3GPP TSG-T plenary meeting #22
Maui, US, 10-12 December 2003

Tdoc TP-030258

Source: T3

Title: CRs to TS 51.013: Test specification for the SIM API for Java Card

Document for: Approval

This document contains the following change requests:

Spec	CR	Rev	Phase	Subject	Cat	new ver.	Doc-2nd-Level
51.013	002	-	Rel-4	Essential corrections	F	4.1.0	T3-031025
51.013	003	-	Rel-5	Essential corrections	F	5.1.0	T3-031026

3GPP TSG-T3 Meeting #29
Dallas, US, 18-21 November 2003

Tdoc ⌘ ***T3-031025***
Revised T3-030926

CR-Form-v7

CHANGE REQUEST

⌘ **51.013 CR 002** ⌘**rev** **-** ⌘ Current version: **4.0.1** ⌘

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

Proposed change affects: UICC apps⌘ ☒ ME □ Radio Access Network □ Core Network □

Title: ⌘ Essential corrections

Source: ⌘ T3

Work item code: ⌘ TEI ***Date:*** ⌘ 21/11/2003

Category: ⌘ **F** ***Release:*** ⌘ Rel-4

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)*
Rel-6	*(Release 6)*

Reason for change: ⌘ Some essential corrections are needed in specification and in tests.

Summary of change: ⌘

- §6.3.8.6.3: test FWK_TIN_ACDO, testcase 5:
  point 1, the selected EF is changed to EF-CNR instead of EF-CNU
  point 5, the selected EF is changed to EF_CNU instead of EF-CNR
- §C.1: Replace EF_IM by EF_IMG
- Change 'Applet 1' by 'Applet1', 'Applet 2' by 'Applet2', 'Applet 3' by 'Applet3'
- Change 'ot' by 'to'
- Annex E, FWK_TIN_PRLV_10A.java, line 123: Change 'true' to 'false'.
- Annex E, FWK_APT_EPDW.scr, FWK_HIN_PRHD.scr: In TERMINAL RESPONSE commands , change "Type of command" value according to prior
  FETCH command.
- Annex E, API_2_MEP_CHEC_BSS.java, line 67: call check(byte[] mask, short offset, short length) instead of check(byte index)
- Annex E, API_2_TKR_ATIM_1.par: change AppletClassName for instances 2 and 3.
- Annex E, FWK_HIN_ENHD.java: Correct the source file to be in accordance with CRRN1.

Consequences if not approved: ⌘ Errors reside in specification and tests.

Clauses affected:	⌘	§6.3.8.6.3, Annex E FWK_TIN_PRLV_10A.java, Annex E FWK_APT_EPDW.scr, Annex E FWK_HIN_PRHD.scr, Annex E, API_2_MEP_CHEC_BSS.java, Annex E API_2_TKR_ATIM_1.par, Annex E FWK_HIN_ENHD.java			
		Y	**N**		
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/specs/CR.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 API Test Plan

6.2.9 Class ToolkitRegistry

6.2.9.1 Method allocateTimer

Test Area Reference: API_2_TKR_ATIM

6.2.9.1.1 Conformance requirement:

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

```
public byte allocateTimer() throws ToolkitException
```

6.2.9.1.1.1 Normal execution

- CRRN1: the returned timer identifier shall be between 01 and 08 inclusive.
- CRRN2: the returned timer identifier shall be different from a previously allocated but not released one.
- CRRN3: The SIM Toolkit Framework shall trigger the applet when receiving an ENVELOPE(TIMER EXPIRATION) command for the allocated timer.
- CRRN4: A call to isEventSet() method for EVENT_TIMER_EXPIRATION should return true if the applet has at least one timer allocated.

6.2.9.1.1.2 Parameters error

No requirements.

6.2.9.1.1.3 Context errors

- CRRC1: Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE if all the timers are allocated.
- CRRC2: Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE if the maximum number of timers have been allocated to this applet according to installation parameter.

6.2.9.1.2 Test suite files

Test Script:	API_2_TKR_ATIM_1.scr
Test Applet:	API_2_TKR_ATIM_1.java
	API_2_TKR_ATIM_2.java
	API_2_TKR_ATIM_3.java

- Installation parameters:
- For this test procedure the non-volatile memory of each instance is 200 (Hexa).
- The maximum timer parameter value is as follows for each applet:
  - applet-1 (API_2_TKR_ATIM_1): 8 timers

- applet~~-~~2 (API_2_TKR_ATIM_2): 4 timers
- applet~~-~~3 (API_2_TKR_ATIM_3): 0 timer

Load Script: API_2_TKR_ATIM_1.ldr

- The load script installs the 6 instances.

Cleanup Script: API_2_TKR_ATIM_1.clr

Parameter File: API_2_TKR_ATIM_1.par

6.2.9.1.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Allocates up to 8 timers (applet~~-~~1)** `8 * allocateTimer().`	No exception shall be thrown. Timer ID returned shall be between 01 and 08 inclusive. It shall be different after each call.	
2	**Allocate timers more than the maximum (applet~~-~~1)** `The applet~~-~~1 allocates 1 more timer.`	Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE.	

3	**Check applet is Triggered by ENVELOPE(TIMER_EXPIRATION) command (applet1)**
	`Send ENVELOPE(TIMER EXPIRATION) with all timers id (not in an increase order).`
	`Calls releaseTimer(id) each time a timer expires.`
		Shall trigger each time an ENVELOPE(TIMER EXPIRATION) is sent to the SIM, for Timer ID = '01' to '08'.	

4	**Allocate up to 4 timers (applet~~-~~2)** `4 * allocateTimer().`	No exception shall be thrown. Each time, the returned timer identifier shall be between '01' and '08' inclusive. It shall be different after each call.	
5	**Allocate timers more than the maximum (~~applet 3~~applet3)** `The ~~applet 3~~applet3 allocates 1 more timer.`	Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE.	

6.2.9.1.4 Test Coverage

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

6.3 SIM Toolkit Framework

6.3.2 Handler Integrity

6.3.2.2 ProactiveResponseHandler

Test Area Reference: FWK_HIN_~~-~~PRHD

6.3.2.2.1 Conformance Requirement

6.3.2.2.1.1 Normal Execution

- CRRN1: The ProactiveResponseHandler content is changed after the call to ProactiveHandler.send method and remains unchanged until next call to the ProactiveHandler.send method.
- CRRN2: The ProactiveResponseHandler may not be available before the first call to ProactiveHandler.send method, if available the content is cleared.

6.3.2.2.1.2 Parameters error

No requirements.

6.3.2.2.1.3 Context Errors

No requirements.

6.3.2.2.2 Test Suite Files

Test Script:	FWK_HIN_PRHD_1.scr
Test Applet:	FWK_HIN_PRHD_1.java
Load Script:	FWK_HIN_PRHD_1.ldr
Cleanup Script:	FWK_HIN_PRHD_1.clr
Parameter File:	FWK_HIN_PRHD_1.par

6.3.2.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration and ProactiveResponseHandler obtaining** 1-Applet is registered to all events defined in [7]. Using the methods initMenuEntry for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION and setEventList() for the rest of the events. Terminal Profile command is sent to the SIM without the facilities of SET_EVENT_LIST ,SETUP_IDLE_MODE_TEXT, SETUP_MENU and POLL_INTERVAL. For each event: 2-ProactiveResponseHandler.getTheHandler() is called If handler is available, ProactiveResponseHandler.getLength() is called	1- No exception is thrown 2- Applet is triggered. 3- Behaviour 1: Toolkit Exception HANDLER_NOT_AVAILABLE is thrown. Behaviour 2: No exception is thrown, the return value is 0	
2	**The ProactiveResponseHandler remains unchanged after send method invocation until next send method invocation** 1-Applet builds a proactive command ProactiveHandler.send() method is called	1- The ProactiveResponseHandler contains the terminal response	2- A proactive command is fetched

Id	Description	API/Framework Expectation	APDU Expectation

	`2-ProactiveResponseHandler.getLength() method is called`
	`3-ProactiveHandler.init() method is called`
	`4-ProactiveHandler.send() method is called`
	`5-ProactiveResponseHandler.getLength() method is called`
		3- The return value is 12
		4- No exception is thrown and the Proactive Response Handler remains unchanged
		5- The ProactiveResponseHandler contains the terminal response of the second proactive command
		7- The return value is 15
			The terminal response is sent with length 12
			6- A proactive command is fetched
			The terminal response is sent with length 15

6.3.2.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	1

6.3.3 Applet Triggering

6.3.3.6 EVENT_CALL_CONTROL_BY_SIM

Test Area Reference: FWK_APT_ECCN

6.3.3.6.1 Conformance Requirement

6.3.3.6.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_CALL_CONTROL_BY_SIM once it has registered to this event and an Envelope Call Control is received.
- CRRN2: The applet is not triggered by the EVENT_CALL_CONTROL_BY_SIM once it has deregistered from this event.

6.3.3.6.1.2 Parameters error

No requirements.

6.3.3.6.1.3 Context Errors

No requirements.

6.3.3.6.2 Test Suite Files

Test Script:	FWK_APT_ECCN_1.scr
Test Applet:	FWK_APT_ECCN_1.java

Load Script:	FWK_APT_ECCN_1.ldr
Cleanup Script:	FWK_APT_ECCN_1.clr
Parameter File:	FWK_APT_ECCN_1.par

6.3.3.6.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applets registration to EVENT_CALL_CONTROL_BY_SIM and triggering** Applet1 is registered to EVENT_CALL_CONTROL_BY_SIM. Applet2 is registered to EVENT_FORMATTED_SMS_PP_ENV 1-An Envelope Call control by SIM is sent to SIM	1- Applet1 is triggered	
2	**Applet deregistration and registration of the third applet to EVENT_CALL-CONTROL_BY_SIM.** 1-An Envelope Formatted SMS PP envelope is sent to SIM Applet2 contructs a DISPLAY TEXT proactive command. 2-ProactiveHandler.send() method is called 3-An Envelope Call control by SIM envelope is sent to SIM ToolkitRegistry.clearEvent() is called for EVENT_CALL_CONTROL_BY_SIM. ToolkitRegistry.setEvent() method is called for EVENT_CALL_CONTROL_BY_SIM.	1-Applet2 is triggered by EVENT_FORMATTED_SMS_PP_ENV. 3- Applet1 is triggered Applet1 finalizes. Applet2 finalizes	2- A proactive command DISPLAY TEXT is sent and applet is suspended until the terminal response TERMINAL RESPONSE of DISPLAY TEXT is sent to the SIM
3	**Applet triggering** An Envelope Call control by SIM envelope is sent to~~ot~~ SIM	Applet2 is triggered. (Applet1 is not triggered)	

6.3.3.6.4 Test Coverage

CRR Number	Test Case Number

CRRN1	1, 2, 3
CRRN2	3

6.3.3.7 EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM

Test Area Reference: FWK_APT_EMCN

6.3.3.7.1 Conformance Requirement

6.3.3.7.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM once it has registered to this event and an Envelope MO Short Message Control.
- CRRN2: The applet is not triggered by the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM once it has deregistered from this event.

6.3.3.7.1.2 Parameters error

No requirements.

6.3.3.7.1.3 Context Errors

No requirements.

6.3.3.7.2 Test Suite Files

Test Script:	FWK_APT_EMCN_1.scr
Test Applet:	FWK_APT_EMCN_1.java
	FWK_APT_EMCN_2.java
Load Script:	FWK_APT_EMCN_1.ldr
Cleanup Script:	FWK_APT_EMCN_1.clr
Parameter File:	FWK_APT_EMCN_1.par

6.3.3.7.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM and triggering** Applet1 is reggistered to EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM. Applet2 is registered to EVENT_FORMATTED_SMS_PP_ENV. 1-An Envelope MO short message envelope is sent to SIM	1- Applet1 is triggered.	

