =
0)</ul>
     */
    public short findAndCopyValue(byte tag,
                                  byte occurence,
                                  short valueOffset,
                                  byte[] dstBuffer,
                                  short dstOffset,
                                  short dstLength) throws       NullPointerException,
                                                                ArrayIndexOutOfBoundsException,
                                                                ToolkitException {
                        return 0;
```

```
    }

    /**
     * Looks for the first occurence of a TLV element from beginning of a TLV
     * list and compare its value with a buffer.
     * If no TLV element is found, the <code>UNAVAILABLE_ELEMENT</code> exception is thrown.
     * If the method is successful then the corresponding TLV becomes current,
     * else no TLV is selected.
     * This search method is Comprehension Required flag independent.
     *
     * <p>
     * Notes:<ul>
     * <li><em>If </em><code>compareOffset</code><em> parameter is negative or
</em><code>compareOffset</code>
     * <em> is greater than </em><code>compareBuffer.length</code><em>, the length of the
</em><code>compareBuffer</code>
     * <em> array an </em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown and no
find is performed.</em>
     * </ul>
     *
     * @param tag the tag of the TLV element to search
     * @param compareBuffer a reference to the comparison buffer
     * @param compareOffset the position in the comparison buffer
     *
     * @return the result of the comparison as follows: <ul>
     *      <li><code>0</code> if identical
     *      <li><code>-1</code> if the first miscomparing byte in simple TLV is less than that in
<code>compareBuffer</code>,
     *      <li><code>1</code> if the first miscomparing byte in simple TLV is greater than that in
<code>compareBuffer</code>.</ul>
     *
     * @exception NullPointerException if <code>compareBuffer</code> is <code>null</code>
     * @exception ArrayIndexOutOfBoundsException if findAndCompareValue would cause access of data
outside array bounds.
     * @exception ToolkitException with the following reason codes: <ul>
     *      <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
     *      <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element</ul>
     */
    public byte findAndCompareValue(byte tag,
                                    byte[] compareBuffer,
                                    short compareOffset) throws   NullPointerException,
                                                                  ArrayIndexOutOfBoundsException,
                                                                  ToolkitException {
                        return 0;
    }

    /**
     * Looks for the indicated occurence of a TLV element from the beginning of a
     * TLV list and compare its value with a buffer.
     * If no TLV element is found, the <code>UNAVAILABLE_ELEMENT</code> exception is thrown.
     * If the method is successful then the corresponding TLV becomes current,
     * else no TLV is selected.
     * This search method is Comprehension Required flag independent.
     *
     * <p>
     * Notes:<ul>
     * <li><em>If </em><code>compareOffset</code><em> or </em><code>compareLength</code><em>
parameter is negative an </em><code>ArrayIndexOutOfBoundsException</code>
     * <em> exception is thrown and no find and compare is performed.</em>
     * <li><em>If </em><code>compareOffset+compareLength</code><em> is greater than
</em><code>compareBuffer.length</code><em>, the length
```

```
    * of the </em><code>compareBuffer</code><em> array an
</em><code>ArrayIndexOutOfBoundsException</code><em> exception is thrown
    * and no find and compare is performed.</em>
    * </ul>
    *
    * @param tag the tag of the TLV element to search
    * @param occurrence the occurrence number of the TLV element (1 for the first, 2 for the second...)
    * @param valueOffset the offset of the first byte in the source TLV element
    * @param compareBuffer a reference to the comparison buffer
    * @param compareOffset the position in the comparison buffer
    * @param compareLength the length to be compared
    *
    * @return the result of the comparison as follows: <ul>
    *     <li><code>0</code> if identical
    *     <li><code>-1</code> if the first miscomparing byte in simple TLV is less than that in
<code>compareBuffer</code>,
    *     <li><code>1</code> if the first miscomparing byte in simple TLV is greater than that in
<code>compareBuffer</code>.</ul>
    *
    * @exception NullPointerException if <code>compareBuffer</code> is <code>null</code>
    * @exception ArrayIndexOutOfBoundsException if findAndCompareValue would cause access of data
outside array bounds.
    * @exception ToolkitException with the following reason codes: <ul>
    *     <li><code>HANDLER_NOT_AVAILABLE</code> if the handler is busy
    *     <li><code>UNAVAILABLE_ELEMENT</code> in case of unavailable TLV element
    *     <li><code>OUT_OF_TLV_BOUNDARIES</code> if:
    *      <ul>
    *          <li><code>valueOffset</code> parameter is negative or
    *          <li><code>valueOffset + dstLength</code> is greater than the length of the current TLV
    *      </ul>
~~<code>valueOffset</code>, <code>compareLength</code> or both are out of the current TLV~~
    *     <li><code>BAD_INPUT_PARAMETER</code> if an input parameter is not valid (e.g. occurence =
0)</ul>
    */
   public byte findAndCompareValue(byte tag,
                                   byte occurence,
                                   short valueOffset,
                                   byte[] compareBuffer,
                                   short compareOffset,
                                   short compareLength) throws NullPointerException,
                                                               ArrayIndexOutOfBoundsException,
                                                               ToolkitException {
                       return 0;
   }
```

3GPP T3 (USIM) Meeting #21
Kyoto, Japan, 5-7 November, 2001

Tdoc T3-010771
(revised from Tdoc T3-010646)

CR-Form-v3

CHANGE REQUEST

⌘ **11.13** **CR** **003** ⌘ rev **-** ⌘ Current version: **7.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Specification for framework part
Source:	⌘	T3
Work item code:	⌘	***Date:*** ⌘ 05/11/01
Category:	⌘	**F** ***Release:*** ⌘ R98

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change: ⌘ Add missing text

Summary of change: ⌘
- Framework Test : Addition of conformance requirements, test procedure and test coverage for framework part wich is divided in the points listed below.
  - Minimum Handler Availaility
  - Handler Integrity
  - Applet Triggering
  - Proactive Command Sending
  - Exception Handling
  - Framework Security Management
  - Envelope Response Posting
  - Toolkit Installation
  - File System Context
  - Other parts transferred to framework from API
- Addition of AID numbering and acronyms for Framework tests
- Addition of the parameter file within Test Area Files

Consequences if not approved: ⌘ Potential interoperability problems

Clauses affected: ⌘ 4.3.1, 4.6, 6.3, Annex F and Annex G

Other specs Affected: ⌘
[ ] Other core specifications ⌘
[ ] Test specifications
[ ] O&M Specifications

Other comments: ⌘

4.3.1 Test Area Reference

Each test area is referenced as follows:
API Testing:: 'API_[package name]_[classname]_[methodname]' where
package name:

sim.access package: '1'

sim.toolkit package: '2'

class name:

yyy: 3 letters for each class.

See Annex A for full classes acronyms list.

method name:

zzzz[input parameters]:

See Annex A for full methods name acronyms list.

FWK: framework testing

Chapter name:

xxx: 3 letters for each chapter

See annex F for full chapter acronyms list

Subchapter name

yyyy: : 4 letters for each subchapter

See annex F for full subchapter acronyms list

~~[TBD]~~

LDR: loader testing

[TBD]

4.3.1.1 Conformance requirements

The conformance requirements are expressed in the following way:

- Method prototype as listed in GSM 03.19 [7]specification.
- Normal execution:
  - Contains normal execution and correct parameters limit values, each referenced as a Conformance Requirement Reference Normal (CRRN)
- Parameters error:
  - Contains parameter errors and incorrect parameter limit values, each referenced as a Conformance Requirement Reference Parameter Error (CRRP)
- Context error:
  - Contains errors due to the context the method is used in, each referenced as a Conformance Requirement Reference Context Error (CRRC)

4.3.1.2 Test Area files

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

- Test Script: [Test Area Reference]_[Test script number].scr
- Test Applet: [Test Area Reference]_[Test applet number].java
- Load Script: [Test Area Reference]_[Load Script number].ldr
- Cleanup Script: [Test Area Reference]_[Cleanup Script number].clr
- Parameter File: [Test Area Reference]_[Parameter File number].par

The test script, applet, installation parameters, load script, cleanup script and conversion parameters numbers start from '1'.
The test script, load script and cleanup script shall share a common syntax and format (see Annex B).
The parameter file has an own syntax (see Annex G) and contains parameters to be used for CAP-file conversion and loading/cleanup script generation.

Scripts file shall be run in the following order:

[Test Area Reference]_1.ldr

[Test Area Reference]_1.scr

[Test Area Reference]_1.clr

[Test Area Reference]_2.ldr

[Test Area Reference]_2.scr

[Test Area Reference]_2.clr

….

[Test Area Reference]_n.ldr

[Test Area Reference]_n.scr

[Test Area Reference]_n.clr

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

4.3.1.3 Test Procedure

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

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

4.3.1.4 Test Coverage

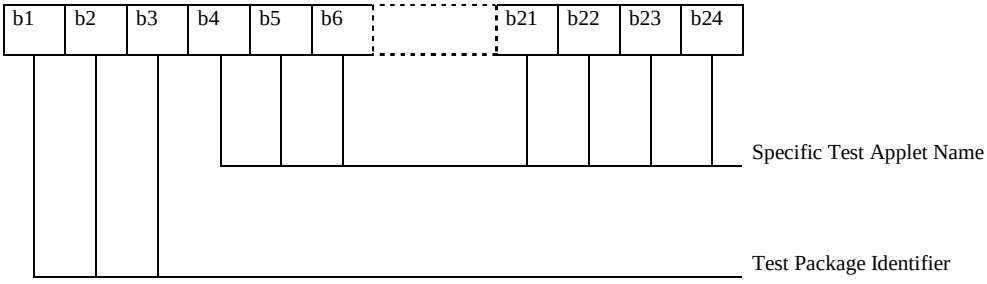
The table at the end of each test procedure indicates the correspondence between the Conformance Requirements Reference (CRR) and the different test cases.

…

4.6 AID Coding

The AID coding for the ~~API~~ Test Packages, Applet classes and Applet shall be as specified in TS 101.220 [14]. In addition, the following TAR values are defined for use within the present document:

TAR Coding (3 bytes/ 24 bits):



Test package Identifier(bits b1-b3):

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

001 API

010 Framework

011 Loader

111 sim.test.util

other values are RFU

Application Provider specific data (1 byte):

'00' for Package

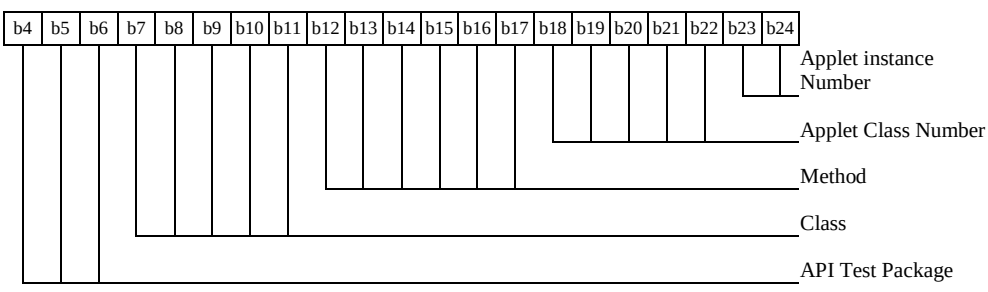
'01' for Applet class

'02' for Applet Instance

For example, the AID of Package sim.test.util is 'A0 00 00 00 09 00 02 FF FF FF FF 89 E0 00 00 00'

4.6.1 Specific Test Applet Name for API

Specific applet test name (bits b4-b24):



for API ~~test~~ Test ~~package~~Package(3 bits)

001 sim.access

010 sim.toolkit

other are RFU

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

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

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

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

~~Application Provider specific data (1 byte):~~

~~'00' for Package~~

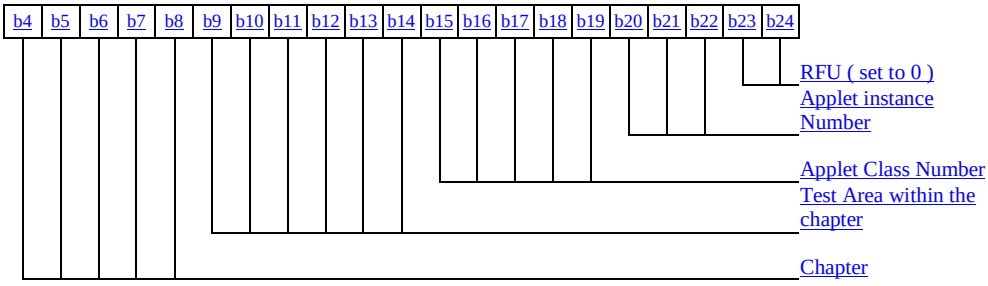
~~'01' for Applet class~~

~~'02' for Applet Instance~~

~~For example, the AID of Package sim.test.util is 'A0 00 00 00 09 00 02 FF FF FF FF 89 E0 00 00 00'~~

4.6.1 Specific Test Applet Name for Framework

Specific applet test name (bits b4-b24):



for Chapter (5 bits)

00001 Toolkit Installation Parameters

00010 Minimum Handler Availability

00011 Handler Integrity

00100 Applet Triggering

00101 Proactive Command Sending

00110 Envelope Response Posting

00111 Framework Security

01000 File System Context

01001 Exception Handling

01010 Other parts transferred to framework from API

other are RFU

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

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

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

...

6.3 SIM Toolkit Framework

6.3.1 Minimum Handler Availability

This test area tests the rules that define the minimum requirements for the availability of the system handlers.

6.3.1.1 ProactiveHandler

Test Area Reference: FWK_MHA_PAHD

6.3.1.1.1 Conformance Requirement

Normal Execution

CRRN1: If a proactive session is not ongoing the ProactiveHandler is available from the invocation to the termination of the processToolkit method for the following events:

EVENT_FORMATTED_SMS_PP_ENV

EVENT_UNFORMATTED_SMS_PP_ENV

EVENT_UNFORMATTED_SMS_CB

EVENT_MENU_SELECTION

EVENT_MENU_SELECTION_HELP_REQUEST

EVENT_TIMER_EXPIRATION

EVENT_EVENT_DOWNLOAD_MT_CALL

EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE

EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

EVENT_UNRECOGNIZED_ENVELOPE

EVENT_STATUS_COMMAND

EVENT_FORMATTED_SMS_PP_UPD

EVENT_UNFORMATTED_SMS_PP_UPD

EVENT_CALL_CONTROL

EVENT_SMS_MO_CONTROL

EVENT_PROFILE_DOWNLOAD

6.3.1.1.2 Test Suite Files

Test Script: FWK_MHA_PAHD_1.scr

Test Applet: FWK_MHA_PAHD_1.java

FWK_MHA_PAHD_2.java

Load Script: FWK_MHA_PAHD_1.ldr

Cleanup Script: FWK_MHA_PAHD_1.clr

Parameter File: FWK_MHA_PAHD_1.par

6.3.1.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.1.1.4 Test Coverage

CRR Number	Test Case Number

6.3.1.2 ProactiveResponseHandler

Test Area Reference: FWK_MHA_PRHD

6.3.1.2.1 Conformance Requirement

Normal Execution

CRRN1: The ProactiveResponseHandler is available after the first call to the ProactiveHandler.send method to the termination of the processToolkit method for the following events:

EVENT_FORMATTED_SMS_PP_ENV

EVENT_UNFORMATTED_SMS_PP_ENV

EVENT_UNFORMATTED_SMS_CB

EVENT_MENU_SELECTION

EVENT_MENU_SELECTION_HELP_REQUEST

EVENT_TIMER_EXPIRATION

EVENT_EVENT_DOWNLOAD_MT_CALL

EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE

EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

EVENT_UNRECOGNIZED_ENVELOPE

EVENT_STATUS_COMMAND

EVENT_FORMATTED_SMS_PP_UPD

EVENT_UNFORMATTED_SMS_PP_UPD

EVENT_CALL_CONTROL

EVENT_SMS_MO_CONTROL

EVENT_PROFILE_DOWNLOAD

6.3.1.2.2 Test Suite Files

Test Script: FWK_MHA_PRHD_1.scr

Test Applet: FWK_MHA_PRHD_1.java

FWK_MHA_PRHD_2.java

Load Script: FWK_MHA_PRHD_1.ldr

Cleanup Script: FWK_MHA_PRHD_1.clr

Parameter File: FWK_MHA_PRHD_1.par

6.3.1.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.1.2.4 Test Coverage

CRR Number	Test Case Number

6.3.1.3 EnvelopeHandler

Test Area Reference: FWK_MHA_ENHD

6.3.1.3.1 Conformance Requirement

Normal Execution

CRRN1: The EnvelopeHandler and its content are available for all toolkit applets triggered from the invocation to the termination of their processToolkit method for the following events:.

EVENT_FORMATTED_SMS_PP_ENV

EVENT_UNFORMATTED_SMS_PP_ENV

EVENT_UNFORMATTED_SMS_CB

EVENT_MENU_SELECTION

EVENT_MENU_SELECTION_HELP_REQUEST

EVENT_TIMER_EXPIRATION

EVENT_EVENT_DOWNLOAD_MT_CALL

EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE

EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

EVENT_UNRECOGNIZED_ENVELOPE

EVENT_FORMATTED_SMS_PP_UPD

EVENT_UNFORMATTED_SMS_PP_UPD

EVENT_CALL_CONTROL

EVENT_SMS_MO_CONTROL

Context Errors

CRRC1: The EnvelopeHandler and its content are not available for any toolkit applet triggered from the invocation to the termination of their processToolkit method for the following events:

EVENT_STATUS_COMMAND

EVENT_PROFILE_DOWNLOAD

6.3.1.3.2 Test Suite Files

Test Script: FWK_MHA_ENHD_1.scr

Test Applet: FWK_MHA_ENHD_1.java

FWK_MHA_ENHD_2.java

Load Script: FWK_MHA_ENHD_1.ldr

Cleanup Script: FWK_MHA_ENHD_1.clr

Parameter File: FWK_MHA_ENHD_1.par

6.3.1.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.1.3.4 Test Coverage

CRR Number	Test Case Number

6.3.1.4 EnvelopeResponseHandler

Test Area Reference: FWK_MHA_ERHD

6.3.1.4.1 Conformance Requirement

Normal Execution

CRRN1: The handler is available for all triggered toolkit applets from the invocation of the processToolkit method of the toolkit applet until a toolkit applet has posted an envelope response or the first invocation of the ProactiveHandler.send method for the following events:.

EVENT_FORMATTED_SMS_PP_ENV

EVENT_UNFORMATTED_SMS_PP_ENV

EVENT_CALL_CONTROL

EVENT_SMS_MO_CONTROL

EVENT_UNRECOGNIZED_ENVELOPE

CRRN2: After a call to the post method the handler is not longer available

Context Errors

CRRC1: The handler is not available for the following events:

EVENT_UNFORMATTED_SMS_CB

EVENT_MENU_SELECTION

EVENT_MENU_SELECTION_HELP_REQUEST

EVENT_TIMER_EXPIRATION

EVENT_EVENT_DOWNLOAD_MT_CALL

EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE

EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

EVENT_STATUS_COMMAND

EVENT_FORMATTED_SMS_PP_UPD

EVENT_UNFORMATTED_SMS_PP_UPD

EVENT_PROFILE_DOWNLOAD

6.3.1.4.2 Test Suite Files

Test Script: FWK_MHA_ERHD_1.scr

Test Applet: FWK_MHA_ERHD_1.java

FWK_MHA_ERHD_2.java

Load Script: FWK_MHA_ERHD_1.ldr

Cleanup Script: FWK_MHA_ERHD_1.clr

Parameter File: FWK_MHA_ERHD_1.par

6.3.1.4.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.1.4.4 Test Coverage

CRR Number	Test Case Number

6.3.2 Handler Integrity

6.3.2.1 ProactiveHandler

Test Area Reference: FWK_HIN_PAHD

6.3.2.1.1 Conformance Requirement

Normal Execution

CRRN1: At the processToolkit invocation the TLV-List is cleared.

CRRN2: After a call to ProactiveHandler.send method the handler will remain unchanged until the ProactiveHandler.init or appendTLV method are called.

6.3.2.1.2 Test Suite Files:

Test Script: FWK_HIN_PAHD_1.scr

Test Applet: FWK_HIN_PAHD_1.java

FWK_HIN_PAHD_2.java

Load Script: FWK_HIN_PAHD_1.ldr

Cleanup Script: FWK_HIN_PAHD_1.clr

Parameter File: FWK_HIN_PAHD_1.par

6.3.2.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.2.1.4 Test Coverage

CRR Number	Test Case Number

6.3.2.2 ProactiveResponseHandler

Test Area Reference: FWK_HIN_PRHD

6.3.2.2.1 Conformance Requirement

Normal Execution

CRRN1: The ProactiveResponseHandler content is changed after the call to ProactiveHandler.send method and remains unchanged until next call to the ProactiveHandler.send method.

CRRN2: The ProactiveResponseHandler may not be available before the first call to ProactiveHandler.send method, if available the content is cleared.

6.3.2.2.2 Test Suite Files

Test Script: FWK_HIN_PRHD_1.scr

Test Applet: FWK_HIN_PRHD_1.java

Load Script: FWK_HIN_PRHD_1.ldr

Cleanup Script: FWK_HIN_PRHD_1.clr

Parameter File: FWK_HIN_PRHD_1.par

6.3.2.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.2.2.4 Test Coverage

CRR Number	Test Case Number

6.3.2.3 EnvelopeHandler

Test Area Reference: FWK_HIN_ENHD

6.3.2.3.1 Conformance Requirement

Normal Execution

CRRN1: The EnvelopeHandler content shall have the same value during the processToolkit

CRRN2: The SIM Toolkit Framework guarantees that all triggered toolkit applets receive the data.

6.3.2.3.2 Test Suite Files

Test Script: FWK_HIN_ENHD_1.scr

Test Applet: FWK_HIN_ENHD_1.java

Load Script: FWK_HIN_ENHD_1.ldr

Cleanup Script: FWK_HIN_ENHD_1.clr

Parameter File: FWK_HIN_ENHD_1.par

6.3.2.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.2.3.4 Test Coverage