Id	Description	API/Framework Expectation	APDU Expectation

2	**Applet deregistration and registration of the third applet to EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM.**
	The STF shall not reply busy to a call control envelope
	`1-An Envelope formatted SMS PP envelope is sent to SIM.`
	`Applet2 builds a DISPLAY TEXT proactive command.`
	`2-ProactiveHandler.send() method is called.`
	`3-An Envelope MO Short message envelope is sent to SIM`
	`ToolkitRegistry.clearEvent() for EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM.`
	`ToolkitRegistry.setEvent() method is called for EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM.`
		1- Applet2 is triggered.
		3- Applet1 is triggered.
		Applet1 finalizes.
		Applet2 finalizes.
			2- A Proactive command DISPLAY TEXT is sent and applet is suspended until the terminal response
			TERMINAL RESPONSE of DISPLAY TEXT is sent to the SIM

3	**Applet3 triggering** `An Envelope MO SMS control by SIM envelope is sent ~~ot~~to SIM`	Applet2 is triggered. (Applet1 is not triggered)	

6.3.3.7.4 Test Coverage

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

6.3.3.18 EVENT_STATUS_COMMAND

Test Area Reference: FWK_APT_ESTC

6.3.3.18.1 Conformance Requirement

6.3.3.18.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_STATUS_COMMAND once it has registered to this event and a Status Command is received.
- CRRN2: The applet is not triggered by the EVENT_STATUS_COMMAND once it has deregistered from this event.

6.3.3.18.1.2 Parameters error

No requirements.

6.3.3.18.1.3 Context Errors

No requirements.

6.3.3.18.2 Test Suite Files

Test Script:	FWK_APT_ESTC_1.scr
Test Applet:	FWK_APT_ESTC_1.java
	FWK_APT_ESTC_2.java
	FWK_APT_ESTC_3.java
Load Script:	FWK_APT_ESTC_1.ldr
Cleanup Script:	FWK_APT_ESTC_1.clr
Parameter File:	FWK_APT_ESTC_1.par

6.3.3.18.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applets registration to EVENT_STATUS_COMMAND and triggering** Applet1 is registered to EVENT_STATUS_COMMAND using the requestPollInterval() command. Applet2 is registered to EVENT_STATUS_COMMAND using the RequestPollInterval() command. Applet3 is registered to EVENT_FORMATTED_SMS_PP_ENV. 1-A status command is sent to SIM	1- Applet1 is triggered. Applet1 finalizes 2- Applet2 is triggered. Applet2 finalizes 3- ~~Applet 3~~Applet3 is not triggered	

Id	Description	API/Framework Expectation	APDU Expectation
2	**Applet deregistration and registration of the third applet to EVENT_STATUS_COMMAND. The STF shall not reply busy to a call control envelope** 1-A formatted sms pp envelope is sent to SIM Applet3 builds a DISPLAY TEXT. 2- ProactiveHandler.send() is called 3-A status command is sent to SIM. requestPollInteval with POLL_NO_DURATION is called requestPollInteval with POLL_NO_DURATION is called requestPollInterval() method is called.	1- Applet3 is triggered. 3- Applet1 is triggered. Applet1 finalizes 4- Applet2 is triggered. Applet2 finalizes Applet3 finalizes	2- A proactive command DISPLAY TEXT is sent and applet is suspended until the terminal response 5- TERMINAL RESPONSE of DISPLAY TEXT is sent to the SIM
3	**Applet3 triggering** Perform SIM initialization with all the facilities supported Status command is sent to SIM.	Applet3 is triggered. (Applet1 and Applet2 are not triggered)	

6.3.3.18.4 Test Coverage

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

6.3.6 Framework Security Management

Security Parameters

The table that follows contains the security parameters that shall be used when the TS 23.048 [8]security is required in the test cases developed in the current subclause.

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 subclause 4.7.3.1 apply.

6.3.6.1 Input Data

Test Area Reference: FWK_FWS_INDA

6.3.6.1.1 Conformance Requirements

6.3.6.1.1.1 Normal Execution

- CRRN1: If the SIM receives an envelope APDU containing an SMS_PP_DATADOWNLOAD BER TLV formatted according to 3GPP TS 23.048 [8], the SIM Toolkit Framework shall verify the security of the SMS TPDU.
- CRRN2: The toolkit applet will only be triggered if the TAR is known and the security verified.
- CRRN3: If the SIM receives an envelope APDU containing an SMS_CB_DATADOWNLOAD formatted according to 3GPP TS 23.048 [8], the SIM Toolkit Framework shall verify the security of the cell broadcast page.

6.3.6.1.1.2 Parameters error

No requirements.

6.3.6.1.1.3 Context Errors

No requirements.

6.3.6.1.2 Test Area Files

Test Script:	FWK_FWS_INDA_1.scr
Test Applet:	FWK_FWS_INDA_1.java
	FWK_FWS_INDA_2.java
	FWK_FWS_INDA_3.java
	FWK_FWS_INDA_4.java
Load Script:	FWK_FWS_INDA_1.ldr
Cleanup Script:	FWK_FWS_INDA_1.clr
Parameter File:	FWK_FWS_INDA_1.par

Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Framework checks the Cryptographic checksum and deciphers the data** Applet1 is loaded and installed 1-Envelope(SMS-PP) formatted is sent to the SIM with this features: Ciphering; Cryptographic checksum; No proof of receipt; Data = 01	1- Applet1 is triggered.	

Id	Description	API/Framework Expectation	APDU Expectation
2	**Framework checks the Cryptographic checksum and deciphers the data** Applet2 is installed 1-Envelope(SMS-PP) formatted is sent to the SIM with this features: Ciphering; Cryptographic checksum; No proof of receipt; TAR of Applet1 Data = 02 2-Envelope(SMS-PP) 03.48 formatted is sent to the SIM with this features: No ciphering; No cryptographic checksum; No proof of receipt; TAR of Applet2 Data = 03	1- Applet1 is triggered 3- Applet2 is triggered	2- The SIM answers to the Envelope with status words 9000 The SIM answers to the Envelope with status words 9000

3	**Envelope(SMS-PP) formatted with wrong cryptographic checksum**
	No ciphering;
	Wrong Cryptographic checksum;
	No proof of receipt;
	TAR of Applet1
	Data = 04
		No applet is triggered	1- The SIM answers to the Envelope with status words 9000

4	**Framework checks the Cryptographic checksum and deciphers the data** Applet3 is loaded and installed 1-Envelope(SMS-CB) formatted is sent to the SIM with this features: Ciphering; Cryptographic checksum; No proof of receipt; Data = 01	1- Applet3 is triggered.	1- The SIM answers to the Envelope with status words 9000
5	**Triggering two different applets with different security on Envelope(SMS-CB) formatted** Applet4 is installed 1-Envelope(SMS-CB) formatted is sent to the SIM with this features: Ciphering; Cryptographic checksum; No proof of receipt; TAR of ~~Applet 3~~Applet3 Data = 02 2-Envelope(SMS-CB) formatted is sent to the SIM with this features: No ciphering; Cryptographic checksum; No proof of receipt; TAR of ~~Applet 4~~Applet4 Data = 03	1- Applet3 is triggered 3- Applet4 is triggered	2- The SIM answers to the Envelope with status words 9000 4- The SIM answers to the Envelope with status words 9000

6	**Envelope(SMS-CB) formatted with wrong cryptographic checksum**
	No ciphering;
	Wrong Cryptographic checksum;
	No proof of receipt;
	TAR of ~~Applet 3~~Applet3
	Data = 04
		No applet is triggered	1- 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
CRRN3	4, 5, 6

6.3.7 Envelope Response Posting

6.3.7.1 EVENT_CALL_CONTROL_BY_SIM

Test Area Reference: FWK_ERP_ECCN

6.3.7.1.1 Conformance Requirements

6.3.7.1.1.1 Normal Execution

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

6.3.7.1.1.2 Parameters error

No requirements.

6.3.7.1.1.3 Context Errors

No requirements.

6.3.7.1.2 Test Area Files

Test Script:	FWK_ERP_ECCN_1.scr
Test Applet:	FWK_ERP_ECCN_1.java
	FWK_ERP_ECCN_2.java
	FWK_ERP_ECCN_3.java
Load Script:	FWK_ERP_ECCN_1.ldr
Cleanup Script:	FWK_ERP_ECCN_1.clr
Parameter File:	FWK_ERP_ECCN_1.par

6.3.7.1.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet1 is registered on the EVENT_CALL_CONTROL_BY_SIM, Applet2 is registered and triggered on the EVENT_MENU_SELECTION.**		
	`1-Applet2 invokes the method send()and no fetch is performed`	Applet2 is suspended	
	`2-Envelope(Call Control) is sent to the SIM`	Applet1 is triggered.	
	`3-Applet1 calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming dialling number into +11 22 33 44.`		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.
	`4-A Fetch command is sent to the SIM`		
	`5-A Terminal Response command is sent to the SIM`	Applet2's execution shall continue.	
	`6-Delete ~~applet1 ~~Applet1 & ~~applet2~~Applet2`		
	`7-Install ~~applet3~~Applet3`		
2	**~~Applet 3~~Applet3 is registered on both the events EVENT_CALL_CONTROL_BY_SIM and EVENT_MENU_SELECTION.**		
	`1-Envelope Menu Selection is sent to the SIM.`	Applet3 is triggered on the EVENT_MENU_SELECTION	
	`2-Applet3 invokes the method send()and no fetch is performed)`	Applet3 is suspended on the send() method	
	`3-Envelope(Call Control) is sent to the SIM`	Applet3 is triggered on the EVENT_CALL_CONTROL_BY_SIM.	
	`4-Applet3 calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming dialling number into +11 22 33 44.`		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.
	`5-A Fetch command is sent to the SIM`		
	`6-A Terminal Response command is sent to the SIM`	The Applet3'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

6.3.7.2.1.1 Normal Execution

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

6.3.6.2.1.2 Parameters error

No requirements.

6.3.6.2.1.3 Context Errors

No requirements.

6.3.7.2.2 Test Area Files

Test Script:	FWK_ERP_EMCN_1.scr
Test Applet:	FWK_ERP_EMCN_1.java
	FWK_ERP_EMCN_2.java
	FWK_ERP_EMCN_3.java
Load Script:	FWK_ERP_EMCN_1.ldr
Cleanup Script:	FWK_ERP_EMCN_1.clr
Parameter File:	FWK_ERP_EMCN_1.par

6.3.7.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet1 is registered on the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM; Applet2 is registered and triggered on the EVENT_MENU_SELECTION.** ~~.~~ 1-Applet2 invokes the method send()and no fetch is performed) 2-Envelope(MO-SM control) is sent to the SIM 3-Applet1 calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming TP_Destination_Address and any RP_Destination_Address of the Service Center into +11 22 33 44 4-A Fetch command is sent to the SIM 5-A Terminal Response command is sent to the SIM 6-Delete ~~applet1~~ Applet1 & ~~applet2~~Applet2 7-Install ~~applet3~~Applet3	Applet2 is suspended Applet1 is triggered. The Applet's execution shall continue.	The SIM answers 9Fxx to the Envelope(MO-Short Message Control) The TP_Destination_Address is retrieved with a GetResponse command. The SIM answers to the Get Response command with status words 91xx.
2	**~~Applet 3~~Applet3 is registered on both the events EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM and EVENT_MENU_SELECTION.** 1-Applet3 invokes the method send()and no fetch is performed) 2-Envelope(MO-SM control) is sent to the SIM 3-Applet3 calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming TP_Destination_Address and any RP_Destination_Address of the Service Center into +11 22 33 44. 4-A Fetch command is sent to the SIM 5-A Terminal Response command is sent to the SIM	~~Applet 3~~Applet3 is suspended on the send() method Applet3 is triggered on the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM. The Applet3's execution shall continue.	The SIM answers 9Fxx to the Envelope(MO-Short Message Control) The TP_Destination_Address is retrieved with a GetResponse command. The SIM answers to the Get Response command with status words 91xx.

6.3.7.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2

6.3.8 Toolkit Installation

6.3.8.6 Access Domain

Test Area Reference: FWK_TIN_ACDO

6.3.8.6.1 Conformance Requirements

6.3.8.6.1.1 Normal execution

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

6.3.8.6.1.2 Parameters errors

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

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

6.3.8.6.1.3 Context errors

No requirements.

6.3.8.6.2 Test suite files

Test Script:	FWK_TIN_ACDO_1.scr
Test Applet:	FWK_TIN_ACDO_1.java
	FWK_TIN_ACDO_2.java
	FWK_TIN_ACDO_3.java
Load Script:	FWK_TIN_ACDO_1.ldr
Cleanup Script:	FWK_TIN_ACDO_1.clr
Parameter File:	FWK_TIN_ACDO_1.par

6.3.8.6.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
0	Install (install) applet1 with: - Length of Access Domain field value is '1' - Access Domain Parameter value is '00' (full access to the GSM File System) Install (install) applet2 with: - Length of Access Domain field value is '1'		

Id	Description	API/Framework Expectation	APDU Expectation
	- Access Domain Parameter value is 'FF' (No access to the GSM File System) Install (install) applet3 with: - Length of Access Domain field value is '1' - Access Domain Parameter value is '00' (full access to the GSM File System)		

1	**readBinary/readRecord method with full Access Domain Parameter**
	
	1- Select EF-TARU file whose Read access condition is ALWAYS
	Perform the readBinary method:
	fileOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
	
	2- Select EF-SMS file whose Read access condition is CHV1
	Perform the readRecord method:
	
	recNumber = 1
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
	
	3- Select EF-TRAC file whose Read access condition is CHV2
	Perform the readBinary method:
	fileOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
	
	4- Select EF-SUME file Read access condition is ADM0
	Perform the readBinary method:
	fileOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
	
	5- Select EF-TNR file whose Read access condition is NEVER
	Perform the readBinary method:
	fileOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
		1 to 4- no exception is thrown
		
		5- SIMViewException AC_NOT_FULFILLED is thrown	

Id	Description	API/Framework Expectation	APDU Expectation
2	**updateBinary/updateRecord method with full Access Domain Parameter**	1 to 4- no exception is thrown	

3	**invalidate method with full Access Domain Parameter**
	`1- Select EF-TNR file whose Invalidate access condition is ALWAYS`
	`Perform the invalidate method`
	`2- Select EF-TIAC file whose Invalidate access condition is CHV1`
	`Perform the invalidate method`
	`3- Select EF-ADN file whose Invalidate access condition is CHV2`
	`Perform the invalidate method`
	`4- Select EF-SUME file Invalidate access condition is ADM0`
	`Perform the invalidate method`
	`5- Select EF-CNIV file whose Invalidate access condition is NEVER`
	`Perform the invalidate method`
		1 to 4- no exception is thrown
		5- SIMViewException AC_NOT_FULFILLED is thrown	

Id	Description	API/Framework Expectation	APDU Expectation
4	**rehabilitate method with full Access Domain Parameter** 1- Select EF-TNR file whose Rehabilitate access condition is ALWAYS Perform the rehabilitate method 2- Select EF-IMSI file whose Rehabilitate access condition is CHV1 Perform the rehabilitate method 3- Select EF-ADN file whose Rehabilitate access condition is CHV2 Perform the rehabilitate method 4- Select EF-SUME file Rehabilitate access condition is ADM0 Perform the rehabilitate method 5- Select EF-CNRI file whose Rehabilitate access condition is NEVER Perform the rehabilitate method	1 to 4- no exception is thrown 5- SIMViewException AC_NOT_FULFILLED is thrown	

5	**increase method with full Access Domain Parameter**
	
	1- Select EF-CNR~~U~~ file whose Increase access condition is ALWAYS
	Perform the increase method:
	incr = abIncreaseValue[]
	incrOffset = 0
	resp = abRead[]
	respOffset = 0
	
	2- Select EF-ACM file whose Increase access condition is CHV1
	Perform the increase method:
	incr = abIncreaseValue[]
	incrOffset = 0
	resp = abRead[]
	respOffset = 0
	
	3- Select EF-CIAC file whose Increase access condition is CHV2
	Perform the increase method:
	incr = abIncreaseValue[]
	incrOffset = 0
	resp = abRead[]
	respOffset = 0
	
	4- Select EF-CIAA file Increase access condition is ADM0
	Perform the increase method:
	incr = abIncreaseValue[]
	incrOffset = 0
	resp = abRead[]
	respOffset = 0
	
	5- Select EF-CNU~~R~~ file whose Increase access condition is NEVER
	Perform the increase method
		1 to 4- no exception is thrown
		
		5- SIMViewException AC_NOT_FULFILLED is thrown	

Id	Description	API/Framework Expectation	APDU Expectation
6	**readBinary method with no Access Domain Parameter** Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event. Select EF-TARU file whose Read access condition is ALWAYS Perform the readBinary method: fileOffset = 0 resp = abRead[] respOffset = 0 respLength = 3 t	SIMViewException AC_NOT_FULFILLED is thrown	

7	**updateRecord method with no Access Domain Parameter**
	Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event.
	Select EF-SMS file whose Update access condition is CHV1
	Perform the updateRecord method:
	fileOffset = 0
	resp = abUpdate[]
	respOffset = 0
	respLength = 3
		SIMViewException AC_NOT_FULFILLED is thrown	

8	**invalidate method with no Access Domain Parameter** Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event. Select EF-ADN file whose Invalidate access condition is CHV2 Perform the invalidate method	SIMViewException AC_NOT_FULFILLED is thrown	
9	**rehabilitate method with no Access Domain Parameter** Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event. Select EF-SUME file Rehabilitate access condition is ADM0 Perform the rehabilitate method	SIMViewException AC_NOT_FULFILLED is thrown	
10	**increase method with no Access Domain Parameter** Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event. Select EF-CNR file whose Increase access condition is NEVER Perform the increase method	SIMViewException AC_NOT_FULFILLED is thrown Applet2 finalizes Applet3 restore EF-SUME	

6.3.8.6.4 Test Coverage

NOTE: As Item Position management is not fully specified in the 3GPP TS 43.019 [7] or 3GPP TS 23.048 [8] all possible tests cannot be performed.

CRR number	Test case number
CRRN1	1, 2, 3, 4, 5
CRRP1	Not tested
CRRP2	6, 7, 8, 9, 10

Annex C (normative): Default Prepersonalization

C.1 General Default Prepersonalization

This table shows the default prepersonalization, the file system and the files' content, that the test SIM cards shall contain unless otherwise stated.

Name	Identifier	Default Value	Special Features
EF$_{ICCID}$	2FE2	0F FF FF FF FF FF FF FF FF FF	This value is not compliant with 3GPP TS 51.011 [3]
EF$_{IMSI}$	6F07	FF FF FF FF FF FF FF FF FF	This value is not compliant with 3GPP TS 51.011 [3]
EF$_{LP}$	6F05	01 FF FF FF	
EF$_{Kc}$	6F20	FF FF FF FF FF FF FF FF 07	
EF$_{PLMNsel}$	6F30	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
EF$_{HPLMN}$	6F31	05	
EF$_{ACMmax}$	6F37	00 00 00	Access condition UPDATE: CHV1
EF$_{SST}$	6F38	FF 3F C3 0F 0C 00 FF 0F 00 33	
EF$_{ACM}$	6F39	00 00 00	Access condition UPDATE: CHV1
EF$_{PUCT}$	6F41	FF FF FF 00 00	Access condition UPDATE: CHV1
EF$_{BCCH}$	6F74	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
EF$_{ACC}$	6F78	00 00	
EF$_{FPLMN}$	6F7B	FF FF FF FF FF FF FF FF FF FF FF FF	
EF$_{LOCI}$	6F7E	FF FF FF FF 00 F0 00 00 00 FF 01	
EF$_{AD}$	6FAD	00 FF FF	
EF$_{Phase}$	6FAE	03	
EF$_{FDN}$	6F3B	Default value in all the records: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 5
EF$_{SMSP}$	6F42	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF$_{LND}$	6F44	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF$_{SMSS}$	6F43	FF FF	
EF$_{SMS}$	6F3C	1^{st} record: 00 FF … FF(length 176) 2^{nd} record:00 FF … FF(length 176) 3^{rd} record: 00 FF … FF(length 176)	Records: 3
EF$_{ADN}$	6F3A	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF$_{CCP}$	6F3D	FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
EF$_{MSISDN}$	6F40	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF$_{SDN}$	6F49	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF$_{SUME}$	6F54	85 0C 54 4F 4F 4C 4B 49 54 20 54 45 53 54 FF FF FF FF	
EF$_{CBMI}$	6F45	FF FF	
EF$_{CBMID}$	6F48	10 80	

EF_{CBMIR}	6F50	10 80 10 9F	
EF_{IMG}	4F20	FF FF FF FF FF FF FF FF FF FF FF	

The default value for the CHV1 shall be "0x31 0x31 0x31 0x31 0xFF 0xFF 0xFF 0xFF" and its state shall be 'disabled' during test applets execution.

3GPP TSG-T3 Meeting #29
Dallas, US, 18-21 November 2003

Tdoc ⌘ T3-031026
Revised T3-030927

CR-Form-v7

CHANGE REQUEST

⌘	**51.013**	**CR**	**003**	⌘ **rev**	**-**	⌘	Current version:	**5.0.1**	⌘

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

Proposed change affects: UICC apps⌘ ☒ ME □ Radio Access Network □ Core Network □

Title: ⌘ Essential corrections

Source: ⌘ T3

Work item code: ⌘ TEI

Date: ⌘ 21/11/2003

Category: ⌘ **F**

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.

Release: ⌘ Rel-5

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)*
Rel-6	*(Release 6)*

Reason for change: ⌘ Essential changes in specification and in tests, and an essential correction in a test writing needed.

Summary of change: ⌘

- §6.3.2.3.3: test case 16 is redundant with test case 19. Suppress test case 19 and update test coverage table accordingly (§6.3.2.3.4)
- §6.3.8.6.3: test FWK_TIN_ACDO, testcase 5:
  point 1, the selected EF is changed to EF-CNR instead of EF-CNU
  point 5, the selected EF is changed to EF_CNU instead of EF-CNR
- §C.1: Replace EF_IM by EF_IMG
- Change ‘Applet 1’ by ‘Applet1’, ‘Applet 2’ by ‘Applet2’, ‘Applet 3’ by ‘Applet3’
- Change ‘ot’ by ‘to’
- Annex E, FWK_FWS_INDA.ldr: Suppress inserted lines between data and Satus Word.
- Annex E, FWK_TIN_PRLV_10A.java, line 123: Change ‘true’ to ‘false’.
- Annex E, FWK_PCS_PCCO.scr, FWK_APT_EPDW.scr, FWK_HIN_PRHD.scr: In TERMINAL RESPONSE commands , change "Type of command" value according to prior FETCH command.
- Annex E, API_2_MEP_CHEC_BSS.java, line 67: call check(byte[] mask, short offset, short length) instead of check(byte index)
- Annex E, API_2_TKR_ATIM_1.par: change AppletClassName for

		instances 2 and 3. • Annex E, API_2_PRH_CCHD_BSS_1.java: relocate ProactiveResponseHandler.getTheHandler() method call after the first send() method, in order to be in accordance with ProactiveResponseHandler definition. • Annex E, FWK_HIN_ENHD.java: Correct the source file to be in accordance with CRRN1.
Consequences if not approved:	⌘	Errors reside in specification and tests.