CRR Number	Test Case Number

6.3.3 Applet Triggering

6.3.3.1 EVENT_PROFILE_DOWNLOAD

Test Area Reference: FWK_APT_EPDW

6.3.3.1.1 Conformance Requirement

Normal Execution

CRRN1: Upon the reception of Terminal Profile command by the SIM, the STF stores the ME Profile and then triggers the registered toolkit applets.

CRRN2: The applet is not triggered by the EVENT_PROFILE_DOWNLOAD once it has deregistered from this event.

CRRN3: The STF shall not reply busy to a Terminal Profile command

6.3.3.1.2 Test Suite Files

Test Script: FWK_APT_EPDW_1.scr

Test Applet: FWK_APT_EPDW_1.java

FWK_APT_EPDW_2.java

FWK_APT_EPDW_3.java

Load Script: FWK_APT_EPDW_1.ldr

Cleanup Script: FWK_APT_EPDW_1.clr

Parameter File: FWK_APT_EPDW_1.par

6.3.3.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.1.4 Test Coverage

CRR Number	Test Case Number

6.3.3.2 EVENT_MENU_SELECTION

Test Area Reference: FWK_APT_EMSE

6.3.3.2.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_MENU_SELECTION when an Envelope Menu Selection is received with the item identifier of a menu entry of this applet if no proactive session is ongoing.

6.3.3.2.2 Test Suite Files

Test Script: FWK_APT_EMSE_1.scr

Test Applet: FWK_APT_EMSE_1.java

Load Script: FWK_APT_EMSE_1.ldr

Cleanup Script: FWK_APT_EMSE_1.clr

Parameter File: FWK_APT_EMSE_1.par

6.3.3.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.2.4 Test Coverage

CRR Number	Test Case Number

6.3.3.3 EVENT_MENU_SELECTION_HELP_REQUEST

Test Area Reference: FWK_APT_EMSH

6.3.3.3.1 Conformance Requirement

Normal Execution

CRRN1: If and ENVELOPE (MENU_SELECTION_HELP_SUPPORTED) command is received for one entry supporting help, then STF shall trigger the corresponding applet.

6.3.3.3.2 Test Suite Files

none

6.3.3.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.3.4 Test Coverage

CRR Number	Test Case Number

6.3.3.4 EVENT_FORMATTED_SMS_PP_ENV

Test Area Reference: FWK_APT_EFSE

6.3.3.4.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_FORMATTED_SMS_PP_ENV once it has registered to this event and Formatted Envelope DataDownLoad with the corresponding TAR, defined at the applet loading, is received and no proactive session is ongoing

CRRN2: The applet is not triggered by the EVENT_FORMATTED_SMS_PP_ENV once it has deregistered from this event.

6.3.3.4.2 Test Suite Files

Test Script: FWK_APT_EFSE_1.scr

Test Applet: FWK_APT_EFSE_1.java

Load Script: FWK_APT_EFSE_1.ldr

Cleanup Script: FWK_APT_EFSE_1.clr

Parameter File: FWK_APT_EFSE_1.par

6.3.3.4.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.4.4 Test Coverage

CRR Number	Test Case Number

6.3.3.5 EVENT_UNFORMATTED_SMS_PP_ENV

Test Area Reference: FWK_APT_EUSE

6.3.3.5.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_ UNFORMATTED_SMS_PP_ENV once it has registered to this event and an Unformatted Envelope DataDownLoad is received if no proactive session is ongoing

CRRN2: The applet is not triggered by the EVENT_ UNFORMATTED_SMS_PP_ENV once it has deregistered from this event.

6.3.3.5.2 Test Suite Files

Test Script: FWK_APT_EUSE_1.scr

Test Applet: FWK_APT_EUSE_1.java

Load Script: FWK_APT_EUSE_1.ldr

Cleanup Script: FWK_APT_EUSE_1.clr

Parameter File: FWK_APT_EUSE_1.par

6.3.3.5.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.5.4 Test Coverage

CRR Number	Test Case Number

6.3.3.6 EVENT_CALL_CONTROL_BY_SIM

Test Area Reference: FWK_APT_ECCN

6.3.3.6.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_CALL_CONTROL_BY_SIM once it has registered to this event and an Envelope Call Control is received

CRRN2: The applet is not triggered by the EVENT_CALL_CONTROL_BY_SIM once it has deregistered from this event.

6.3.3.6.2 Test Suite Files

Test Script: FWK_APT_ECCN_1.scr

Test Applet: FWK_APT_ECCN_1.java

Load Script: FWK_APT_ECCN_1.ldr

Cleanup Script: FWK_APT_ECCN_1.clr

Parameter File: FWK_APT_ECCN_1.par

6.3.3.6.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.6.4 Test Coverage

CRR Number	Test Case Number

6.3.3.7 EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM

Test Area Reference: FWK_APT_EMCN

6.3.3.7.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM once it has registered to this event and an Envelope MO Short Message Control.

CRRN2: The applet is not triggered by the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM once it has deregistered from this event.

6.3.3.7.2 Test Suite Files

Test Script: FWK_APT_EMCN_1.scr

Test Applet: FWK_APT_EMCN_1.java

Load Script: FWK_APT_EMCN_1.ldr

Cleanup Script: FWK_APT_EMCN_1.clr

Parameter File: FWK_APT_EMCN_1.par

6.3.3.7.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.7.4 Test Coverage

CRR Number	Test Case Number

6.3.3.8 EVENT_TIMER_EXPIRATION

Test Area Reference: FWK_APT_ETEX

6.3.3.8.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_TIMER_EXPIRATION once it has been registered to this event and an Envelope Timer Expiration with a Timer Identifier of the applet is received if no proactive session is ongoing.

CRRN2: The applet is not triggered by the EVENT_TIMER_EXPIRATION once it has been deregistered from this event.

6.3.3.8.2 Test Suite Files

Test Script: FWK_APT_ETEX_1.scr

Test Applet: FWK_APT_ETEX_1.java

Load Script: FWK_APT_ETEX_1.ldr

Cleanup Script: FWK_APT_ETEX_1.clr

Parameter File: FWK_APT_ETEX_1.par

6.3.3.8.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.8.4 Test Coverage

CRR Number	Test Case Number

6.3.3.9 EVENT_UNFORMATTED_SMS_CB

Test Area Reference: FWK_APT_EUCB

6.3.3.9.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_UNFORMATTED_SMS_CB once it has registered to this event and an Envelope Cell Broadcast DownLoad is received.

CRRN2: The applet is not triggered by the EVENT_UNFORMATTED_SMS_CB once it has deregistered from this event.

6.3.3.9.2 Test Suite Files

Test Script: FWK_APT_EUCB_1.scr

Test Applet: FWK_APT_EUCB_1.java

Load Script: FWK_APT_EUCB_1.ldr

Cleanup Script: FWK_APT_EUCB_1.clr

Parameter File: FWK_APT_EUCB_1.par

6.3.3.9.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.9.4 Test Coverage

CRR Number	Test Case Number

6.3.3.10 EVENT_EVENT_DOWNLOAD_MT_CALL

Test Area Reference: FWK_APT_EDMC

6.3.3.10.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_MT_CALL once it has registered to this event and an Envelope Event DownLoad MT Call is received.

CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_MT_CALL once it has deregistered from this event.

6.3.3.10.2 Test Suite Files

Test Script: FWK_APT_EMSE_1.scr

Test Applet: FWK_APT_EMSE_1.java

Load Script: FWK_APT_EMSE_1.ldr

Cleanup Script: FWK_APT_EMSE_1.clr

Parameter File: FWK_APT_EMSE_1.par

6.3.3.10.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.10.4 Test Coverage

CRR Number	Test Case Number

6.3.3.11 EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

Test Area Reference: FWK_APT_EDCC

6.3.3.11.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_CALL_CONNECTED once it has registered to this event and an Envelope Event DownLoad Call Connected is received.

CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_CALL_CONNECTED once it has deregistered from this event.

6.3.3.11.2 Test Suite Files

Test Script: FWK_APT_EDCC_1.scr

Test Applet: FWK_APT_EDCC_1.java

Load Script: FWK_APT_EDCC_1.ldr

Clean-up Script: FWK_APT_EDCC_1.clr

6.3.3.11.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.11.4 Test Coverage

CRR Number	Test Case Number

6.3.3.12 EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

Test Area Reference: FWK_APT_EDCD

6.3.3.12.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED once it has registered to this event and an Envelope Event DownLoad Call Disconnected is received.

CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED once it has deregistered from this event.

6.3.3.12.2 Test Suite Files

Test Script: FWK_APT_EDCD_1.scr

Test Applet: FWK_APT_EDCD_1.java

Load Script: FWK_APT_EDCD_1.ldr

Cleanup Script: FWK_APT_EDCD_1.clr

Parameter File: FWK_APT_EDCD_1.par

6.3.3.12.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.12.4 Test Coverage

CRR Number	Test Case Number

6.3.3.13 EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

Test Area Reference: FWK_APT_EDLS

6.3.3.13.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_LOCATION_STATUS once it has registered to this event and an Envelope Event DownLoad Location Status is received.

CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_LOCATION_STATUS once it has deregistered from this event.

6.3.3.13.2 Test Suite Files

Test Script: FWK_APT_EDLS_1.scr

Test Applet: FWK_APT_EDLS_1.java

Load Script: FWK_APT_EDLS_1.ldr

Cleanup Script: FWK_APT_EDLS_1.clr

Parameter File: FWK_APT_EDLS_1.par

6.3.3.13.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.13.4 Test Coverage

CRR Number	Test Case Number

6.3.3.14 EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

Test Area Reference: FWK_APT_EDUA

6.3.3.14.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY once it has registered to this event and an Envelope Event DownLoad User Activity is received.

CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_USER_ACTIVITY once it has deregistered from this event.

6.3.3.14.2 Test Suite Files

Test Script: FWK_APT_EDUA_1.scr

Test Applet: FWK_APT_EDUA_1.java

Load Script: FWK_APT_EDUA_1.ldr

Cleanup Script: FWK_APT_EDUA_1.clr

Parameter File: FWK_APT_EDUA_1.par

6.3.3.14.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.14.4 Test Coverage

CRR Number	Test Case Number

6.3.3.15 EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE

Test Area Reference: FWK_APT_EDIS

6.3.3.15.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE once it has registered to this event and an Envelope Event DownLoad Idle Screen Available is received.

CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE once it has deregistered from this event.

6.3.3.15.2 Test Suite Files

Test Script: FWK_APT_EDIS_1.scr

Test Applet: FWK_APT_EDIS_1.java

Load Script: FWK_APT_EDIS_1.ldr

Cleanup Script: FWK_APT_EDIS_1.clr

Parameter File: FWK_APT_EDIS_1.par

6.3.3.15.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

Id	Description	API Expectation	APDU Expectation

6.3.3.15.4 Test Coverage

CRR Number	Test Case Number

6.3.3.16 EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

Test Area Reference: FWK_APT_EDCR

6.3.3.16.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS once it has registered to this event and Envelope Event DownLoad Card Reader Status is received.

CRRN2: The applet is not triggered by the EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS once it has deregistered from this event.

6.3.3.16.2 Test Suite Files

Test Script: FWK_APT_EDCR_1.scr

Test Applet: FWK_APT_EDCR_1.java

Load Script: FWK_APT_EDCR_1.ldr

Cleanup Script: FWK_APT_EDCR_1.clr

Parameter File: FWK_APT_EDCR_1.par

6.3.3.16.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.16.4 Test Coverage

CRR Number	Test Case Number

6.3.3.17 EVENT_UNRECOGNIZED_ENVELOPE

Test Area Reference: FWK_APT_EUEV

6.3.3.17.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_UNRECOGNIZED_ENVELOPE once it has registered to this event and an Unrecognized Envelope is received.

CRRN2: The applet is not triggered by the EVENT_UNRECOGNIZED_ENVELOPE once it has deregistered from this event.

6.3.3.17.2 Test Suite Files

Test Script: FWK_APT_EUEN_1.scr

Test Applet: FWK_APT_EUEN_1.java

Load Script: FWK_APT_EUEN_1.ldr

Cleanup Script: FWK_APT_EUEN_1.clr

Parameter File: FWK_APT_EUEN_1.par

6.3.3.17.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.17.4 Test Coverage

CRR Number	Test Case Number

6.3.3.18 EVENT_STATUS_COMMAND

Test Area Reference: FWK_APT_ESTC

6.3.3.18.1 Conformance Requirement

Normal Execution

CRRN1: The applet is triggered by the EVENT_STATUS_COMMAND once it has registered to this event and a Status Command is received.

CRRN2: The applet is not triggered by the EVENT_STATUS_COMMAND once it has deregistered from this event.

6.3.3.18.2 Test Suite Files

Test Script: FWK_APT_ESTC_1.scr

Test Applet: FWK_APT_ESTC_1.java

Load Script: FWK_APT_ESTC_1.ldr

Cleanup Script: FWK_APT_ESTC_1.clr

Parameter File: FWK_APT_ESTC_1.par

6.3.3.18.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

6.3.3.18.4 Test Coverage

CR Number	Test Case Number

6.3.4 Proactive Command Sending by the STF

6.3.4.1 System Proactive Commands

Test Area Reference: FWK_PCS_SPCO

6.3.4.1.1 Conformance Requirements

Normal Execution

CRRN1: When a toolkit applet changes a menu entry of its registry object, the SIM Toolkit Framework shall dynamically* update the menu stored in the ME during the current card session

CRRN2: The STF shall use the data of the EFsume file when issuing the SET UP MENU proactive command.

CRRN3: For all EVENT_EVENT_DOWNLOAD_*: When a toolkit applet changes one or more of these requested events of its registry object, the STF shall dynamically* update the event list stored in the ME during the current card session by SET UP EVENT LIST proactive command.

*The STF shall send its system proactive command as soon as no proactive session is pending and all the applets registered to the current events have been triggered and have returned from the processToolkit method invocation.

6.3.4.1.2 Test Suite Files

Test Script: FWK_PCS_SPCO_1.scr

Test Applet: FWK_PCS_SPCO_1.java

Load Script: FWK_PCS_SPCO_1.ldr

Cleanup Script: FWK_PCS_SPCO_1.clr

Parameter File: FWK_PCS_SPCO_1.par

6.3.4.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Install/install Applet** `Constructor: ToolkitRegistry.setEvent`		setEventList proactive command [Event list]= '19020003'
2	**TEST is triggered by ENVELOPE (MENU SELECTION) command** `ToolkitRegistry.clearEvent`		1. DISPLAY TEXT Proactive command 2. SET UP EVENT LIST Proactive command [CommandQualifier]= 00h

6.3.4.1.4 Test Coverage

CRR number	Test case number
N1	see: chapter 6.2.9.2, CRRN1, chapter 6.2.9.4, CRRN3, chapter 6.2.9.5 CRRN4, chapter 6.2.9.8 CRRN1
N2	see: chapter 6.2.9.2 CRRN1, chapter 6.2.9.8 CRRN1
N3	1,2

6.3.4.2 Interaction with GSM commands

Test Area Reference: FWK_PCS_IGCO

6.3.4.2.1 Conformance Requirements

Normal Execution

CRRN1: The STF shall process a GSM command even when a proactive command is pending (before and after the FETCH command until the terminal response). The STF shall answer with the SW1 and SW2 described in [3] and [4].

6.3.4.2.2 Test Suite Files

Test Script: FWK_PCS_IGCO_1.scr

Test Applet: FWK_PCS_IGCO_1.java

Load Script: FWK_PCS_IGCO_1.ldr

Cleanup Script: FWK_PCS_IGCO_1.clr

Parameter File: FWK_PCS_IGCO_1.par

6.3.4.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Interaction with GSM Commands after TERMINAL PROFILE in connection with FETCH and TERMINAL RESPONSE** Applet calls initMenuEntry ATR TERMINAL PROFILE (Profile : supports all facilities except: SET UP EVENT LIST, POLL INTERVAL and POLLING OFF) 1- System issues a proactive command SETUP_MENU 2- SELECT MF 3- GET RESPONSE (6 Bytes) 4- Failed SELECT File 5- FETCH 6- SELECT MF 7- GET RESPONSE (6 Bytes) 8- TERMINAL RESPONSE		1- 91xx 2- 9Fxx 3- 91xx 4- 9404 5- Proactive Command: SETUP MENU 6- 9Fxx 7- 9000 8- 9000
2	**Interaction with GSM Commands after ENVELOPE (MENU SELECTION) in connection with FETCH and TERMINAL RESPONSE** Menu Entry ID = 0x01 1- SELECT MF 2- GET RESPONSE (6 Bytes) 3- Failed SELECT File 4- FETCH 5- SELECT MF 6- GET RESPONSE (6 Bytes) 7- TERMINAL RESPONSE		1- 9FXX 2- 91XX 3- 9404 4- Proactive Command: DISPLAY TEXT 5- 9FXX 6- 9000 7- 9000
3	**Interaction with GSM Commands after TERMINAL RESPONSE in proactive command**		

Id	Description	API Expectation	APDU Expectation