Clauses affected:	⌘	§6.3.2.3.3, §6.3.8.6.3, §C.1, Annex E FWK_FWS_INDA.ldr, Annex E FWK_TIN_PRLV_10A.java, Annex E FWK_PCS_PCCO.scr, Annex E FWK_APT_EPDW.scr, Annex E FWK_HIN_PRHD.scr, Annex E, API_2_MEP_CHEC_BSS.java, Annex E API_2_TKR_ATIM_1.par, Annex E API_2_PRH_CCHD_BSS_1.java, Annex E FWK_HIN_ENHD.java				
		Y	**N**			
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/specs/CR.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 API Test Plan

6.2.9 Class ToolkitRegistry

6.2.9.1 Method allocateTimer

Test Area Reference: API_2_TKR_ATIM

6.2.9.1.1 Conformance requirement:

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

```
public byte allocateTimer() throws ToolkitException
```

6.2.9.1.1.1 Normal execution

- CRRN1: the returned timer identifier shall be between 01 and 08 inclusive.
- CRRN2: the returned timer identifier shall be different from a previously allocated but not released one.
- CRRN3: The SIM Toolkit Framework shall trigger the applet when receiving an ENVELOPE(TIMER EXPIRATION) command for the allocated timer.
- CRRN4: A call to isEventSet() method for EVENT_TIMER_EXPIRATION should return true if the applet has at least one timer allocated.

6.2.9.1.1.2 Parameters error

No requirements.

6.2.9.1.1.3 Context errors

- CRRC1: Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE if all the timers are allocated.
- CRRC2: Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE if the maximum number of timers have been allocated to this applet according to installation parameter.

6.2.9.1.2 Test suite files

Test Script:	API_2_TKR_ATIM_1.scr
Test Applet:	API_2_TKR_ATIM_1.java
	API_2_TKR_ATIM_2.java
	API_2_TKR_ATIM_3.java

- Installation parameters:
- For this test procedure the non-volatile memory of each instance is 200 (Hexa).
- The maximum timer parameter value is as follows for each applet:
  - applet-1 (API_2_TKR_ATIM_1): 8 timers
  - applet-2 (API_2_TKR_ATIM_2): 4 timers
  - applet-3 (API_2_TKR_ATIM_3): 0 timer

Load Script: API_2_TKR_ATIM_1.ldr

- The load script installs the 6 instances.

Cleanup Script: API_2_TKR_ATIM_1.clr

Parameter File: API_2_TKR_ATIM_1.par

6.2.9.1.3 Test procedure

Id	Description	API Expectation	APDU Expectation
1	**Allocates up to 8 timers (applet-1)** `8 * allocateTimer().`	No exception shall be thrown. Timer ID returned shall be between 01 and 08 inclusive. It shall be different after each call.	
2	**Allocate timers more than the maximum (applet-1)** `The applet-1 allocates 1 more timer.`	Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE.	

3	**Check applet is Triggered by ENVELOPE(TIMER_EXPIRATION) command (applet1)**
	`Send ENVELOPE(TIMER EXPIRATION) with all timers id (not in an increase order).`
	`Calls releaseTimer(id) each time a timer expires.`
		Shall trigger each time an ENVELOPE(TIMER EXPIRATION) is sent to the SIM, for Timer ID = '01' to '08'.	

4	**Allocate up to 4 timers (applet-2)** `4 * allocateTimer().`	No exception shall be thrown. Each time, the returned timer identifier shall be between '01' and '08' inclusive. It shall be different after each call.	
5	**Allocate timers more than the maximum (~~applet 3~~applet3)** The ~~applet 3~~applet3 allocates 1 more timer.	Shall throw a ToolkitException with reason NO_TIMER_AVAILABLE.	

6.2.9.1.4 Test Coverage

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

6.2.9.12 Method setEvent

Test Area Reference: API_2_TKR_SEVTB

6.2.9.12.1 Conformance Requirement:

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

```
public void setEvent(byte id)
        throws ToolkitException,
               javacard.framework.TransactionException
```

6.2.9.12.1.1 Normal execution

- CRRN1: a following call to isEventSet() method with the same event id shall answer true for the applet.
- CRRN2: the SIM Toolkit Framework shall trigger the applet if an occurrence of the set event happens.
- CRRN3: the method shall accept all the events defined in 3GPP TS 43.019 [7] except: EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION , EVENT_STATUS_COMMAND
- CRRN4: no exception shall be thrown if the applet registers more than once to the same event.
- CRRN5: all updates in the ToolkitRegistry are atomic.

6.2.9.12.1.2 Parameters error

- CRRP1: shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason if event is 0.
- CRRP2: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if event is EVENT_MENU_SELECTION.
- CRRP3: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if event is EVENT_MENU_SELECTION_HELP_REQUEST.
- CRRP4: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if event is EVENT_TIMER_EXPIRATION.
- CRRP5: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if event is EVENT_STATUS_COMMAND.

6.2.9.12.1.3 Context errors

- CRRC1: shall throw a ToolkitException with EVENT_ALREADY_REGISTERED if event is EVENT_CALL_CONTROL_BY_SIM but another applet is already registered to it.
- CRRC2: shall throw a ToolkitException with EVENT_ALREADY_REGISTERED if event is EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM but another applet is already registered to it.
- CRRC3: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_PP_ENV and the applet has no TAR defined.
- CRRC4: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_PP_UPD and the applet has no TAR defined.
- CRRC5: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_CB_ENV and the applet has no TAR defined.
- CRRC6: shall throw javacard.framework.TransactionException - if the operation would cause the commit capacity to be exceeded.

6.2.9.12.2 Test suite files

Test Script:	API_2_TKR_SEVTB_1.scr
Test Applet:	API_2_TKR_SEVTB_1.java
	API_2_TKR_SEVTB_2.java
	API_2_TKR_SEVTB_3.java
	API_2_TKR_SEVTB_4.java
Load Script:	API_2_TKR_SEVTB_1.ldr
	The load script installs the 4 instances.
Cleanup script:	API_2_TKR_SEVTB_1.clr

Parameter File: API_2_TKR_SEVTB_1.par

6.2.9.12.3 Test Procedure

Id	Description	API Expectation	APDU Expectation
1	**Applet-1 is triggered by ENVELOPE(SMS_PP_FORMATTED) command.** `Send ENVELOPE(SMS_PP_FORMATTED)`	Applet-1 shall be triggered	

2	**Setting ALLOWED and SUPPORTED events**
	`1- For all allowed events (-1, 1 to 24 and 127 excepted 7, 8, 11, 19) defined in TS 43.019 [7]*: EVENT_PROFILE_DOWNLOAD, EVENT_FORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_PP_UPD, EVENT_FORMATTED_SMS_CB, EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_CB, EVENT_CALL_CONTROL_BY_SIM, EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM, EVENT_EVENT_DOWNLOAD_MT_CALL, EVENT_EVENT_DOWNLOAD_CALL_CONNECTED, EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED, EVENT_EVENT_DOWNLOAD_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_USER_ACTIVITY, EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE, EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS, EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION, EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION, EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS, EVENT_FIRST_COMMAND_AFTER_SELECT, EVENT_UNRECOGNIZED_ENVELOPE`
	`1.1- clearEvent(event)`
	`1.2- isEventSet(event)`
	`1.3- setEvent(event)`
	`1.4- isEventSet(event)`
	`1.5- clearEvent(event)`
		1.1- No exception shall be thrown.
		1.2- Shall return false.
		1.3- No exception shall be thrown.
		1.4- Shall return true.
		1.5- No exception shall be thrown.	

3	**Event 0** `Call setEvent(0)`	Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason code.	
4	**Setting EVENT_MENU_SELECTION** `Call setEvent(EVENT_MENU_SELECTION)`	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
5	**Setting EVENT_MENU_SELECTION_HELP_REQUEST** `Call setEvent(EVENT_MENU_SELECTION_HELP_REQUEST)`	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
6	**Setting EVENT_TIMER_EXPIRATION** `Call setEvent(EVENT_TIMER_EXPIRATION)`	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
7	**Setting EVENT_STATUS_COMMAND** `Call setEvent(EVENT_STATUS_COMMAND)`	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason	

Id	Description	API Expectation	APDU Expectation
		code.	
8	**Setting EVENT_CALL_CONTROL_BY_SIM** `Call setEvent(EVENT_CALL_CONTROL_BY_SIM)`	No Exception shall be thrown	
9	**Setting EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM** `Call setEvent(EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM)`	No Exception shall be thrown	
10	**Check applet is triggered by an ENVELOPE(CALL_CONTROL_BY_SIM)** Trigger the applet	Applet is trigged by an ENVELOPE(CALL_CONTROL_BY_SIM)	
11	**Check applet is triggered by an ENVELOPE(MO_SHORT_MESSAGE_CONTROL_BY_SIM)** Trigger the Applet	Applet is trigged by an ENVELOPE(MO_SHORT_MESSAGE_CONTROL_BY_SIM)	
12	**Applet~~-~~2 is triggered by ENVELOPE(SMS_ PP_DOWNLOAD) command.** `Trigger the ~~a~~Applet~~–~~2`	Applet~~-~~2 is trigged by an ENVELOPE(SMS_ PP_DOWNLOAD) command	
13	**Applet~~-~~2 registers to CALL_CONTROL_BY_SIM but it is already assigned** `SetEvent(EVENT_CALL_CONTROL_BY_SIM)`	Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	
14	**Applet~~-~~2 registers to MO_MESSAGE_CONTROL_BY SIM but it is already assigned** `setEvent(EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM)`	Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	

15	**~~Applet 3~~Applet3 with no TAR defined registers to EVENT_UNFORMATTED_SMS_CB**
	`1- send ENVELOPE(CELL_BROADCAST_DATA_DOWNLOAD)`
	`2- setEvent(FORMATTED_SMS_PP_ENV)`
	`3- setEvent(FORMATTED_SMS_PP_UPD)`
	`4- setEvent(FORMATTED_SMS_CB_ENV)`
		1- ~~Applet 3~~Applet3 shall be triggered
		2- ToolkitException with reason code TAR_NOT_DEFINED should be thrown
		3- ToolkitException with reason code TAR_NOT_DEFINED should be thrown
		ToolkitException with reason code TAR_NOT_DEFINED should be thrown	

16	**~~Applet 4~~Applet4 registers multiple to EVENT_FORMATTED_SMS_PP_ENV**
	`1- send ENVELOPE(EVENT_FORMATTED_SMS_PP_ENV)`
	`2- setEvent(EVENT_FORMATTED_SMS_PP_ UPD)`
	`3- setEvent(EVENT_FORMATTED_SMS_PP_ UPD)`
	`4- send ENVELOPE(EVENT_FORMATTED_SMS_PP_UPD)`
		1- ~~Applet 4~~Applet4 shall be triggered
		2- no Exception shall be thrown
		3- no Exception shall be thrown
		4- ~~Applet 4~~Applet4 shall be triggered	

NOTE: Although the method setEvent is defined for a range from –128 to 127 only the allowed events are tested, because the range from -128 to –2 is reserved for propriatary use in TS TS 43.019 [7] chapter 6.2 and the range from 25 to 126 is omitted for compatibility with future releases of TS 43.019 [7]

6.2.9.12.4 Test Coverage

CRR number	Test case number
N1	2
N2	1,8,9,10,11,12
N3	2,4,5,6,7
N4	16
N5	not testable
P1	3
P2	4
P3	5
P4	6
P5	7
C1	13
C2	14
C3	15
C4	15
C5	15
C6	not testable

6.2.9.13 Method setEventList

Test Area Reference: API_2_TKR_SEVL_BSS

6.2.9.13.1 Conformance Requirement:

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

```
public void setEventList(byte[] eventList,
                         short offset,
                         short length)
                throws   java.lang.NullPointerException,
                         java.lang.ArrayIndexOutOfBoundsException,
                         ToolkitException,
                         javacard.framework.TransactionException
```

6.2.9.13.1.1 Normal execution

- CRRN1: for all events set successfully by this method, a call to isEventSet() method should return true.
- CRRN2: the SIM Toolkit Framework shall trigger the applet if an occurrence of one of the successfully registered events happens.
- CRRN3: this method shall accept all the events defined in 3GPP TS 43.019 [7] except: EVENT_MENU_SELECTION, EVENT_MENU_SELECTION_HELP_REQUEST, EVENT_TIMER_EXPIRATION , EVENT_STATUS_COMMAND.
- CRRN4: all updates on the ToolkitRegistry are atomic
- CRRN5: No exception shall be thrown if the applet registers more than once to the same event.