	session in connection with FETCH and TERMINAL RESPONSE Menu Entry ID = 0x02 1- SELECT MF 2- GET RESPONSE (6 Bytes) 3- FETCH 4- SELECT MF 5- GET RESPONSE (6 Bytes) 6- Failed SELECT File 7- TERMINAL RESPONSE 8- SELECT MF 9- GET RESPONSE (6 Bytes) 10-Failed SELECT File 11-FETCH 12-SELECT MF 13-GET RESPONSE (6 Bytes) 14-TERMINAL RESPONSE		1- 9FXX 2- 91XX 3- Proactive Command: DISPLAY TEXT 4- 9FXX 5- 9000 6- 9404 7- 9000 8- 9FXX 9- 91XX 10-9404 11-Proactive Command: DISPLAY TEXT 12-9FXX 13-9000 14-9000

6.3.4.2.4 Test Coverage

CRR number	Test case number
N1	1,2,3

6.3.5 Exception Handling

6.3.5.1 Hide Exceptions from the ME

Test Area Reference: FWK_EXH_HEME

6.3.5.1.1 Conformance Requirements

Normal Execution

CRRN1 : A toolkit applet may throw an exception, but this error will not be sent to the ME.

* Because the behaviour of the SIM is not exactly defined for the above CRRN, there are no tests defined here yet.

6.3.5.2 Interaction with Multiple Triggering

Test Area Reference: FWK_EXH_IMTG

6.3.5.2.1 Conformance Requirements

Normal Execution:

CRRN1 : An exception thrown by a toolkit applet, will not influence toolkit applets registered to the same event

6.3.5.2.2 Test Suite Files

Test Script: FWK_EXH_IMTR_1.scr

Test Applet: : FWK_EXH_IMTR_1.java

Load Script: : FWK_EXH_IMTR_1.ldr

Cleanup Script: FWK_EXH_IMTR_1.clr

Parameter File: FWK_EXH_IMTR_1.par

6.3.5.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

0	**Load/install 2 toolkit applets registered to EVENT_STATUS_COMMAND, EVENT_PROFILE_DOWNLOAD, EVENT_UNRECOGNISED_ENVELOPE, EVENT_EVENT_DOWNLOAD_MT_CALL, EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_CB**
	
	`applet1: Priority= 0x01,`
	`applet2: Priority= 0x02,`
	`(i.e. applet1 is triggered before applet2)`		

1	**STATUS_COMMAND is sent** Applet 1 is triggered Applet 2 is triggered	Applet1: `NullPointerException` is thrown	
2	**PROFILE_DOWNLOAD is sent** Applet 1 is triggered Applet 2 is triggered	Applet1: `NullPointerException` is thrown	
3	**UNRECOGNISED_ENVELOPE is sent** Applet 1 is triggered Applet 2 is triggered	Applet1: `NullPointerException` is thrown	
4	**EVENT_DOWNLOAD_MT_CALL is sent** Applet 1 is triggered Applet 2 is triggered	Applet1: `NullPointerException` is thrown	
5	**UNFORMATTED_SMS_PP_ENV is sent** Applet 1 is triggered Applet 2 is triggered	Applet1: `NullPointerException` is thrown	
6	**UNFORMATTED_SMS_PP_UPD is sent** Applet 1 is triggered Applet 2 is triggered	Applet1: `NullPointerException` is thrown	
7	**UNFORMATTED_SMS_CB is sent** Applet 1 is triggered	Applet1: `NullPointerException` is	

Id	Description	API Expectation	APDU Expectation
	Applet 2 is triggered	thrown	

6.3.5.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1,2,3,4,5,6,7

6.3.6 Framework Security Management

Security Parameters

The table that follows contains the security parameters that shall be used when the 03.48 security is required in the test cases developed in the current section.

Parameter	Value in hexadecimal
KIC	11
KID	11
CNTR	00 00 00 00 01
Key for ciphering	01 41 42 7F DA E8 91 A7
Key for RC/CC/DS	01 23 45 67 89 AB CD EF

If a parameter is not listed explicitly in the above table, the default values of section 4.7.3.1 apply.

6.3.6.1 Input Data

Test Area Reference: FWK_FWS_INDA

6.3.6.1.1 Conformance Requirements

Normal Execution

CRRN1: If the SIM receives an envelope APDU containing an SMS_DATADOWNLOAD BER TLV formatted according to GSM03.48, the SIM Toolkit Framework shall verify the GSM03.48 security of the SMS TPDU.

CRRN2: The toolkit applet will only be triggered if the TAR is known and the security verified.

6.3.6.1.2 Test Area Files

Test Script: FWK_FWS_INDA_1.scr

Test Applet: FWK_FWS_INDA_1.java

FWK_FWS_INDA_2.java

Load Script: FWK_FWS_INDA_1.ldr

Cleanup Script: FWK_FWS_INDA_1.clr

Parameter File: FWK_FWS_INDA_1.par

6.3.6.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Load and install applet 1**		
	Envelope(SMS-PP) 03.48 formatted	The applet is triggered.	

Id	Description	API Expectation	APDU Expectation
	Ciphering; Cryptographic checksum; No proof of receipt; Data = 01		
2	**Load and install applet 2**		
	Envelope(SMS-PP) 03.48 formatted Ciphering; Cryptographic checksum; No proof of receipt; TAR of Applet 1 Data = 02	This Envelope(SMS-PP) triggers Applet 1	The SIM answers to the Envelope with status words 9000
	Envelope(SMS-PP) 03.48 formatted No ciphering; No cryptographic checksum; No proof of receipt; TAR of Applet 2 Data = 03	This Envelope(SMS-PP) triggers Applet 2	The SIM answers to the Envelope with status words 9000
3	**Envelope(SMS-PP) 03.48 formatted** No ciphering; Wrong Cryptographic checksum; No proof of receipt; TAR of Applet 1 Data = 04	No applet is triggered	The SIM answers to the Envelope with status words 9000

6.3.6.1.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1,2,3
CRRN2	3

6.3.6.2 Output Data

Test Area Reference: FWK_FWS_OUDA

6.3.6.2.1 Conformance Requirements

Normal Execution

CRRN1: The SIM Toolkit Framework shall secure and send the response packet.

6.3.6.2.2 Test Area Files

Test Script: FWK_FWS_OUDA_1.scr

Test Applet: FWK_FWS_OUDA_1.java

FWK_FWS_OUDA_2.java

Load Script: FWK_FWS_OUDA_1.ldr

Cleanup Script: FWK_FWS_OUDA_1.clr

Parameter File: FWK_FWS_OUDA_1.par

6.3.6.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Envelope(SMS-PP) 03.48 formatted** Ciphering; Cryptographic checksum; proof of receipt response shall be sent using SMS-Deliver-Report; no security applied to proof of receipt Data in plain text = "APPLET1"	The applet is triggered and sends a "Display Text" proactive command with the data received in the Envelope.	The SIM answers to the Envelope with status words 9Fxx and a PoR is retrieved with a GetResponse command. The PoR has no application data. The SIM answers to the Get Response command with status words 91xx to issue a Display Text "APPLET1".
2	**Envelope(SMS-PP) 03.48 formatted** Ciphering; Cryptographic checksum; proof of receipt response shall be sent using SMS-Deliver-Report; no security applied to proof of receipt	The applet posts application data. It does not call the ProactiveHandler.send() method	The SIM answers to the Envelope with status words 9Fxx and a PoR is retrieved with a GetResponse command. The PoR has the application data posted by the application. The SIM answers to the Get Response command with status words 9000.
3	**Envelope(SMS-PP) 03.48 formatted** Ciphering; Cryptographic checksum; proof of receipt response shall be sent using SMS-Deliver-Report; no security applied to proof of receipt Data in plain text = "TEST"	The applet posts application data and calls the ProactiveHandler.send() method to send a "Display Text" proactive command with the data received in the Envelope.	The SIM answers to the Envelope with status words 9Fxx and a PoR is retrieved with a GetResponse command. The PoR has the application data posted by the application. The SIM answers to the Get Response command with status words 91xx to issue the Display Text "TEST".
4	**Envelope(SMS-PP) 03.48 formatted** Ciphering; Cryptographic checksum; proof of receipt response shall be sent using SMS-Deliver-Report; proof of receipt shall be ciphered Data in plain text = "TEST"	The applet posts application data and calls the ProactiveHandler.send() method to send a "Display Text" proactive command with the data received in the Envelope.	The SIM answers to the Envelope with status words 9Fxx and a PoR is retrieved with a GetResponse command. The PoR has the application data posted by the application. The SIM answers to the Get Response command with status words 91xx to issue the Display Text "TEST".
5	**Envelope(SMS-PP) 03.48 formatted** No ciphering; Wrong Cryptographic checksum; proof of receipt response shall be sent using SMS-Deliver-Report; no security applied to proof of receiptData in plain text = "TEST"	No applet is triggered	The SIM answers to the Envelope with status words 9Exx and a PoR is retrieved with a GetResponse command. The Response Status Code Octet shall be '01'.

6.3.6.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1,2,3,4,5

6.3.7 Envelope Response Posting

6.3.7.1 EVENT_CALL_CONTROL_BY_SIM

Test Area Reference: FWK_ERP_ECCN

6.3.7.1.1 Conformance Requirements

Normal Execution

CRRN1: The SIM Toolkit Framework can't reply busy when an Envelope(Call Control) is sent to the SIM.

6.3.7.1.1 Test Area Files

Test Script: FWK_ERP_ECCN_1.scr

Test Applet: FWK_ERP_ECCN_1.java

FWK_ERP_ECCN_2.java

FWK_ERP_ECCN_3.java

Load Script: FWK_ERP_ECCN_1.ldr

Cleanup Script: FWK_ERP_ECCN_1.clr

Parameter File: FWK_ERP_ECCN_1.par

6.3.7.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Applet 1 is registered on the EVENT_CALL_CONTROL_BY_SIM.** It calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming dialling number into +11 22 33 44.		
	Applet 2 is registered and triggered on the EVENT_MENU_SELECTION. It is suspended (the method send() has been called and no fetch has been performed)		
	Envelope(Call Control) is sent to the SIM	The Applet 1 is triggered.	The SIM answer 9Fxx to the Envelope(Call Control) The dialling number is retrieved with a GetResponse command.
	A Fetch command is sent to the SIM.	The Applet 2's execution shall continue.	The SIM answers to the Get Response command with status words 91xx.
2	**Applet 3 is registered on both the events EVENT_CALL_CONTROL_BY_SIM and EVENT_MENU_SELECTION.** It calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming dialling number into +11 22 33 44.		

Id	Description	API Expectation	APDU Expectation
	Applet 3 is triggered on the EVENT_MENU_SELECTION and is suspended on the send() method.		
	Envelope(Call Control) is sent to the SIM	The applet is triggered on the EVENT_CALL_CONTROL_BY_SIM.	The SIM answer 9Fxx to the Envelope(Call Control) The dialling number is retrieved with a GetResponse command. The SIM answers to the Get Response command with status words 91xx.
	A Fetch command is sent to the SIM.	The applet's execution shall continue.	

6.3.7.1.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1,2

6.3.7.2 EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM

Test Area Reference: FWK_ERP_EMCN

6.3.7.2.1 Conformance Requirements

Normal Execution

CRRN1: The SIM Toolkit Framework can't reply busy when an Envelope(MO-Short Message Control) is sent to the SIM.

6.3.7.2.2 Test Area Files

Test Script: FWK_ERP_EMCN_1.scr

Test Applet: FWK_ERP_EMCN_1.java

FWK_ERP_EMCN_2.java

FWK_ERP_EMCN_3.java

Load Script: FWK_ERP_EMCN_1.ldr

Cleanup Script: FWK_ERP_EMCN_1.clr

Parameter File: FWK_ERP_EMCN_1.par

6.3.7.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Applet 1 is registered on the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM.** `It calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming TP_Destination_Address`		

Id	Description	API Expectation	APDU Expectation
	into +11 22 33 44.		
	Applet 2 is registered and triggered on the EVENT_MENU_SELECTION. It is suspended (the method send() has been called and no fetch has been performed)		
	An Envelope(MO-Short Message Control) is sent to the SIM	The applet 1 is triggered.	The SIM answers 9Fxx to the Envelope(MO-Short Message Control) The TP_Destination_Address is retrieved with a GetResponse command. The SIM answers to the Get Response command with status words 91xx.
	A Fetch command is sent to the SIM.	The applet's execution shall continue.	
2	**Applet 3 is registered on both the events EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM and EVENT_MENU_SELECTION.** It calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming TP_Destination_Address into +11 22 33 44.		
	Applet 3 is triggered on the EVENT_MENU_SELECTION and is suspended on the send() method.		
	An Envelope(MO-Short Message Control) is sent to the SIM	The applet is triggered on the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM.	The SIM answers 9Fxx to the Envelope(MO-Short Message Control) The TP_Destination_Address is retrieved with a GetResponse command. The SIM answers to the Get Response command with status words 91xx.
	A Fetch command is sent to the SIM.	The applet's execution shall continue.	

6.3.7.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1,2

6.3.7.3 EVENT_UNRECOGNIZED_ENVELOPE

Test Area Reference: FWK_ERP_EUEN

6.3.7.3.1 Conformance Requirements

Normal Execution

CRRN1: The EnvelopeResponseHandler is available for the EVENT_UNRECOGNIZED_ENVELOPE.

6.3.7.3.2 Test Area Files

Test Script: FWK_ERP_EUEN_1.scr

Test Applet: FWK_ERP_EUEN_1.java

Load Script: FWK_ERP_EUEN_1.ldr

Cleanup Script: FWK_ERP_EUEN_1.clr

Parameter File: FWK_ERP_EUEN_1.par

6.3.7.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**An applet triggered on the EVENT_UNRECOGNIZED_ENVELOPE calls the EnvelopeResponseHandler.post() method**	The post() method returns no exception	The SIM answers to the Envelope with status words 9Fxx. The data retrieved with the GetResponse command are the ones posted by the applet.

6.3.7.3.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1

6.3.8 Toolkit Installation

6.3.8.1 Timers Allocation

Test Area Reference: FWK_TIN_TMAL

6.3.8.1.1 Conformance Requirements

Normal execution

CRRN1: One toolkit applet can register to several timers, but a timer can only be allocated to one toolkit applet.

Context errors

CRRC1 : Allocated timers shall not exceed the maximum number of timers allowed for this applet instance defined during installation.

CRRC2 : The total number of timers allocated for all the applets shall not exceed 8.

6.3.8.1.2 Test suite files

Test Script: FWK_TIN_TMAL_1.scr

Test Applet: 1. FWK_TIN_TMAL_1.java

2. FWK_TIN_TMAL_2.java

3. FWK_TIN_TMAL_3.java

6.3.8.2 Maximum Text Length for a menu entry

Test Area Reference: FWK_TIN_MLME

6.3.8.2.1 Conformance Requirements

Normal execution

CRRN1: The maximum length of item text string is defined at the installation of the toolkit applet.

Parameters errors

CRRP1: If initMenuEntry length parameter is greater than the allocated space (Maximum Text Length for a menu entry), then a ToolkitException ALLOWED_LENGTH_EXCEEDED is thrown.

CRRP2: If changeMenuEntry length parameter is greater than the allocated space (Maximum Text Length for a menu entry), then a ToolkitException ALLOWED_LENGTH_EXCEEDED is thrown.

6.3.8.2.2 Test suite files

Test Script: FWK_TIN_MLME_1.scr

Test Applet: FWK_TIN_MLME_1.java

Load Script: FWK_TIN_MLME_1.ldr

Cleanup Script: FWK_TIN_MLME_1.clr

Parameter File: FWK_TIN_MLME_1.par

6.3.8.2.1 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**Installation of applet with 2 menus not exceeding the maximum text length** Install one applet with 2 menu entries allowed and max. text length equal to 10. initMenuEntry defined at the install (install) command MenuEntry = "Toolkitap1" Offset = 0 Length = 10 NextAction = '00' HelpSupported = false IconQualifier = '00' IconIdentifier = 0		
1	**initMenuEntry with a too large length** initMenuEntry with length equal to 11 MenuEntry = "Toolkitap02" Offset = 0 Length = 11 NextAction = '00' HelpSupported = false IconQualifier = '00' IconIdentifier = 0	ToolkitException ALLOWED_LENGTH_EXCEEDED is thrown	
2	**initMenuEntry with a right length** initMenuEntry with length parameter equal to 10 MenuEntry = "Toolkitap2" Offset = 0 Length = 10 NextAction = '00' HelpSupported = false IconQualifier = '00' IconIdentifier = 0		a SET UP MENU (2 items) is issued with TLV item length equal to 11 (Identifier + Text string of item)
3	**changeMenuEntry with a too large length**	ToolkitException	Shall not receive a SET UP

Id	Description	API Expectation	APDU Expectation
	Applet 1 is triggered by a EVENT_MENU_SELECTION. ChangeMenuEntry of menu 1, with length parameter equal to 11 Id = '02' MenuEntry = " Toolkitap04" Offset = 0 Length = menuEntry.length NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0 Return from processToolkit	ALLOWED_LENGTH_EXCEEDED is thrown	MENU different from the previous one
4	**changeMenuEntry with a right length** Applet 1 is triggered by a EVENT_MENU_SELECTION. changeMenuEntry of menu 1, with length parameter equal to 10 Id = '01' MenuEntry = " Toolkitap3" Offset = 0 Length = menuEntry.length NextAction = 0 HelpSupported = false IconQualifier = 0 IconIdentifier = 0 Return from processToolkit		a SET UP MENU (2 items) is issued with TLV item length equal to 11 (Identifier + Text string of item)

6.3.8.2.3 Test Coverage

CRR number	Test case number
CRRN1	2, 4
CRRP1	1
CRRP2	3

6.3.8.3 Maximum number of menu entries

Test Area Reference: FWK_TIN_NBME

6.3.8.3.1 Conformance Requirements

Normal execution

CRRN1: The maximum number of menu entries is defined at the installation of the toolkit applet and can be the maximum number of invocations of the method initMenuEntry .

Parameters errors

CRRP1: If the menu entry cannot be initialised (e.g. no more item data in applet loading parameter), a ToolkitException with the REGISTRY_ERROR reason code is thrown.

6.3.8.3.2 Test suite files

Test Script: FWK_TIN_NBME_1.scr

FWK_TIN_NBME_2.scr

Test Applet: FWK_TIN_NBME_1.java

FWK_TIN_NBME_2.java

Load Script: FWK_TIN_NBME_1.ldr

FWK_TIN_NBME_2.ldr

Cleanup Script: FWK_TIN_NBME_1.clr

FWK_TIN_NBME_2.clr

Parameter File: FWK_TIN_NBME_1.par

FWK_TIN_NBME_2.par

6.3.8.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Installation of applet with 3 menus** Install (install) applet with max. number of menu entry is '3', defined at the install (install) command. initMenuEntry for each menu entry allowed (3 times) MenuEntry = "menu1", "menu2", "menu3" Offset = 0 Length = 5 NextAction = '00' HelpSupported = false IconQualifier = '00' IconIdentifier = 0	No Exception is thrown	
2	**init of a 4th menu** initMenuEntry one more time MenuEntry = "menu4" Offset = 0 Length = 5 NextAction = '00' HelpSupported = false IconQualifier = '00' IconIdentifier = 0	ToolkitException REGISTRY_ERROR is thrown	SET UP MENU (3 items) is issued with TLV item length equal to 6 (Identifier + Text string of item)
3	**Installation of 2nd applet with 0 menu** Install (install) another instance of the same applet, with max. number of menu entry is '0', defined at the install (install) command. initMenuEntry once MenuEntry = "menu1" Offset = 0 Length = 5 NextAction = '00' HelpSupported = false IconQualifier = '00' IconIdentifier = 0	ToolkitException REGISTRY_ERROR is thrown	Shall not receive a SET UP MENU different from the previous one

6.3.8.3.4 Test Coverage

CRR number	Test case number
CRRN1	1
CRRP1	2, 3

6.3.8.4 Access Domain

Test Area Reference: FWK_TIN_ACDO

6.3.8.4.1 Conformance Requirements

Normal execution

CRRN1: The Access Domain parameter indicates the mechanism used to control the applet instance access to the GSM file System (‘00’ means full access to the GSM File System, ‘FF’ means no access to the GSM File System).

Parameters errors

CRRP1: If the Access Domain Parameter requested is not supported, the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

CRRP2: If an applet with Access Domain Parameter 'FF' (i.e. No Access to the GSM File System) tries to access a GSM file (e.g. invoke the updateBinary(..) method) the framework shall throw a SIMViewException with a AC_NOT_FULFILLED reason.

6.3.8.4.2 Test suite files

Test Script: FWK_TIN_ACDO_1.scr

Test Applet: FWK_TIN_ACDO_1.java

Load Script: FWK_TIN_ACDO_1.ldr

Cleanup Script: FWK_TIN_ACDO_1.clr

Parameter File: FWK_TIN_ACDO_1.par

6.3.8.4.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	Install (install) applet with: - Length of Access Domain field value is ‘1’ - Access Domain Parameter value is ‘00’ (full access to the GSM File System)		
1	**readBinary/readRecord method with full Access Domain Parameter** 1- Select EFTARU file whose Read access condition is ALWAYS Perform the readBinary method: fileOffset = 0 resp = baRead[] respOffset = 0 respLength = 3 2- Select EFSMS file whose Read access condition is CHV1 Perform the readRecord method: fileOffset = 0 resp = baRead[] respOffset = 0 respLength = 3 3- Select EFTRAC file whose Read access condition is CHV2 Perform the readBinary method: fileOffset = 0 resp = baRead[] respOffset = 0 respLength = 3 4- Select EFSUME file Read access condition is ADM0	1 to 4- no exception is thrown 5- SIMViewException AC_NOT_FULFILLED is thrown	

Id	Description	API Expectation	APDU Expectation
	Perform the readBinary method: fileOffset = 0 resp = baRead[] respOffset = 0 respLength = 3 5- Select EFTNR file whose Read access condition is NEVER Perform the readBinary method: fileOffset = 0 resp = baRead[] respOffset = 0 respLength = 3		
2	**updateBinary/updateRecord method with full Access Domain Parameter** 1- Select EFTNR file whose Update access condition is ALWAYS Perform the updateBinary method: fileOffset = 0 resp = baUpdate[FFFFFF] respOffset = 0 respLength = 3 2- Select EFSMS file whose Update access condition is CHV1 Perform the updateRecord method: fileOffset = 0 resp = baUpdate[] respOffset = 0 respLength = 3 3- Select EFFDN file whose Update access condition is CHV2 Perform the updateBinary method: fileOffset = 0 resp = baUpdate[] respOffset = 0 respLength = 3 4- Select EFSUME file Update access condition is ADM0 Perform the updateBinary method: fileOffset = 0 resp = baUpdate[] respOffset = 0 respLength = 3 5- Select EFTNU file whose Update access condition is NEVER Perform the updateBinary method: fileOffset = 0 resp = baUpdate[] respOffset = 0 respLength = 3	1 to 4- no exception is thrown 5- SIMViewException AC_NOT_FULFILLED is thrown	
3	**invalidate method with full Access Domain Parameter** 1- Select EFTNR file whose Invalidate access condition is ALWAYS Perform the invalidate method 2- Select EFTIAC file whose Invalidate access condition is CHV1 Perform the invalidate method 3- Select EFADN file whose Invalidate access condition is CHV2 Perform the invalidate method	1 to 4- no exception is thrown 5- SIMViewException AC_NOT_FULFILLED is thrown	

Id	Description	API Expectation	APDU Expectation
	4- Select EFSUME file Invalidate access condition is ADM0 Perform the invalidate method 5- Select EFTNU file whose Invalidate access condition is NEVER Perform the invalidate method		
4	**rehabilitate method with full Access Domain Parameter** 1- Select EFTNR file whose Rehabilitate access condition is ALWAYS Perform the rehabilitate method 2- Select EFIMSI file whose Rehabilitate access condition is CHV1 Perform the rehabilitate method 3- Select EFADN file whose Rehabilitate access condition is CHV2 Perform the rehabilitate method 4- Select EFSUME file Rehabilitate access condition is ADM0 Perform the rehabilitate method 5- Select EFTNU file whose Rehabilitate access condition is NEVER Perform the rehabilitate method	1 to 4- no exception is thrown 5- SIMViewException AC_NOT_FULFILLED is thrown	
5	**increase method with full Access Domain Parameter** 1- Select EFCNU file whose Increase access condition is ALWAYS Perform the increase method 2- Select EFACM file whose Increase access condition is CHV1 Perform the increase method 3- Select EFCIAC file whose Increase access condition is CHV2 Perform the increase method 4- Select EFCIAA file Increase access condition is ADM0 Perform the increase method 5- Select EFCNR file whose Increase access condition is NEVER Perform the increase method Delete instance of applet	1 to 4- no exception is thrown 5- SIMViewException AC_NOT_FULFILLED is thrown	
6	**readBinary method with no Access to the GSM File System as Access Domain Parameter** Install (install) with: - Length of Access Domain field value is '1' - Access Domain Parameter value is 'FF' (No access to the GSM File System) Select EFTARU file whose Read access condition is ALWAYS Perform the readBinary method: fileOffset = 0 resp = baRead[] respOffset = 0 respLength = 3	SIMViewException AC_NOT_FULFILLED is thrown	

Id	Description	API Expectation	APDU Expectation
	Delete instance of applet		

6.3.8.4.4 Test Coverage

CRR number	Test case number
CRRN1	1, 2, 3, 4, 5
CRRP1	Not tested
CRRP2	6

6.3.8.5 Priority Level

Test Area Reference: FWK_TIN_PRLV

6.3.8.5.1 Conformance Requirements

Normal execution

CRRN1: The priority specifies the order of activation of an applet compared to the other applet registered to the same event ('01' : Highest priority level, 'FF' : Lowest priority level)

CRRN2: If two or more applets are registered to the same event and have the same priority level, the applets are activated according to their installation date (i.e. the most recent applet is activated first)

6.3.8.5.2 Test suite files

Test Script: FWK_TIN_PRLV_1.scr

Test Applet: FWK_TIN_PRLV_1.java

Load Script: FWK_TIN_PRLV_1.ldr

Cleanup Script: FWK_TIN_PRLV_1.clr

Parameter File: FWK_TIN_PRLV_1.par

6.3.8.5.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	All applets are registered on an EVENT_UNFORMATTED_SMS_PP_ENV event		
1	**Trigger 2 applets with 2 different maximum Priority Levels** Install (install) applet 1 with priority level '2' and applet 2 with priority level '1', from package P. Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Check that applet 2 is triggered before applet 1. A static variable is used to validate triggering order. Delete applets instances		

2	**Trigger 2 applets with 2 different maximum Priority Levels** Install (install) applet 1 with priority level '1' and applet 2 with priority level '2', from package P. Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Check that applet 1 is triggered before applet 2. A static variable is used to validate triggering order. Delete applets instances			
3	**Trigger 2 applets with 2 different 'key' Priority Levels** Install (install) applet 1 with priority level '80' and applet 2 with priority level '7F', from package P. Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Check that applet 2 is triggered before applet 1 A static variable is used to validate triggering order. Delete applets instances			
4	**Trigger 3 applets with the same Priority Level** Install (install) applet 1, 2, 3 in this order with same priority level from package P. Send an Envelope that triggers the 3 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Check that applet 3 is triggered before applet 2 Check that applet 2 is triggered before applet 1. A static variable is used to validate triggering order. Delete applets instances.			
5	**Trigger 2 applets from 2 classes, with 2 different Priority Level** Install (install) applet 1 from class C with priority level '2' Install (install) applet 2 from class D with priority level '1' A static variable is used to validate triggering order Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Check that applet 2 is activated before applet 1 Delete applets instances			
6	**Trigger 2 applets from 2 classes, with the same Priority Level** Install (install) applet 1 from class C with priority level '1' Install (install) applet 2 from class D with priority level '1' A static variable is used to validate triggering order			

	Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Check that applet 2 is activated before applet 1 Delete applets instances		
7	**Trigger 2 applets from 2 packages, with 2 different Priority Level** Install (install) applet 1 from package P with priority level '2' Install (install) applet 2 from package Q with priority level '1' A static variable is used to validate triggering order Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Check that applet 2 is activated before applet 1 Delete applets instances		
8	**Trigger 2 applets from 2 packages, with the same Priority Level** Install (install) applets 1 and 2 in this order, with same priority level Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Check that applet 10 is triggered before applet 9 Delete applets instances		
9	**Trigger 4 applets from 2 packages** Install (install) 2 applets 1 then 2 with priority levels 1 and 2. Send an Envelope that triggers the 2 applets with the EVENT_UNFORMATTED_SMS_PP_ENV event. Check that applet 1 is triggered before applet 2 Install (install) 2 applets 3 then 4 with priority levels 1 and 2. Send an Envelope that triggers the 4 applets. Check that applet 3 is triggered before applet 1, 4, then 2. Delete applets instances		

6.3.8.5.4 Test Coverage

CRR number	Test case number
CRRN1	1, 2, 3, 5, 7, 9
CRRN2	4, 6, 8

6.3.9 File System Context

6.3.9.1 Initial Context

Test Area Reference: FWK_FSC_INIT

6.3.9.1.1 Conformance Requirements

Normal Execution

CRRN1: At the invocation of the processToolkit method of a toolkit applet, the current file is the MF.

6.3.9.1.2 Test Suite Files

Test Script: FWK_FSC_INIT_1.scr

Test Applet: FWK_FSC_INIT_1.java

FWK_FSC_INIT_2.java

Load Script: FWK_FSC_INIT_1.ldr

Cleanup Script: FWK_FSC_INIT_1.clr

FWK_FSC_INIT_2.clr

Parameter File: FWK_FSC_INIT_1.par

6.3.9.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialisation**	Responses ignored.	
1	**MF is the selected DF in processToolkit()** An ENVELOPE APDU containing a formatted SMS PP for Applet 1 is issued to the SIM byte[] fci = new byte[10] fciOffset = 0 fciLength = 7 status()	No exception shall be thrown. Shall return 7. fci shall contain the following part of the FCI structure: < XX XX XX XX 3F 00 01 >	
2	**No EF is selected** rehabilitate ()	SIMView exception shall be thrown with reason NO_EF_SELECTED	
3	**MF is selected even when an applet triggered before selected any other file...** Applets 1 and 2 register to EVENT_DOWNLOAD_USER_ACTIVITY. Applet 1 has higher priority than Applet 2. An ENVELOPE "EVENT - USER ACTIVITY" is sent to the SIM 1 - Applet 1: - is triggered by event_event_download_user_activity - selects DF_GSM and EF_IMSI 2 - Applet 2: - is triggered by event_event_download_user_activity fciOffset = 0 fciLength = 7 status() 3 - rehabilitate ()	1 - No exception shall be thrown. 2 - No exception shall be thrown. Shall return 7. fci shall contain the following part of the FCI structure: < XX XX XX XX 3F 00 01 > 3 - SIMView exception shall be thrown with reason NO_EF_SELECTED	

6.3.9.1.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3

6.3.9.2 Context Preservation (current file)

Test Area Reference: FWK_FSC_CUFI

6.3.9.2.1 Conformance Requirements

Normal execution

CRRN1: When calling the method select (), the current files (file context) of any other applets shall not be changed (see GSM 03.19 [] - §5.2).

CRRN2: The select() methods select a file without changing the current file of any other applet or of the subscriber session.

CRRN3: After invocation of ProactiveHandler.send() method: the current file context of the toolkit applet is unchanged (see GSM 03.19 [] - §5.2.).

6.3.9.2.2 Test Suite Files

Test Script: FWK_FSC_CUFI_1.scr

Test Applet: FWK_FSC_CUFI_1.java

FWK_FSC_CUFI_2.java

Load Script: FWK_FSC_CUFI_1.ldr

Cleanup Script: FWK_FSC_CUFI_1.clr

FWK_FSC_CUFI_2.clr

Parameter File: FWK_FSC_CUFI_1.par

6.3.9.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialisation**	Responses ignored.	