6.2.9.13.1.2 Parameters error

- CRRP1: shall throw a java.lang.NullPointerException if eventList is null.
- CRRP2: shall throw a java.lang.ArrayIndexOutOfBoundsException if offset would cause access outside array bounds.
- CRRP3: shall throw a java.lang.ArrayIndexOutOfBoundsException if length would cause access outside array bounds.
- CRRP4: shall throw a java.lang.ArrayIndexOutOfBoundsException if both offset and length would cause access outside array bounds.

- CRRP5: shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason if event is 0.
- CRRP6: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if eventList contains EVENT_MENU_SELECTION.
- CRRP7: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if eventList contains EVENT_MENU_SELECTION_HELP_REQUEST.
- CRRP8: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if eventList contains EVENT_TIMER_EXPIRATION.
- CRRP9: shall throw a ToolkitException with EVENT_NOT_ALLOWED reason if eventList contains EVENT_STATUS_COMMAND.

6.2.9.13.1.3 Context errors

- CRRC1: shall throw a ToolkitException with EVENT_ALREADY_REGISTERED if eventList contains EVENT_CALL_CONTROL_BY_SIM but another applet is already registered to it.
- CRRC2: shall throw a ToolkitException with EVENT_ALREADY_REGISTERED if eventList contains EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM but another applet is already registered to it.
- CRRC3: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_PP_ENV and the applet has no TAR defined.
- CRRC4: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_PP_UPD and the applet has no TAR defined.
- CRRC5: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_CB_ENV and the applet has no TAR defined.
- CRRC6: shall throw javacard.framework.TransactionException - if the operation would cause the commit capacity to be exceeded.

6.2.9.13.2 Test suite files

Test Script:	API_2_TKR_SEVL_BSS_1.scr
Test Applet:	API_2_TKR_SEVL_BSS_1.java
	API_2_TKR_SEVL_BSS_2.java
	API_2_TKR_SEVL_BSS_3.java
Load Script:	API_2_TKR_SEVL_BSS_1.ldr
	The load script installs the 4 instances.
Cleanup script:	API_2_TKR_SEVL_BSS_1.clr
Parameter File:	API_2_TKR_SEVL_BSS_1.par

6.2.9.13.3 Test Procedure

Id	Description	API Expectation	APDU Expectation

1	**Applet-1 Registering all eventList buffer**
	`EventList = all allowed events (-1, 1 to 24 and 127 excepted 7, 8, 11, 19) defined in TS 43.019[7]:`
	`EVENT_PROFILE_DOWNLOAD,`
	`EVENT_FORMATTED_SMS_PP_ENV,`
	`EVENT_FORMATTED_SMS_PP_UPD,`
	`EVENT_FORMATTED_SMS_CB,`
	`EVENT_UNFORMATTED_SMS_PP_ENV,`
	`EVENT_UNFORMATTED_SMS_PP_UPD,`
	`EVENT_UNFORMATTED_SMS_CB,`
	`EVENT_CALL_CONTROL_BY_SIM,`
	`EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM,`
	`EVENT_EVENT_DOWNLOAD_MT_CALL,`
	`EVENT_EVENT_DOWNLOAD_CALL_CONNECTED,`
	`EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED,`
	`EVENT_EVENT_DOWNLOAD_LOCATION_STATUS,`
	`EVENT_EVENT_DOWNLOAD_USER_ACTIVITY,`
	`EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE,`
	`EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS,`
	`EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION,`
	`EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION,`
	`EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE,`
	`EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS,`
	`EVENT_FIRST_COMMAND_AFTER_SELECT,`
	`EVENT_UNRECOGNIZED_ENVELOPE`
	`1- For each event in EventList clearEvent(event)`
	`2- setEventList(eventList)`
	`Offset = 0`
	`Length = eventList.lentgh`
	`3- For all events in eventList isEventSet(event)`
	`4- For each event in EventList clearEvent(event)`
		1- No exception shall be thrown.
		2- No exception shall be thrown.
		3- Each time shall return true.
		4- No exception shall be thrown.	

2	**Registering part of eventList buffer**
	`EventList = all allowed events defined in TS 43.019[7] (see test case 1).`
	`1- For each event in EventList clearEvent(event)`
	`2- setEventList(eventList, offset, length)`
	`Offset > 0`
	`Length = eventList.lentgh – offset`
	`3- For all events in eventList:`
	`isEventSet(event)`
	`4- For each event in EventList: clearEvent(event)`
		1- No exception shall be thrown.
		2- No exception shall be thrown.
		3- Each time shall return true for events ranging from offset to offset+length else shall return false.
		4- No exception shall be thrown.	

3	**Null buffer** `EventList = null`	Shall throw a java.lang.NullPointerException Exception	

4	**Out of bounds offset**
	`Offset = eventList.length`
	`Length = 1`
		Shall throw a java.lang.ArrayIndexOutOfBounds Exception	

Id	Description	API Expectation	APDU Expectation

5	**Out of bounds and big offset**
	`Offset = 255`
	`Length = 1`
		Shall throw a java.lang.ArrayIndexOutOfBounds Exception	

6	**Offset < 0**
	`Offset = -1`
	`Length = 1`
		Shall throw a java.lang.ArrayIndexOutOfBounds Exception	

7	**Out of bounds length**
	`Offset = 0`
	`Length = eventList.length + 1`
		Shall throw a java.lang.ArrayIndexOutOfBounds Exception	

8	**Out of bounds and big length**
	`Offset = 0`
	`Length = 255`
		Shall throw a java.lang.ArrayIndexOutOfBounds Exception	

9	**Length < 0**
	`Offset = 0`
	`Length = -1`
		Shall throw a java.lang.ArrayIndexOutOfBounds Exception	

10	**Out of bounds offset + Length** `Offset + length > eventList.length + 1`	Shall throw a java.lang.ArrayIndexOutOfBounds Exception	
11	**Event 0** `Call setEventList(eventList) with eventList indicating event 0`	Shall throw a ToolkitException with EVENT_NOT_SUPPORTED reason code.	
12	**EVENT_MENU_SELECTION** `Call setEventList(eventList) with eventList indicating EVENT_MENU_SELECTION`	Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	
13	**EVENT_MENU_SELECTION_HELP_REQUEST** `Call setEventList(eventList) with eventList indicating EVENT_MENU_SELECTION_HELP_REQUEST`	Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	
14	**EVENT_TIMER_EXPIRATION** `Call setEventList(eventList) with eventList indicating EVENT_TIMER_EXPIRATION`	Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	
15	**EVENT_STATUS_COMMAND** `Call setEventList(eventList) with eventList indicating EVENT_STATUS_COMMAND`	Shall throw a ToolkitException with reason code EVENT_NOT_ALLOWED.	
16	**Setting EVENT_CALL_CONTROL_BY_SIM** `setEventList(List, 0, 2) with List containing EVENT_CALL_CONTROL_BY_SIM & EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM`	Shall not throw an exception	

17	**Check applet is triggered by an ENVELOPE(CALL_CONTROL_BY_SIM)**
	`Reset and initialise the card`
	`Trigger the applet`
		Applet is trigged by an ENVELOPE(CALL_CONTROL_BY_SIM)	

18	**Check applet is triggered by an ENVELOPE(MO_SHORT_MESSAGE_CONTROL_BY_SIM)** `Trigger the applet`	Applet is trigged by an ENVELOPE(MO_SHORT_MESSAGE_CONTROL_BY_SIM)	

Id	Description	API Expectation	APDU Expectation
19	**Applet~~-~~2 registers to CALL_CONTROL_BY_SIM but it is already assigned** setEventList(MonoEventList,0,1) with MonoEventList containing EVENT_CALL_CONTROL_BY_SIM	Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	
20	**Applet~~-~~2 registers to MO_SHORT_MESSAGE_CONTROL_BY_SIM but it is already assigned setEventList(MonoEventList,0,1) with MonoEventList containing EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM**	Shall throw a ToolkitException with EVENT_ALREADY_REGISTERED reason code.	
21	**~~Applet 3~~Applet3 with no TAR defined registers to EVENT_UNFORMATTED_SMS_CB** 1- send ENVELOPE(EVENT_UNFORMATTED_SMS_CB) 2- setEventList(EVENT_FORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_ENV) 3- setEventList(EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_PP_ENV) 4- setEventList(EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_FORMATTED_SMS_CB_ENV) 5- isEventSet(EVENT_UNFORMATTED_SMS_PP_ENV) 6- isEventSet(EVENT_UNFORMATTED_SMS_PP_UPD) 7- isEventSet(EVENT_FORMATTED_SMS_PP_ENV) 8- isEventSet(EVENT_FORMATTED_SMS_PP_UPD) 9- isEventSet(EVENT_FORMATTED_SMS_CB_ENV)	1- Applet3 shall be triggered 2- ToolkitException with reason code TAR_NOT_DEFINED should be thrown 3- ToolkitException with reason code TAR_NOT_DEFINED should be thrown 4- ToolkitException with reason code TAR_NOT_DEFINED should be thrown 5- method should return FALSE 6- method should return FALSE 7- method should return FALSE 8- method should return FALSE 9- method should return FALSE	
22	1- setEventList(EVENT_UNFORMATTED_SMS_PP_ENV, EVENT_UNFORMATTED_SMS_PP_ENV) 2- isEventSet(EVENT_UNFORMATTED_SMS_PP_ENV)	1- no exception should be thrown 2- method should return true	

6.2.9.13.4 Test Coverage

CRR number	Test case number
N1	1,2
N2	16,17,18
N3	1,2,11,12,13,14,15
N4	21
N5	22
P1	3
P2	4,5,6
P3	7,8,9
P4	10
P5	11
P6	12

P7	13
P8	14
P9	15
C1	19
C2	20
C3	21
C4	21
C5	21
C6	not testable

6.3 SIM Toolkit Framework

6.3.2 Handler Integrity

6.3.2.2 ProactiveResponseHandler

Test Area Reference: FWK_HIN_-PRHD

6.3.2.2.1 Conformance Requirement

6.3.2.2.1.1 Normal Execution

- CRRN1: The ProactiveResponseHandler content is changed after the call to ProactiveHandler.send method and remains unchanged until next call to the ProactiveHandler.send method.
- CRRN2: The ProactiveResponseHandler may not be available before the first call to ProactiveHandler.send method, if available the content is cleared.

6.3.2.2.1.2 Parameters error

No requirements.

6.3.2.2.1.3 Context Errors

No requirements.