1	**No change to file context by another applet** Applet1 registers to EVENT_FORMATTED_SMS_PP_ENV. Applet2 registers to EVENT_CALL_CONTROL_BY_SIM 1 - Applet 1: - is triggered by a formatted SMS - selects DF_SIMTEST and EF_TARU - fileOffset = 0; dataLength = 2; dataOffset = 0; - buffer = {0xCA, 0xFE } - updateBinary (): first 2 bytes of EF_TARU are written as 'CA FE'. - issues a proactive command "Get Inkey". 2 - An ENVELOPE APDU containing a CALL CONTROL BY SIM is issued to the SIM Applet 2: - is triggered by a CALL CONTROL BY SIM - selects DF_TELECOM and EF_ADN. 3 - The terminal response for Get Inkey reactivates Applet 1: - fileOffset = 0; respLength = 2; respOffset = 0; - readBinary () info buffer2	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - No exception shall be thrown. The value of buffer2 is { 0xCA, 0xFE }	A GET INKEY proactive command is fetched from the SIM
2	**No change to file context by subscriber session** 1 - Applet 1 - issues a proactive command "Get Inkey".	1 - No exception shall be thrown. 3 - No exception shall be thrown. The value of buffer2 is { 0xCA, 0xFE }	1 - A GET INKEY proactive command is fetched from the SIM

Id	Description	API Expectation	APDU Expectation
	2 - Subscriber session selects DF_TELECOM and EF_ADN. 3 - The terminal response for Get Inkey reactivates Applet 1: - fileOffset = 0; respLength = 2; respOffset = 0; - readBinary () info buffer2		
3	**No change by applet of subscriber session context** 1 - Applet 1: - selects DF_SIMTEST and EF_TNU - issues a proactive command “Get Inkey”. 2 - subscriber session reads record 1 of current file (shall be EF_ADN) 3 - The terminal response for Get Inkey reactivates Applet 1, which terminates execution	1 - No exception shall be thrown. 3 - No exception shall be thrown.	1 - A GET INKEY proactive command is fetched from the SIM 2 - READ RECORD absolute number 1 shall read “FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF” (from EF_{ADN})

6.3.9.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1
CRRN2	1,2,3
CRRN3	1,2

6.3.9.3 Context Preservation (current record pointer)

Test Area Reference: FWK_FSC_CURE

6.3.9.3.1 Conformance Requirements

Normal execution

CRRN1: When the seek method is called by one applet, the record pointer of any other applet is not changed.

CRRN2: *updateRecord*: the current record pointer of other applets / subscriber shall not be changed in case of linear fixed EF

CRRN3: *updateRecord*: the record pointer of a cyclic EF shall be changed for all other applets / subscriber to the record number 1.

CRRN4: *readRecord:* read data bytes of the linear fixed or cyclic EF currently selected by the applet without changing the current record pointer of any other applet / subscriber.

CRRN5: *increase*: the last updated record of the cyclic EF currently selected becomes record number 1 for every other applet and subscriber session.

6.3.9.3.2 Test Suite Files

Test Script: FWK_FSC_CURE_1.scr

Test Applet: FWK_FSC_CURE_1.java

FWK_FSC_CURE_2.java

Load Script: FWK_FSC_CURE_1.ldr

Cleanup Script: FWK_FSC_CURE_1.clr

FWK_FSC_CURE_2.clr

Parameter File: FWK_FSC_CURE_1.par

6.3.9.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
0	**SIM Initialisation**	Responses ignored.	
1	**Seek without affecting another record pointer** aApplet1 registers to EVENT_FORMATTED_SMS_PP_ENV Applet 2 registers to EVENT_CALL_CONTROL_BY_SIM 1 - Applet 1: - is triggered by a formatted SMS event - selects DF_SIMTEST and EF_LARU - reads record 2 using NEXT so that the current record pointer is set to record 2 - issues a proactive command, e.g. Get Inkey. 2 - An ENVELOPE APDU containing a CALL CONTROL BY SIM is issued to the SIM Applet 2: - is triggered by a CALL CONTROL event - selects DF_SIMTEST and EF_LARU - performs a seek of pattern {0x55} from beginning forward, which finds record 1. - returns from processToolkit 3 - The terminal response for Get Inkey reactivates Applet 1: - call readRecord() using CURRENT - the record read should still be record 2 of EF_LARU, containing {0xAA, 0xAA, 0xAA, 0xAA}	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - No exception shall be thrown.	1 - A GET INKEY proactive command is fetched from the SIM
2	**updateRecord in linear fixed EF without affecting current pointer of others** 1 - Applet 1: - is triggered by a formatted SMS event - selects DF_SIMTEST and EF_LARU - reads record 2 using NEXT so that the current record pointer is set to record 2 - issues a proactive command, e.g. Get Inkey. 2 - An ENVELOPE APDU containing a CALL CONTROL BY SIM is issued to the SIM Applet 2: - is triggered by a CALL CONTROL BY SIM event - selects DF_SIMTEST and EF_LARU - updates record 1, by using mode “NEXT”. - returns from processToolkit 3 - The terminal response for Get Inkey reactivates Applet 1: - call readRecord() using CURRENT - the record read should still be record 2 of EF_LARU, containing {0xAA, 0xAA, 0xAA, 0xAA}	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - No exception shall be thrown.	1 - A GET INKEY proactive command is fetched from the SIM
3	**readRecord in linear fixed EF without affecting current pointer of others** 1 - Applet 1: - is triggered by a formatted SMS event - selects DF_SIMTEST and EF_LARU - reads record 2 using NEXT so that the current record pointer is set to record 2 - issues a proactive command, e.g. Get Inkey.	1 - No exception shall be thrown. 2 - No exception shall be thrown. 3 - No exception shall be thrown.	1 - A GET INKEY proactive command is fetched from the SIM

Id	Description	API Expectation	APDU Expectation
	2 - An ENVELOPE APDU containing a CALL CONTROL BY SIM is issued to the SIM Applet 2: - is triggered by a CALL CONTROL BY SIM event - selects DF_SIMTEST and EF_LARU - reads record 1, by using mode “NEXT”. - returns from processToolkit 3 - The terminal response for Get Inkey reactivates Applet 1: - call readRecord() using CURRENT - the record read should still be record 2 of EF_LARU, containing {0xAA, 0xAA, 0xAA, 0xAA}		
4	**updateRecord cyclic EF: record pointer changed to record number 1** 1 – The subscriber session selects DF_SIMTEST and EF_CARU - reads record, by using mode “NEXT”. 2 – Applet 1: - is triggered by a formatted SMS event - selects DF_SIMTEST and EF_CARU - readRecord(), by using mode “NEXT”. - issues a proactive command, e.g. Get Inkey. 3 – An ENVELOPE APDU containing a CALL CONTROL BY SIM is issued to the SIM Applet 2: - is triggered by a CALL CONTROL BY SIM event - selects DF_SIMTEST and EF_CARU - updates record using “PREVIOUS” to ’11 11 11’ - returns from processToolkit 4 – The subscriber session - reads record, by using mode “CURRENT”. 5 – The terminal response for Get Inkey reactivates Applet 1: - readRecord() using mode “CURRENT”	2 - No exception shall be thrown. 3 - No exception shall be thrown. 5 - No exception shall be thrown. Value “11 11 11” is read.	1 - The value “AA AA AA” is obtained as a response to READ RECORD. 2 - A GET INKEY proactive command is fetched from the SIM 4 - The value “11 11 11” is obtained as a response to READ RECORD.
5	**increase cyclic EF: last increased record becomes number 1** 1 – The subscriber session selects DF_SIMTEST and EF_CARU - reads record, by using mode “NEXT”. 2 – Applet 1: - is triggered by a formatted SMS event - selects DF_SIMTEST and EF_CARU - readRecord(), by using mode “NEXT”. - issues a proactive command, e.g. Get Inkey. 3 – An ENVELOPE APDU containing a CALL CONTROL BY SIM is issued to the SIM Applet 2: - is triggered by a CALL CONTROL BY SIM event - selects DF_SIMTEST and EF_CARU - increase() with an increment of ’11 11 11’ - returns from processToolkit 4 – The subscriber session - reads record, by using mode “CURRENT”. 5 – The terminal response for Get Inkey	2 - No exception shall be thrown. 3 - No exception shall be thrown. 5 - No exception shall be thrown. Value “22 22 22” is read.	1 - The value “55 55 55” is obtained as a response to READ RECORD. 2 - A GET INKEY proactive command is fetched from the SIM 4 - The value “22 22 22” is obtained as a response to READ RECORD.

Id	Description	API Expectation	APDU Expectation
	reactivates Applet 1: - readRecord() using mode “CURRENT”		

6.3.9.3.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1
CRRN2	2
CRRN3	4
CRRN4	3
CRRN5	5

6.3.10 Other parts transferred to framework from API

6.3.10.1 A handler is a temporary JCRE Entry Point object

Test Area Reference: FWK_API_HEPO

6.3.10.1.1 Conformance Requirement:

Normal execution

CRRN1: The EnvelopeHandler is a Temporary JCRE Entry Point Object (see Javacard 2.1 Runtime Environment (JCRE) Specification [12]).

CRRN2: The EnvelopeResponseHandler is a Temporary JCRE Entry Point Object (see Javacard 2.1 Runtime Environment (JCRE) Specification [12]).

CRRN3: The ProactiveHandler is a Temporary JCRE Entry Point Object (see Javacard 2.1 Runtime Environment (JCRE) Specification [12]).

CRRN4: The ProactiveResponseHandler is a Temporary JCRE Entry Point Object (see Javacard 2.1 Runtime Environment (JCRE) Specification [12]).

6.3.10.1.2 Test suite files

Test Script: FWK_API_HEPO_1.scr

Test Applet: FWK_API_HEPO_1.java

Load Script: FWK_API_HEPO_1.ldr

Cleanup Script: FWK_API_HEPO_1.clr

Parameter File: FWK_API_HEPO_1.par

6.3.10.1.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**EnvelopeHandler.getTheHandler and store it in a static field of the toolkit applet**	SecurityException is thrown	
2	**EnvelopeHandler.getTheHandler and store it in a field of the toolkit applet**	SecurityException is thrown	
3	**EnvelopeResponseHandler.getTheHandler and store it in a static field of the toolkit applet**	SecurityException is thrown	
4	**EnvelopeResponseHandler.getTheHandler and store it in a field of the toolkit applet**	SecurityException is thrown	
5	**ProactiveHandler.getTheHandler and store it in a static field of the toolkit applet**	SecurityException is thrown	
6	**ProactiveHandler.getTheHandler and store it in a field of the toolkit applet**	SecurityException is thrown	
7	**Build and send a DISPLAY TEXT command to be able to get the reference of the ProactiveReponseHandler**		Proactive command fetched and terminal response is

	ProactiveResponseHandler.getTheHandler and store it in a static field of the toolkit applet	SecurityException is thrown	issued
8	**ProactiveResponseHandler.getTheHandler and store it in a field of the toolkit applet**	SecurityException is thrown	

6.3.10.1.4 Test Coverage

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

6.3.10.2 Transaction

Test Area Reference: FWK_API_TRAN

6.3.10.2.1 Conformance Requirement:

Normal execution

CRRN1: A pending toolkit applet transaction at the ProactiveHandler.send() method invocation is aborted..

6.3.10.2.2 Test suite files

Test Script: FWK_API_TRAN_1.scr

Test Applet: FWK_API_TRAN_1.java

Load Script: FWK_API_TRAN_1.ldr

Cleanup Script: FWK_API_TRAN_1.clr

Parameter File: FWK_API_TRAN_1.par

6.3.10.2.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Verify that transaction is aborted when a proactive command is sent** Initialise a byte field with 0x05 Build a display text proactive command. beginTransaction() Update the byte with 0x02 send the proactive command Verify that the byte value is 0x05 JCSystem.getTransactionDepth()	 Shall return 0	 Proactive command fetched and terminal response is issued

6.3.10.2.4 Test Coverage

CRR number	Test case number
N1	1

6.3.10.3 Timer Id between Applets

Test Area Reference: FWK_API_TMID

6.3.10.3.1 Conformance Requirement:

Context errors

CRRC1: The method ToolkitRegistry.releaseTimer() shall throw a ToolkitException with INVALID_TIMER_ID reason if the timer is valid but isn't allocated to this applet.

6.3.10.3.2 Test suite files

Test Script: FWK_API_TMID_1.scr

Test Applet: FWK_API_TMID_1.java

Load Script: FWK_API_TMID_1.ldr

Cleanup Script: FWK_API_TMID_1.clr

Parameter File: FWK_API_TMID_1.par

6.3.10.3.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**During installation :** **First instance allocate a timer and store the returned value in a static field.** **Second instance allocate a timer.** **Trig second instance and try to releaseTimer() with the static field value.**	**releaseTimer() shall throw a ToolkitException with INVALID_TIMER_ID reason**	

6.3.10.3.4 Test Coverage

CRR number	Test case number
N1	1

...

Annex F (Normative): AID numbering and acronyms for Framework tests

F.1 Toolkit Installation Parameters (TIN)

Test Area within the chapter	Acronyms	Numbering on 6 bits
Timer allocation	TMAL	000001
Item identifier	ITID	000010
Item position	ITPO	000011
Access conditions	ACCO	000100
Priority level	PRLV	000101
Maximum length for each menu entry	MLME	000110
Number of menu entries	NBME	000111
Memory space	MESP	001000

F.2 Minimum Handler Availability (MHA)

Test Area within the chapter	Acronyms	Numbering on 6 bits
ProactiveHandler	PAHD	000001
ProactiveResponseHandler	PRHD	000010
EnvelopeHandler	ENHD	000011
EnvelopeResponseHandler	ERHD	000100

F.3 Handler Integrity (HIN)

Test Area within the chapter	Acronyms	Numbering on 6 bits
ProactiveHandler	PAHD	000001
ProactiveResponseHandler	PRHD	000010
EnvelopeHandler	ENHD	000011
EnvelopeResponseHandler	ERHD	000100

F.4 Applet Triggering (APT)

Test Area within the chapter	Acronyms	Numbering on 6 bits
EVENT_PROFILE_DOWNLOAD	EPDW	000001
EVENT_MENU_SELECTION	EMSE	000010
EVENT_MENU_SELECTION_HELP_REQUEST	EMSH	000011
EVENT_FORMATTED_SMS_PP_ENV	EFSE	000100
EVENT_UNFORMATTED_SMS_PP_ENV	EUSE	000101
EVENT_CALL_CONTROL_BY_SIM	ECCN	000110
EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM	EMCN	000111
EVENT_TIMER_EXPIRATION	ETEX	001000
EVENT_UNFORMATTED_SMS_CB	EUCB	001001
EVENT_EVENT_DOWNLOAD_MT_CALL	EDMC	001010
EVENT_EVENT_DOWNLOAD_CALL_CONNECTED	EDCC	001011
EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED	EDCD	001100
EVENT_EVENT_DOWNLOAD_LOCATION_STATUS	EDLS	001101
EVENT_EVENT_DOWNLOAD_USER_ACTIVITY	EDUA	001110
EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE	EDIS	001111
EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS	EDCR	010000
EVENT_UNRECOGNIZED_ENVELOPE	EUEN	010001
EVENT_STATUS_COMMAND	ESTC	010010

F.5 Proactive Command Sending (PCS)

Test Area within the chapter	Acronyms	Numbering on 6 bits
System Proactive commands	SPCO	000001
Interaction with GSM commands	IGCO	000010
Errors during proactive command sending	EPCS	000011

F.6 Envelope Response Posting (ERP)

Test Area within the chapter	Acronyms	Numbering on 6 bits
EVENT_CALL_CONTROL_BY_SIM	ECCN	000001
EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM	EMCN	000010
EVENT_UNRECOGNIZED_ENVELOPE	EUEN	000011

F.7 Framework Security (FWS)

Test Area within the chapter	Acronyms	Numbering on 6 bits
Input data	INDA	000001
Output data	OUDA	000010

F.8 File System Context (FSC)

Test Area within the chapter	Acronyms	Numbering on 6 bits
Initial Context	INIT	000001
Context Preservation for Current File	CUFI	000010
Context Preservation for Current Record	CURE	000011

F.9 Exception Handling (EXH)

Test Area within the chapter	Acronyms	Numbering on 6 bits
Hide exception to the mobile	HEME	000001
Interaction with multi-triggering	IMTG	000010

F.10 Other parts transferred to framework from API (API)

Test Area within the chapter	Acronyms	Numbering on 6 bits
A handler is a temporary JCRE Entry Point object	HEPO	000001
Transaction	TRAN	000010
Timer Id between Applets	TMID	000011

Annex G (Normative): Configuration Parameters File

This file describes all the mandatory and optional parameters that are used in order to create the loading script(s) for one test area. The configuration parameters file contains the values for the parameters needed in order to generate the loading and cleanup scripts.
The name of the parameters file will be **<*test area reference*>_<*n*>.par**.
The number <n> is associated with the loading/cleanup script number, i.e. API_2_TKR_ SEVL_BSS_1.par is used to generate API_2_TKR_ SEVL_BSS_1.ldr etc.

Syntax

The general syntax for this file will be:

```
<file> ::= <section>+
<section> ::= <section heading> <line break> <section body>
<section heading> ::= '[' <name> ']'
<section body> ::= <parameter assignment>+
<parameter assignment> ::= <name> '=' <value> <line break>
```

Where '+' indicates one or more repetitions of the previous syntax element.
Any text included between the symbol ';' and the end of line is considered a comment and ignored by parsing tools.
Empty values are considered valid. They are used to indicate that an optional value is not present.
Names of sections, names of parameters and values are case-sensitive.
Blank spaces and Tabs between tokens are allowed and will be ignored by the parser.
When values represent a sequence of bytes, they are expressed in hexadecimal format, where every 2 digits represent one byte. Blank space between bytes is optional.
Example:

```
; comment

[Section1]
Parameter11 = 00 11 22 33
Parameter12 = 0101   ; another comment

[Section2]
Parameter21 = vvwwxxyyzz
```

File Contents and Organisation

Parameters in this file are organised in the following sections:

[CONVERT]	Conversion parameters used during conversion (i.e. CAP file generation)
[INSTALL(load)]	Parameters used by the Install for Load command
[LOAD]	Parameters used by the Load command
[INSTALL(install)]	Parameters used by the Install for Install command

All sections may appear only once in the file, except for the "INSTALL(install)" section. If that section appears more than once, it will apply to different applet instances, in sequence.

Default values, order and processing

The ordering of the parameters and the sections is relevant, since parameter names may be repeated and apply to different applets.
When one single parameter is repeated within one section, it refers to different applets. The value of the n^{th} appearance of the parameter applies to applet *n*.
When one section is repeated (INSTALL(install)), then the n^{th} appearance of the section applies to applet *n*.
Parameter/value pairs which are found in one appearance of the section are valid for the subsequent applets as long as they are not overriden. For example, first INSTALL(install) may contain all values for parameters, whereas the subsequent INSTALL(install) sections may only contain parameters whose values change.
If one required parameter is missing from one section, the last defined value of this parameter in a previous section of the same file will be used.

CONVERT Section

These parameters allow configuration of the conversion process of the Java class file(s) into one CAP file.

Parameter	Description
PackageAID	AID of the package
PackageName	Fully qualified name of the package
PackageVersion	Version of the package
AppletClassAID	AID of the applet
AppletClassName	Name of the applet

INSTALL(load) Section

Here are the parameters to be included in the Install(Load) command (as specified in GSM 03.48 [8]).

Parameter	Description
PackageAID	AID of the package
PackageNonVolatileMemSize	Non Volatile memory space (in bytes) required for package loading
InstallationNonVolatileMemSize	Non volatile memory required for installation, in bytes
InstallationVolatileMemSize	Volatile memory required for installation, in bytes

LOAD Section

Here are the parameters to be included in the Load command (as specified in GSM 03.48 [8]).

Parameter	Description
MaxLoadCommandDataLength	Maximum length of the data provided in the load command (P3 parameter of the LOAD APDU embedded in the command packet)

INSTALL(install) Section

Here are the parameters to be included in the Install(Install) command (as specified in GSM 03.48 [8]

Parameter	Description
PackageAID	AID of the package
AppletClassAID	AID of the applet
InstanceAID	AID of the instance of the applet
InstallationNonVolatileMemSize	Non volatile memory required for installation, in bytes
InstallationVolatileMemSize	Volatile memory required for installation, in bytes
AccessDomain	Specify the SIM files that may be accessed by the applet and the operations allowed on these files. This parameter includes the Access Domain Parameter (ADP) and Access Domain Data (ADD)
PriorityLevel	Priority level of the Toolkit applet instance
MaxNumberOfTimers	Maximum number of timers allowed for this applet instance
MaxMenuEntryTextLength	Maximum text length for a menu entry
MaxNumberOfMenuEntries	Maximum number of menu entries allowed for this applet instance
MenuEntriesPositionIdentifier	For each menu entry: Position and identifier of that menu entry
AppletSpecificParameters	Parameters specific to the applet

The applet shall be installed with install(install and make selectable) command.

Full example

```
[CONVERT]
PackageAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 00
PackageName = sim.test.access.api_1_svw_updrbs
PackageVersion = 1.0
AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 01
AppletClassName = API_1_SVW_UPDRBS_1
AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 02
AppletClassName = API_1_SVW_UPDRBS_2

[INSTALL(load)]
PackageNonVolatileMemSize = 0D27
;InstallationNonVolatileMemSize = 0400
;InstallationVolatileMemSize = 0000
```

```
[LOAD]
MaxLoadCommandDataLength = 6C ; max value

[INSTALL(install)]
AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 01
InstanceAID =    A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 01
InstallationNonVolatileMemSize = 0400
InstallationVolatileMemSize = 0000
AccessDomain = 00
PriorityLevel = FF
MaxNumberOfTimers = 00
MaxMenuEntryTextLength = 10
MaxNumberOfMenuEntries = 01
MenuEntriesPositionIdentifier = 0001
AppletSpecificParameters =

[INSTALL(install)]
AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 02
InstanceAID =    A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 02
InstallationNonVolatileMemSize = 0200
InstallationVolatileMemSize = 0000
MenuEntriesPositionIdentifier = 0002
```