6.3.2.2.2 Test Suite Files

Test Script:	FWK_HIN_PRHD_1.scr
Test Applet:	FWK_HIN_PRHD_1.java
Load Script:	FWK_HIN_PRHD_1.ldr
Cleanup Script:	FWK_HIN_PRHD_1.clr
Parameter File:	FWK_HIN_PRHD_1.par

6.3.2.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration and ProactiveResponseHandler obtaining** 1-Applet is registered to all events defined in [7]. Using the methods initMenuEntry for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION and setEventList() for the rest of the events. Terminal Profile command is sent to the SIM without the facilities of SET_EVENT_LIST ,SETUP_IDLE_MODE_TEXT, SETUP_MENU and POLL_INTERVAL. For each event: 2-ProactiveResponseHandler.getTheHandler() is called If handler is available, ProactiveResponseHandler.getLength() is called	1- No exception is thrown 2- Applet is triggered. 3- Behaviour 1: Toolkit Exception HANDLER_NOT_AVAILABLE is thrown. Behaviour 2: No exception is thrown, the return value is 0	
2	**The ProactiveResponseHandler remains unchanged after send method invocation until next send method invocation** 1-Applet builds a proactive command ProactiveHandler.send() method is called 2-ProactiveResponseHandler.getLength() method is called 3-ProactiveHandler.init() method is called 4-ProactiveHandler.send() method is called 5-ProactiveResponseHandler.getLength() method is called	1- The ProactiveResponseHandler contains the terminal response 3- The return value is 12 4- No exception is thrown and the Proactive Response Handler remains unchanged 5- The ProactiveResponseHandler contains the terminal response of the second proactive command 7- The return value is 15	2- A proactive command is fetched The terminal response is sent with length 12 6- A proactive command is fetched The terminal response is sent with length 15

6.3.2.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2
CRRN2	1

6.3.2.3 EnvelopeHandler

Test Area Reference: FWK_HIN_ENHD

6.3.2.3.1 Conformance Requirement

6.3.2.3.1.1 Normal Execution

- CRRN1: The EnvelopeHandler and its content are available for all triggered toolkit applets, from the invocation to the termination of their processToolkit method
- CRRN2: The SIM Toolkit Framework guarantees that all triggered toolkit applets receive the data.
- CRRN3: The SIM Toolkit Framework shall convert the Update Record EFsms in the EnvelopeHandler TLV List containing Device Identities TLV, Address TLV and SMS TPDU TLV.
- CRRN4: The getEnvelopeTag() method shall return *BTAG_SMS_PP_DOWNLOAD*.
- CRRN5: The getLength() method shall return the Simple TLV list length.
- CRRN6 The Device Identity Simple TLV is used to store the information about the absolute record number in the EFsms file and the value of the EFsms record status byte.

6.3.2.3.1.2 Parameters error

No requirements.

6.3.2.3.1.3 Context Errors

No requirements.

6.3.2.3.2 Test Suite Files

Test Script:	FWK_HIN_ENHD_1.scr
Test Applet:	FWK_HIN_ENHD_1.java
Load Script:	FWK_HIN_ENHD_1.ldr
Cleanup Script:	FWK_HIN_ENHD_1.clr
Parameter File:	FWK_HIN_ENHD_1.par

6.3.2.3.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet initialization and Envelope Handler integrity checks with EVENT_MENU_SELECTION_HELP_REQUEST**		
	1- Applet is registered to all events defined in TS 43.019 [7] except EVENT_PROFILE_DOWNLOAD and EVENT_STATUS_COMMAND. Using the methods initMenuEntry() for EVENT_MENU_SELECTION, allocateTimer()for EVENT_TIMER_EXPIRATION, and setEventList() for the rest of the events. Perform SIM initialization with all the facilities supported	1-No exception is thrown	
	2-Envelope menu selection with help request is sent to the SIM	2- Applet is triggered	
	3-EnvelopeHandler.getTheHandler() method is called	3- No exception is thrown.	
	4-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	4- No exception is thrown	
	The EnvelopeHandler.findTLV() method is called with TAG_HELP_REQUEST		
	5-A proactive command DISPLAY TEXT is sent		5- 91 xx.
	6-Envelope call control by sim is sent to SIM	6- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	7- It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	7- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			A proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	Check that the TAG_HELP_REQUEST is the TLV selected		
	8-The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	8- The contents of the envelope handler shall be the same as stored in buffer 1	
2	**Envelope Handler integrity checks with EVENT_MENU_SELECTION**		
	1-An envelope menu selection is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	

Id	Description	API/Framework Expectation	APDU Expectation
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_ITEM_IDENTIFIER		
	4-A proactive command DISPLAY TEXT is sent		
	5-Envelope call control by sim is sent to SIM		4- 91 XX
		5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6- It's checked the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
	It's checked that the TAG_ITEM_IDENTIFIER is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()		

Id	Description	API/Framework Expectation	APDU Expectation
3	**Envelope Handler integrity checks with EVENT_FORMATTED_SMS_PP_ENV**		
	`1-A formatted sms pp envelope is sent to SIM`	1- Applet is triggered	
	`2-EnvelopeHandler.getTheHandler() method is called`	2- No exception is thrown.	
	`3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy( )`	3- No exception is thrown.	
	`The EnvelopeHandler.findTLV() method is called with TAG_SMS_TPDU`		
	`4-A proactive command DISPLAY TEXT is sent`		4- 91 XX
	`5-Envelope call control by sim is sent to SIM`	5- Applet is triggered	
	`EnvelopeHandler.getTheHandler() method is called`		
	`6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy( ) and Util.arrayCompare methods`	6- No exception is thrown and the handler contains the envelope call control by SIM	
	`The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES`		
	`Call Control execution is finished.`		
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
	`It’s checked that the TAG_SMS_TPDU is the TLV selected`		

Id	Description	API/Framework Expectation	APDU Expectation
4	**Envelope Handler integrity checks with EVENT_UNFORMATTED_SMS_PP_ENV**		
	`1-A unformatted sms pp envelope is sent to SIM`	1- Applet is triggered	
	`2-EnvelopeHandler.getTheHandler() method is called`	2- No exception is thrown.	
	`3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy( )`	3- No exception is thrown.	
	`The EnvelopeHandler.findTLV method is called with TAG_DEVICE_IDENTITIES`		
	`4-A proactive command DISPLAY TEXT is sent`		4- 91 XX
	`5-Envelope call control by sim is sent to SIM`	5- Applet is triggered	
	`EnvelopeHandler.getTheHandler() method is called`		
	`6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods`	6- No exception is thrown and the handler contains the envelope call control by SIM	
	`The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES`		
	`Call Control execution is finished.`		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	`It’s checked that the TAG_DEVICE_IDENTITIES is the TLV selected`		
	`7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()`	7- The contents of the envelope handler shall be the same as stored in buffer 1.	

Id	Description	API/Framework Expectation	APDU Expectation
5	**Envelope Handler integrity checks with EVENT_UNFORMATTED_SMS_CB** 1-A unformatted cellbroadcast envelope is sent to SIM 2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_CELLBROADCAST_PAGE 4-A proactive command DISPLAY TEXT is sent 5-Envelope call control by sim is sent to SIM EnvelopeHandler.getTheHandler() method is called 6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy and Util.arrayCompare() methods The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES Call Control execution is finished. It’s checked that the TAG_CELLBROADCAST_PAGE is the TLV selected	1- Applet is triggered 2- No exception is thrown. 3- No exception is thrown. 5- Applet is triggered 6- No exception is thrown and the handler contains the envelope call control by SIM	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
6	**Envelope Handler integrity checks with EVENT_TIMER_EXPIRATION** 1-A timer expiration envelope is sent to SIM 2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_TIMER_ID 4-A proactive command DISPLAY TEXT is sent 5-Envelope call control by sim is sent to SIM	1- Applet is triggered 2- No exception is thrown. 3- No exception is thrown. 5- Applet is triggered	4- 91 XX

Id	Description	API/Framework Expectation	APDU Expectation
	EnvelopeHandler.getTheHandler() method is called		
	6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It’s checked that the TAG_TIMER_ID is the TLV selected		

Id	Description	API/Framework Expectation	APDU Expectation
7	**Envelope Handler integrity checks with EVENT_CALL_CONTROL_BY_SIM**		
	1-A call control envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_ADDRESS		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It’s checked that the TAG_ADDRESS is the TLV selected		
8	**Envelope Handler integrity checks with EVENT_ MO_SHORT_MESSAGE_CONTROL_BY_SIM**		
	1-A mo short message control by sim envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_ADDRESS		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		

Id	Description	API/Framework Expectation	APDU Expectation
	6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES Call Control execution is finished. It's checked that the TAG_ADDRESS is the TLV selected	6- No exception is thrown and the handler contains the envelope call control by SIM	Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
9	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_MT_CALL** 1-A event download mt call envelope is sent to SIM 2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_ADDRESS 4-A proactive command DISPLAY TEXT is sent 5-Envelope call control by sim is sent to SIM EnvelopeHandler.getTheHandler() method is called 6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES Call Control execution is finished. It’s checked that the TAG_ADDRESS is the TLV selected	1- Applet is triggered 2- No exception is thrown. 3- No exception is thrown. 5- Applet is triggered 6- No exception is thrown and the handler contains the envelope call control by SIM	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM

Id	Description	API/Framework Expectation	APDU Expectation
10	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_CALL_CONNECTED**		
	1-A event download call connected envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_ADDRESS		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It’s checked that the TAG_ADDRESS is the TLV selected		

Id	Description	API/Framework Expectation	APDU Expectation
11	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_CALL_DISCONNECTED**		
	1-A event download call disconnected envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_ADDRESS		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
	It’s checked that the TAG_ADDRESS is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1.	
12	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_LOCATION_STATUS**		
	1-A event download location status envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_LOCATION_STATUS		
	4-A proactive command DISPLAY TEXT is sent		4-91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		

Id	Description	API/Framework Expectation	APDU Expectation
	6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES Call Control execution is finished. It’s checked that the TAG_LOCATION_STATUS is the TLV selected	6- No exception is thrown and the handler contains the envelope call control by SIM	Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
13	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_USER_ACTIVITY** 1-A event download user activity envelope is sent to SIM 2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES 4-A proactive command DISPLAY TEXT is sent 5-Envelope call control by sim is sent to SIM EnvelopeHandler.getTheHandler() method is called 6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES Call Control execution is finished. It's checked that the TAG_DEVICE_IDENTITIES is the TLV selected	1- Applet is triggered 2- No exception is thrown. 3- No exception is thrown. 5- Applet is triggered 6- No exception is thrown and the handler contains the envelope call control by SIM	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM

Id	Description	API/Framework Expectation	APDU Expectation
14	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE**		
	1-A event download idle screen available envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	4-A proactive command DISPLAY TEXT is sent		
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	4- 91 XX
	EnvelopeHandler.getTheHandler() method is called		
	6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It’s checked that the TAG_DEVICE_IDENTITIES is the TLV selected		

Id	Description	API/Framework Expectation	APDU Expectation
15	**Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_CARD_READER_STATUS**		
	1-A event download card reader status envelope is sent to SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_CARD_READER_STATUS		
	4-A proactive command DISPLAY TEXT is sent		
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	4- 91 XX
	EnvelopeHandler.getTheHandler() method is called		
	6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy and Util.arrayCompare() methods		
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES	6- No exception is thrown and the handler contains the envelope call control by SIM	
			Proactive command Display Text is fetched
	It’s checked that the TAG_CARD_READER_STATUS is the TLV selected		The terminal Response of DISPLAY TEXT is sent to the SIM

Id	Description	API/Framework Expectation	APDU Expectation
16	**Envelope Handler integrity checks with UNRECOGNIZED_ENVELOPE** 1-A unrecognized envelope is sent to SIM 2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() 4-A proactive command DISPLAY TEXT is sent 5-Envelope call control by sim is sent to SIM EnvelopeHandler.getTheHandler() method is called The EnvelopeHandler.getValueLength() is called 6-It's checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES Call Control execution is finished.	1- Applet is triggered 2- No exception is thrown. 3- No exception is thrown. 5- Applet is triggered 6- No exception is thrown and the handler contains the envelope call control by SIM	4- 91 XX Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM

Id	Description	API/Framework Expectation	APDU Expectation
17	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION**		

Id	Description	API/Framework Expectation	APDU Expectation
18	**Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION**		

Id	Description	API/Framework Expectation	APDU Expectation
~~19~~	~~Envelope Handler integrity checks with UNRECOGNIZED_ENVELOPE~~		

Id	Description	API/Framework Expectation	APDU Expectation
19~~20~~	**Envelope Handler integrity checks with EVENT_FORMATTED_SMS_PP_UPD**		
	1-Update Record EFsms instruction single and formatted is sent to the SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy( ) The EnvelopeHandler.findTLV() method is called with TAG_SMS_TPDU	3- No exception is thrown.	
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM EnvelopeHandler.getTheHandler() method is called	5- Applet is triggered	
	6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy( ) and Util.arrayCompare methods The EnvelopeHandler.findTLV() method is called with TAG_SMS_TPDU Call Control execution is finished.	6- No exception is thrown and the handler contains the envelope call control by SIM	
			Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
	It’s checked that the TAG_SMS_TPDU is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	

Id	Description	API/Framework Expectation	APDU Expectation
2~~1~~0	**Envelope Handler integrity checks with EVENT_UNFORMATTED_SMS_PP_UPD**		
	1- Update Record EFsms instruction single and unformatted is sent to the SIM	1- Applet is triggered	
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy( )	3- No exception is thrown.	
	The EnvelopeHandler.findTLV method is called with TAG_SMS_TPDU		
	4-A proactive command DISPLAY TEXT is sent		4- 91 XX
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		
	6-It’s checked that the contents of the envelope handler is the envelope call control using EnvelopeHandler.copy() and Util.arrayCompare() methods	6- No exception is thrown and the handler contains the envelope call control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It’s checked that the TAG_DEVICE_IDENTITIES is the TLV selected		
	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1.	

Id	Description	API/Framework Expectation	APDU Expectation
2~~2~~1	**Check the TLV list conversion for EVENT_FORMATTED_SMS_PP_UPD**		
	1- An EVENT_FORMATTED_SMS_PP_UPD is sent to the SIM.	1- Applet is triggered	
	2- The findTLV(tag == device identities Tag) is called.	2- No exception is thrown.	
	3- The getValueByte(offset == 0) is called.	3- return the absolute record.	
	4- The getValueByte(offset == 1) is called.	4- return the record status	
	5- The findTLV(tag == address Tag) is called.	5- No exception is thrown.	
	6- Check the content		
	7- The findTLV(tag == SMS TPDU Tag) is called.	7- No exception is thrown.	
	8- Check the content		

Id	Description	API/Framework Expectation	APDU Expectation
22~~3~~	**Check TLV list conversion for EVENT_FORMATTED_SMS_PP_UPD** `1- The getLength() method is called`	1. return the Simple TLV list length	

Id	Description	API/Framework Expectation	APDU Expectation
243	**Check TLV list conversion for EVENT_FORMATTED_SMS_PP_UPD** `1- The getEnvelopeTag() method is called`	1- return *BTAG_SMS_PP_DOWNLOAD*	

Id	Description	API/Framework Expectation	APDU Expectation
24~~5~~	**Check the TLV list conversion for EVENT_UNFORMATTED_SMS_PP_UPD**		
	`1- An EVENT_UNFORMATTED_SMS_PP_UPD is sent to the SIM.`	1- Applet is triggered	
	`2- The findTLV(tag == device identities Tag) is called.`	2- No exception is thrown.	
	`3- The getValueByte(offset == 0) is called.`	3- return the absolute record.	
	`4- The getValueByte(offset == 1) is called.`	4- return the record status	
	`5- The findTLV(tag == address Tag) is called.`	5- No exception is thrown.	
	`6- Check the content`		
	`7- The findTLV(tag == SMS TPDU Tag) is called.`	7- No exception is thrown.	
	`8- Check the content`		

Id	Description	API/Framework Expectation	APDU Expectation
2~~6~~5	**Check TLV list conversion for EVENT_UNFORMATTED_SMS_PP_UPD** `1- The getLength() method is called`	1. return the Simple TLV list length	

Id	Description	API/Framework Expectation	APDU Expectation
26~~7~~	**Check TLV list conversion for EVENT_UNFORMATTED_SMS_PP_UPD** `1- The getEnvelopeTag() method is called`	1- return *BTAG_SMS_PP_DOWNLOAD*	

6.3.2.3.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20~~, 21~~
CRRN2	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20~~, 21~~
CRRN3	~~22~~21, ~~25~~24
CRRN4	22~~3~~, ~~26~~25
CRRN5	~~24~~23, ~~27~~26
CRRN6	~~22~~21, ~~25~~24

6.3.3 Applet Triggering

6.3.3.6 EVENT_CALL_CONTROL_BY_SIM

Test Area Reference: FWK_APT_ECCN

6.3.3.6.1 Conformance Requirement

6.3.3.6.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_CALL_CONTROL_BY_SIM once it has registered to this event and an Envelope Call Control is received.
- CRRN2: The applet is not triggered by the EVENT_CALL_CONTROL_BY_SIM once it has deregistered from this event.

6.3.3.6.1.2 Parameters error

No requirements.

6.3.3.6.1.3 Context Errors

No requirements.

6.3.3.6.2 Test Suite Files

Test Script:	FWK_APT_ECCN_1.scr
Test Applet:	FWK_APT_ECCN_1.java
Load Script:	FWK_APT_ECCN_1.ldr
Cleanup Script:	FWK_APT_ECCN_1.clr
Parameter File:	FWK_APT_ECCN_1.par

6.3.3.6.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation

1	**Applets registration to EVENT_CALL_CONTROL_BY_SIM and triggering**
	`Applet1 is registered to EVENT_CALL_CONTROL_BY_SIM.`
	`Applet2 is registered to EVENT_FORMATTED_SMS_PP_ENV`
	`1-An Envelope Call control by SIM is sent to SIM`
		1- Applet1 is triggered	

2	**Applet deregistration and registration of the third applet to EVENT_CALL-CONTROL_BY_SIM.**
	`1-An Envelope Formatted SMS PP envelope is sent to SIM`
	`Applet2 contructs a DISPLAY TEXT proactive command.`
	`2-ProactiveHandler.send() method is called`
	`3-An Envelope Call control by SIM envelope is sent to SIM`
	`ToolkitRegistry.clearEvent() is called for EVENT_CALL_CONTROL_BY_SIM.`
	`ToolkitRegistry.setEvent() method is called for EVENT_CALL_CONTROL_BY_SIM.`
		1-Applet2 is triggered by EVENT_FORMATTED_SMS_PP_ENV.
		3- Applet1 is triggered
		Applet1 finalizes.
		Applet2 finalizes
			2- A proactive command DISPLAY TEXT is sent and applet is suspended until the terminal response
			TERMINAL RESPONSE of DISPLAY TEXT is sent to the SIM

3	**Applet triggering** `An Envelope Call control by SIM envelope is sent ~~ot~~ to SIM`	Applet2 is triggered. (Applet1 is not triggered)	

6.3.3.6.4 Test Coverage

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

6.3.3.7 EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM

Test Area Reference: FWK_APT_EMCN

6.3.3.7.1 Conformance Requirement

6.3.3.7.1.1 Normal Execution

- CRRN1: The applet is triggered by the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM once it has registered to this event and an Envelope MO Short Message Control.
- CRRN2: The applet is not triggered by the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM once it has deregistered from this event.

6.3.3.7.1.2 Parameters error

No requirements.

6.3.3.7.1.3 Context Errors

No requirements.

6.3.3.7.2 Test Suite Files

Test Script:	FWK_APT_EMCN_1.scr
Test Applet:	FWK_APT_EMCN_1.java
	FWK_APT_EMCN_2.java
Load Script:	FWK_APT_EMCN_1.ldr
Cleanup Script:	FWK_APT_EMCN_1.clr
Parameter File:	FWK_APT_EMCN_1.par

6.3.3.7.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet registration to EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM and triggering** Applet1 is reggistered to EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM. Applet2 is registered to EVENT_FORMATTED_SMS_PP_ENV. 1-An Envelope MO short message envelope is sent to SIM	1- Applet1 is triggered.	

Id	Description	API/Framework Expectation	APDU Expectation
2	**Applet deregistration and registration of the third applet to EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM.** **The STF shall not reply busy to a call control envelope** 1-An Envelope formatted SMS PP envelope is sent to SIM. Applet2 builds a DISPLAY TEXT proactive command. 2-ProactiveHandler.send() method is called. 3-An Envelope MO Short message envelope is sent to SIM ToolkitRegistry.clearEvent() for EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM. ToolkitRegistry.setEvent() method is called for EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM.	1- Applet2 is triggered. 3- Applet1 is triggered. Applet1 finalizes. Applet2 finalizes.	2- A Proactive command DISPLAY TEXT is sent and applet is suspended until the terminal response TERMINAL RESPONSE of DISPLAY TEXT is sent to the SIM
3	**Applet3 triggering** An Envelope MO SMS control by SIM envelope is sent ~~ot~~to SIM	Applet2 is triggered. (Applet1 is not triggered)	

6.3.3.7.4 Test Coverage

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

6.3.3.18 EVENT_STATUS_COMMAND

Test Area Reference: FWK_APT_ESTC

6.3.3.18.1 Conformance Requirement

6.3.3.18.1.1 Normal Execution

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

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

6.3.3.18.1.2 Parameters error

No requirements.

6.3.3.18.1.3 Context Errors

No requirements.

6.3.3.18.2 Test Suite Files

Test Script:	FWK_APT_ESTC_1.scr
Test Applet:	FWK_APT_ESTC_1.java
	FWK_APT_ESTC_2.java
	FWK_APT_ESTC_3.java
Load Script:	FWK_APT_ESTC_1.ldr
Cleanup Script:	FWK_APT_ESTC_1.clr
Parameter File:	FWK_APT_ESTC_1.par

6.3.3.18.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applets registration to EVENT_STATUS_COMMAND and triggering** Applet1 is registered to EVENT_STATUS_COMMAND using the requestPollInterval() command. Applet2 is registered to EVENT_STATUS_COMMAND using the RequestPollInterval() command. Applet3 is registered to EVENT_FORMATTED_SMS_PP_ENV. 1-A status command is sent to SIM	1- Applet1 is triggered. Applet1 finalizes 2- Applet2 is triggered. Applet2 finalizes 3- ~~Applet 3~~Applet3 is not triggered	

Id	Description	API/Framework Expectation	APDU Expectation
2	**Applet deregistration and registration of the third applet to EVENT_STATUS_COMMAND. The STF shall not reply busy to a call control envelope**		
	1-A formatted sms pp envelope is sent to SIM	1- Applet3 is triggered.	
	Applet3 builds a DISPLAY TEXT. 2- ProactiveHandler.send() is called		2- A proactive command DISPLAY TEXT is sent and applet is suspended until the terminal response
	3-A status command is sent to SIM.	3- Applet1 is triggered.	
	requestPollInteval with POLL_NO_DURATION is called		
	requestPollInteval with POLL_NO_DURATION is called	Applet1 finalizes 4- Applet2 is triggered.	
	requestPollInterval() method is called.	Applet2 finalizes	
		Applet3 finalizes	5- TERMINAL RESPONSE of DISPLAY TEXT is sent to the SIM
3	**Applet3 triggering**		
	Perform SIM initialization with all the facilities supported		
	Status command is sent to SIM.	Applet3 is triggered. (Applet1 and Applet2 are not triggered)	

6.3.3.18.4 Test Coverage

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

6.3.6 Framework Security Management

Security Parameters

The table that follows contains the security parameters that shall be used when the 3GPP TS 23.048 [8]security is required in the test cases developed in the current subclause.