```
; rest of INSTALL(install) parameters are taken from previous INSTALL(install)...
```

Annex ~~F~~ H (informative): Change history

[…]

3GPP T3 (USIM) Meeting #21
Kyoto, Japan, 5-7 November, 2001

Tdoc T3-010772
(revised from Tdoc T3-010667)

CR-Form-v3

CHANGE REQUEST

⌘ **11.13** **CR** **004** ⌘ rev **-** ⌘ Current version: **7.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM [X] ME/UE [ ] Radio Access Network [ ] Core Network [ ]

Title:	⌘	CR for API part in 3GPP T3 11.13 specification
Source:	⌘	T3
Work item code:	⌘	***Date:*** ⌘ 30/10/01
Category:	⌘	**F** ***Release:*** ⌘ R98

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change: ⌘
- Alignment with 03.19 wording
- It is not possible to perform some test cases with the actual Script file syntax and format description.
- The addition of new files in prepersonalition Annex are neccesary to perform some test cases

Summary of change: ⌘

Test API:
- Changes in conformance requirements in order to match the wording in 03.19

Script syntax:
- In order to be able to implement tests for cases where more than one behavior is allowed by the specification, a new conditional construction must be added in the syntax description.
- The conditional SWI command is added.
- The unused OFF command is suppressed.
- The whole definition of the syntax is converted to a formal, Backus-Naur Form.

Prepersonalisation :
- New files added

Test suite files :
- Parameter files have been eliminated

Consequences if not approved: ⌘ Potential interoperability problems

Clauses affected: ⌘

		Chapter 6.2.9.2.1 : change in CRRN1 Chapter 6.2.9.4.1 : change in CRRN1 Chapter 6.2.9.5.1 : change in CRRN1 Annex B : Syntax for script file change Annex C : Prepersonalisation files change : some files added (C.2.15 to C.2.18) Annex E : suppression of all parameters files	
Other specs Affected:	⌘	☐ Other core specifications ☐ Test specifications ☐ O&M Specifications	⌘
Other comments:	⌘		

Chapter 6.2.9.2.1 : change in CRRN1

6.2.9.2 Method changeMenuEntry

Test Area Reference: API_2_TKR_CMETB_BSSBZBS

6.2.9.2.1 Conformance requirement:

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

```
public void changeMenuEntry(byte id,
                            byte[] menuEntry,
                            short offset,
                            short length,
                            byte nextAction,
                            boolean helpSupported,
                            byte iconQualifier,
                            short iconIdentifier)
                  throws    java.lang.NullPointerException,
                            java.lang.ArrayIndexOutOfBoundsException,
                            ToolkitException
```

Normal execution

CRRN1: The SIM Toolkit Framework shall ~~automatically~~dynamically ~~-~~update the menu stored in the ME by issuing a SET UP MENU proactive command.The later will reflect the changes done for the entry. The SIM Toolkit Framework shall use the data of the EF sume file in order to build the SET UP MENU command.

CRRN2: The default state of the changed menu entry is 'enabled'.

CRRN3: a call to isEventSet() method on EVENT_MENU_SELECTION shall return true before and after the call.

CRRN4: if helpSupported was true then a call to isEventSet() method on EVENT_MENU_SELECTION_HELP_REQUEST event shall return true.

CRRN5: if helpSupported was true then after the completion of the SETUP MENU command, if an ENVELOPE(MENU_SELECTION_HELP_REQUEST) command is received by the SIM for this entry, then the SIM Toolkit framework shall trigger the applet.

CRRN6: if help supported was true, the SIM Toolkit Framework shall issue a SETUP MENU command with command qualifier = '80'

CRRN7: if helpSupported was false and if no entries is supporting help then a call to isEventSet() method on EVENT_MENU_SELECTION_HELP_REQUEST event shall return false .

CRRN8: if helpSupported was false and if no entries is supporting help then after the completion of the SETUP MENU command, if an ENVELOPE(MENU_SELECTION_HELP_REQUEST) command is received by the SIM, then the SIM Toolkit framework shall not trigger the applet.

CRRN9: The SIM Toolkit Framework shall supply in the SET UP MENU command with the icon identifier provided in the icon identifier list within the item icon identifier list Simple TLV if all the applets registered to the EVENT_MENU_SELECTION provide it.

CRRN10: The SIM Toolkit Framework shall set in the SET UP MENU command with the Icon list qualifier transmitted to the ME as 'icon is not self explanatory' if one of the applet registered prefers this qualifier.

CRRN11: If Next Action Indicator was different from '00', the SIM Toolkit Framework shall issue a SETUP MENU proactive command containing an Items Next Action Indicator simple TLV with the comprehension flag set to 0 as defined in GSM 11.14 [4].

Parameters error

CRRP1: Shall throw java.lang.NullPointerException - if menuEntry is null

CRRP2: Shall throw java.lang.ArrayIndexOutOfBoundsException - if offset would cause access outside array bounds

CRRP3: Shall throw java.lang.ArrayIndexOutOfBoundsException - if length would cause access outside array bounds

CRRP4: Shall throw java.lang.ArrayIndexOutOfBoundsException - if both offset and length would cause access outside array bounds

Context errors

CRRC1: Shall throw a ToolkitException with MENU_ENTRY_NOT_FOUND reason if the Menu Identifier isn't associated to the calling applet instance.

CRRC2: Shall throw ALLOWED_LENGTH_EXCEEDED if the menu entry string is bigger than the allocated space.

Chapter 6.2.9.4.1 : change in CRRN3

6.2.9.4.1 Conformance requirement:

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

```
public void disableMenuEntry(byte id)
        throws ToolkitException
```

Normal execution

CRRN1: A call to isEventSet() method on EVENT_MENU_SELECTION shall return the same result before and after the call to disableMenuEntry() method.

CRRN2: A call to isEventSet() method on EVENT_MENU_SELECTION_HELP_REQUEST shall return the same result before and after the call to disableMenuEntry() method.

CRRN3: After invocation of this method the SIM Toolkit Framework shall~~ould~~ dynamically update the menu stored in the ME .

CRRN4: After invocation of this method, if there is no more enabled menu entries then the SIM Toolkit framework shall issue a SETUP MENU proactive command containing Item Data Object for Item 1 TLV with a length of zero and no value part.

Parameters error

No requirements

Context errors

CRRC1: shall throw a ToolkitException with reason = ENTRY_NOT_FOUND if the menu entry doesn't exist for this applet

Chapter 6.2.9.5.1 : change in CRRN3

6.2.9.5.1 Conformance requirement:

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

```
public void enableMenuEntry(byte id)
        throws ToolkitException
```

Normal execution

CRRN1: A call to isEventSet() method on EVENT_MENU_SELECTION shall return the same result before and after the call to enableMenuEntry() method.

CRRN2: A call to isEventSet() method on EVENT_MENU_SELECTION_HELP_REQUEST shall return the same result before and after the call to enableMenuEntry() method.

CRRN3:The SIM Toolkit Framework shall~~ould~~ dynamically issue a SETUP MENU proactive command which does contain an ITEM SIMPLE TLV object for this entry.

Parameters error

No requirements

Context errors

CRRC1: shall throw a ToolkitException with reason = MENU_ENTRY_NOT_FOUND if the menu entry doesn't exist for this applet

Annex B (normative):
Script file syntax and format description

B.1 Syntax description

Following is a syntax description in BNF.

<statement list> ::= [<statement> **\n**] +
<statement> ::= <simple> | <switch> | <*blank line*>
<simple> ::= <reset> | <init> | <command> | <remark>
<reset> ::= **RST**
<init> ::= **INI** <*hexdata*>
<command> ::= **CMD** <*hexdata*> [<response>] (<status>)
<response> ::= **[** <*hexdata*> **]**
<status> ::= **(** <*hexdata*> **)**
<remark> ::= **REM** <*text line*>
<switch> ::= **SWI { [**<labelled list>] + **}**
<labelled list> ::= <label> **: \n** <statement list>

Description of syntax metalanguage :

\n	represents a linebreak
[x]	means x can appear optionally
[x] +	means 1 or more appearances of x
x | y	means x or y
[]{}:	(bold) these are characters that appear literally in the script files
<*text line*>	any character until the end of the line
<*blank line*>	a line containing no text is acceptable
<*hexdata*>	data written in hexadecimal, each byte separated from the following by a whitespace

Each simple statement beginning with 3 characters different than the ones defined indicates another tool command, and shall be ignored by the parser if not recognised.

`' ', '\t'` : Can be used as separator
A long statement can be broken into several lines by using the character '****' at the end of each line which is not the last one in the statement.

For more details refer to the examples in B.3.

B.2 Semantics

Following is the meaning of each of the statements :

CMD : Sends an APDU Command to ~~follow~~ the card, including (optionally) the expected response data and also (optionally) the expected status words SW1, SW2.

RST : Resets and powers on the card

~~**OFF** : Powers off the card~~

INI : Performs the terminal profile with the following data. Afterwards, it shall perform all the fetch and terminal response commands until there is no proactive session in progress.

REM : Used for comments

~~**IFS (***SW***)** : Placed in front of a command and used for conditional command. The following command is to be performed only if status words of the previous command is *SW*.~~

SWI : Activates a switch condition. Every labelled list represents a list of statements to be executed, if the label matches the SW resulting from the previously executed command.

'\n': Empty lines are accepted
' ', '\t' : Can be used as separator
'\' : Continues on next line
XXX : each line beginning with 3 characters indicates other tool command.

Evaluation of expected response and status in the case of a CMD:

<response> data within **[**…**]** data has to be checked, it needs to be present for an outgoing command. Bytes written as XX shall not be checked by the APDU tool.

<status> status contained within **(**…**)** has status to be checked; when several status are valid they shall be separated by commas. Bytes written as XX shall not be checked by the APDU tool.

B.3 Example

```
REM this is an example

RST
INI FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
REM Case 1 example
CMD A0 C2 00 00 00 (91 33 , 69 XX)

REM Case 2 example
CMD A0 C2 B6 00 00 B0 07 \
    [XX XX XX 55 55 XX 55] \
    (91 33 , 679 XX)

CMD A0 C2 00 00 B0 [] (91 33 , 69 XX)
CMD A0 B6 00 00 07 \
    (91 33 , 67 XX)

CMD A0 C0 00 00 1F \
    [10 A0 00 00 00 09 00 02 FF FF FF FF 89 28 A4 05 \
    02 0D CC CC CC CC CC CC CC CC CC CC CC CC ] \
    (90 00)

REM Case 3 example
CMD A0 C2 00 00 33 \
    D1 31 82 02 83 81 06 05 80 11 22 33 44 8B 24 40 \
    08 00 24 23 85 18 41 04 51 10 10 00 00 00 00 13 \
    02 70 00 00 0E 0D 00 00 00 00 28 A4 05 00 00 00 \
    00 00 00 \
    (90 00)

REM Case 4 example with switch statement
CMD 00 A4 04 00 10 \
    A0 00 00 00 09 00 02 FF FF FF FF 89 41 04 44 02 \
    (61 XX14, 6A 82)

SWI {
61 XX:
CMD 00 C0 00 00 14 \
    [10 A0 00 00 00 09 00 02 FF FF FF FF 89 41 04 44 \
    02 02 CC CC] \
    (90 00)

CMD A0 A4 00 00 02 \
    3F 00

6A 82:
RST

}
REM Get Response if previous status words is 61 14
IFS (61 14)\
CMD 00 C0 00 00 14 \
[10 A0 00 00 00 09 00 02 FF FF FF FF 89 41 04 44 \
02 02 CC CC] \
(90 00)
```

~~OFF~~

B.4 Style and formatting

In order to show a common appearance all the scripts shall follow those format rules:

- start always with a 'RST' followed by an 'INI' command.
- The command, data to be checked and status to be checked shall be presented in the following order:

  CMD *COMMAND* [*EXPECTED DATA*] (*EXPECTED STATUS*)
- APDU shall be presented with command (CLA INS P1 P2 P3) in one line and data (if present) in next line grouped 16 bytes per line (see example above).
- The expected data (if present) shall be presented in 16 bytes groups per line (see example above).

Annex C : Prepersonalisation files change : some files added

C.2.14 EF_{CINA} (Cyclic Increase Not Allowed)

Identifier: '6F0D'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: high	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS (see note 1)
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

Note 1: This file will be personalised in a way such that increase is not allowed, as indicated by the FCI byte 8, bit 7 (GSM 11.11: FCI structure of an EF returned by the SELECT command)

C.2.15 EF_{TRAC} (Transparent Read Access Condition CHV2)

Identifier: '6F0E'	Structure: transparent	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	CHV2
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes

C.2.16 EF_{TIAC} (Transparent Invalidate Access Condition CHV1)

Identifier: '6F0F'	Structure: transparent	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ALWAYS
INVALIDATE	CHV1
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes

C.2.17 EF_{CIAC} (Cyclic Increase Access Condition CHV2)

Identifier: '6F10'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	CHV2
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

C.2.18 EF_{CIAA} (Cyclic Increase Access Condition ADM0)

Identifier: '6F11'	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	

Access Conditions:

READ	ALWAYS
UPDATE	ALWAYS
INCREASE	ADM0
INVALIDATE	ALWAYS
REHABILITATE	ALWAYS

Logical Record Number	Description	Default Value	M/O	Length
1	Test Data	00 00 00	M	3 bytes
2	Test Data	00 00 00	M	3 bytes

Annex E (normative):
Test Area files.

See attached file:

- Annex_E_TestAreaFiles_740.zip

T3-010772 Annex_E_TestAreaFiles.zip