Parameter	Value in hexadecimal
KIC	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 subclause 4.7.3.1 apply.

6.3.6.1 Input Data

Test Area Reference: FWK_FWS_INDA

6.3.6.1.1 Conformance Requirements

6.3.6.1.1.1 Normal Execution

- CRRN1: If the SIM receives an envelope APDU containing an SMS_PP_DATADOWNLOAD BER TLV formatted according to 3GPP TS 23.048 [8], the SIM Toolkit Framework shall verify the security of the SMS TPDU.
- CRRN2: The toolkit applet will only be triggered if the TAR is known and the security verified.
- CRRN3: If the SIM receives an envelope APDU containing an SMS_CB_DATADOWNLOAD formatted according to 3GPP TS 23.048 [8], the SIM Toolkit Framework shall verify the security of the cell broadcast page.
- CRRN4: If the SIM receives an Update Record EFsms instruction formatted according to TS 23.048[8], the SIM Toolkit Framework shall verify the security of the SMS.
- CRRN5: The STF shall provide the input data deciphered.

6.3.6.1.1.2 Parameters error

No requirements.

6.3.6.1.1.3 Context Errors

No requirements.

6.3.6.1.2 Test Area Files

Test Script:	FWK_FWS_INDA_1.scr
Test Applet:	FWK_FWS_INDA_1.java
	FWK_FWS_INDA_2.java
	FWK_FWS_INDA_3.java
	FWK_FWS_INDA_4.java
	FWK_FWS_INDA_5.java
	FWK_FWS_INDA_6.java
Load Script:	FWK_FWS_INDA_1.ldr
Cleanup Script:	FWK_FWS_INDA_1.clr
Parameter File:	FWK_FWS_INDA_1.par

6.3.6.1.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation

Id	Description	API/Framework Expectation	APDU Expectation

1	**Framework checks the Cryptographic checksum and deciphers the data**
	Applet1 is loaded and installed
	1-Envelope(SMS-PP) single and formatted is sent to the SIM with this features:
	Ciphering;
	Cryptographic checksum;
	No proof of receipt;
	TAR of Applet~~-~~1;
	Data = 01
	2- Short Message concatenated and formatted is sent to the SIM by an Envelope (SMS PP)with these features:
	Ciphering;
	Cryptographic checksum;
	No proof of receipt;
	TAR of Applet~~-~~1;
	Data length is 150.
		1- ~~The a~~Applet~~-~~1 is triggered and the value integrity is checked.
		2- ~~The a~~Applet~~-~~1 is triggered and the value integrity is checked
			1- The SIM answers to the Envelope with status words 9000
			2- The SIM answers to the Envelope with status words 9000

Id	Description	API/Framework Expectation	APDU Expectation
2	Triggering two different applets with different security Applet2 is installed 1-Envelope(SMS-PP) single and formatted is sent to the SIM with this features: Ciphering; Cryptographic checksum; No proof of receipt; TAR of Applet-1 Data = 03 2- Short Message concatenated and formatted is sent to the SIM by an Envelope (SMS PP)with these features: Ciphering; Cryptographic checksum; No proof of receipt; TAR of Applet-1 Data length = 150 3-Envelope(SMS-PP) single and formatted is sent to the SIM with this features: No ciphering; No cryptographic checksum; No proof of receipt; TAR of Applet-2 Data = 05 4- Short Message concatenated and formatted is sent to the SIM by an Envelope (SMS PP)with these features:~~:~~ No ciphering; No cryptographic checksum; No proof of receipt; TAR of Applet-2 Data length = 150.	1- Applet-1 is triggered and the value integrity is checked 2- Applet-1 is triggered and the value integrity is checked 3- Applet-2 is triggered and the value integrity is checked 4- Applet-2 is triggered and the value integrity is checked	1- The SIM answers to the Envelope with status words 9000 2- The SIM answers to the Envelope with status words 9000 3- The SIM answers to the Envelope with status words 9000 4- The SIM answers to the Envelope with status words 9000
3	**Envelope(SMS-PP) formatted with wrong cryptographic checksum** 1-Envelope 03.48 single and formatted is sent to the SIM with this features: No ciphering; Wrong cryptographic checksum; No proof of receipt; TAR of Applet-1 Data = 07 2- Short Message concatenated and formatted is sent to the SIM by an Envelope (SMS PP)with these features: No ciphering; Wrong cryptographic checksum; No proof of receipt; TAR of Applet-1 Data length = 150	1- No applet is triggered. 2- No applet is triggered.	1- The SIM answers to the Envelope with status words 9000

Id	Description	API/Framework Expectation	APDU Expectation
4	**Framework checks the Cryptographic checksum and deciphers the data** Applet3 is loaded and installed 1-Envelope(SMS-CB) formatted is sent to the SIM with this features: Ciphering; Cryptographic checksum; No proof of receipt; Data = 01	1- Applet3 is triggered and the value integrity is checked	1- The SIM answers to the Envelope with status words 9000
5	**Triggering two different applets with different security on Envelope(SMS-CB) formatted** Applet4 is installed 1-Envelope(SMS-CB) formatted is sent to the SIM with this features: Ciphering; Cryptographic checksum; No proof of receipt; TAR of ~~Applet 3~~Applet3 Data = 02 2-Envelope(SMS-CB) formatted is sent to the SIM with this features: No ciphering; No cryptographic checksum; No proof of receipt; TAR of ~~Applet 4~~Applet4 Data = 03	1- Applet3 is triggered and the value integrity is checked 2- Applet4 is triggered and the value integrity is checked	1- The SIM answers to the Envelope with status words 9000 2- The SIM answers to the Envelope with status words 9000

6	**Envelope(SMS-CB) formatted with wrong cryptographic checksum**
	No ciphering;
	Wrong Cryptographic checksum;
	No proof of receipt;
	TAR of ~~Applet 3~~Applet3
	Data = 04
		No applet is triggered	1- The SIM answers to the Envelope with status words 9000

Id	Description	API/Framework Expectation	APDU Expectation

7	**Framework checks the Cryptographic checksum and deciphers the data**
	Applet5 is installed
	1- Short Message single and formatted is sent to the SIM by Update Record EFsms instruction with these features:
	Ciphering;
	Cryptographic checksum;
	No proof of receipt;
	TAR of Applet5;
	Data = 01
	2- Short Message concatenated and formatted is sent to the SIM by Update Record EFsms instruction with these features:
	Ciphering;
	Cryptographic checksum;
	No proof of receipt;
	TAR of Applet5;
	Data length = 150.
		1- ~~The a~~Applet5 is triggered and the value integrity is checked.
		2- ~~The a~~Applet5 is triggered and the value integrity is checked
			1- The SIM answers to the Update Record EFsms instruction with status words 9000
			2- The SIM answers to the Update Record EFsms instruction with status words 9000

8	**Triggering two different applets with different security** Applet6 is installed 1- Short Message single and formatted is sent to the SIM by Update Record EFsms instruction with these features: Ciphering; Cryptographic checksum; No proof of receipt; TAR of Applet5 Data = 03 2- Short Message concatenated and formatted is sent to the SIM by Update Record EFsms instruction with these features: Ciphering; Cryptographic checksum; No proof of receipt; TAR of Applet5 Data length = 150. 3- Short Message single and formatted is sent to the SIM by Update Record EFsms instruction with these features: No ciphering; No cryptographic checksum; No proof of receipt; TAR of Applet6; Data = 05 4- Short Message concatenated and formatted is sent to the SIM by Update Record EFsms instruction with these features: No ciphering; No cryptographic checksum; No proof of receipt; TAR of Applet6; Data length = 150.	1- Applet5 is triggered and the value integrity is checked. 2- Applet5 is triggered and the value integrity is checked. 3- Applet6 is triggered and the value integrity is checked. 4- Applet6 is triggered and the value integrity is checked.	1- The SIM answers to the Update Record EFsms instruction with status words 9000 2- The SIM answers to the Update Record EFsms instruction with status words 9000 3- The SIM answers to the Update Record EFsms instruction with status words 9000 4- The SIM answers to the Update Record EFsms instruction with status words 9000
9	**Update Record EFsms instruction formatted with wrong cryptographic checksum** 1- Short Message single and formatted is sent to the SIM by Update Record EFsms		

Id	Description	API/Framework Expectation	APDU Expectation
	instruction with these features:No ciphering; Wrong Cryptographic checksum; No proof of receipt; TAR of Applet5 Data = 07 2- Short Message concatenated and formatted is sent to the SIM by Update Record EFsms instruction with these features: No ciphering; Wrong Cryptographic checksum; No proof of receipt; TAR of Applet5 Data length = 150	1- No applet is triggered. 2- No applet is triggered.	1- The SIM answers to the Update Record EFsms instruction with status words 9000 2- The SIM answers to the Update Record EFsms instruction with status words 9000

6.3.6.1.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2, 3
CRRN2	3,6,9
CRRN3	4, 5, 6
CRRN4	7,8,9
CRRN5	1,2,4,5,7,8

6.3.7 Envelope Response Posting

6.3.7.1 EVENT_CALL_CONTROL_BY_SIM

Test Area Reference: FWK_ERP_ECCN

6.3.7.1.1 Conformance Requirements

6.3.7.1.1.1 Normal Execution

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

6.3.7.1.1.2 Parameters error

No requirements.

6.3.7.1.1.3 Context Errors

No requirements.

6.3.7.1.2 Test Area Files

Test Script:	FWK_ERP_ECCN_1.scr
Test Applet:	FWK_ERP_ECCN_1.java
	FWK_ERP_ECCN_2.java
	FWK_ERP_ECCN_3.java
Load Script:	FWK_ERP_ECCN_1.ldr

Cleanup Script:	FWK_ERP_ECCN_1.clr
Parameter File:	FWK_ERP_ECCN_1.par

6.3.7.1.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation

1	**Applet1 is registered on the EVENT_CALL_CONTROL_BY_SIM, Applet2 is registered and triggered on the EVENT_MENU_SELECTION.**
	`1-Applet2 invokes the method send()and no fetch is performed`
	`2-Envelope(Call Control) is sent to the SIM`
	`3-Applet1 calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming dialling number into +11 22 33 44.`
	`4-A Fetch command is sent to the SIM`
	`5-A Terminal Response command is sent to the SIM`
	`6-Delete` ~~`applet1`~~`Applet1 & `~~`applet2`~~`Applet2`
	`7-Install `~~`applet3`~~`Applet3`
		Applet2 is suspended
		Applet1 is triggered.
		Applet2's execution shall continue.
			The SIM answer 9Fxx to the Envelope(Call Control)
			The dialling number is retrieved with a GetResponse command. The SIM answers to the Get Response command with status words 91xx.

2	**~~Applet 3~~Applet3 is registered on both the events EVENT_CALL_CONTROL_BY_SIM and EVENT_MENU_SELECTION.**
	`1-Envelope Menu Selection is sent to the SIM.`
	`2-Applet3 invokes the method send()and no fetch is performed)`
	`3-Envelope(Call Control) is sent to the SIM`
	`4-Applet3 calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming dialling number into +11 22 33 44.`
	`5-A Fetch command is sent to the SIM`
	`6-A Terminal Response command is sent to the SIM`
		Applet3 is triggered on the EVENT_MENU_SELECTION
		Applet3 is suspended on the send() method
		Applet3 is triggered on the EVENT_CALL_CONTROL_BY_SIM.
		The Applet3's execution shall continue.
			The SIM answer 9Fxx to the Envelope(Call Control)
			The dialling number is retrieved with a GetResponse command.
			The SIM answers to the Get Response command with status words 91xx.

6.3.7.1.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2

6.3.7.2 EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM

Test Area Reference: FWK_ERP_EMCN

6.3.7.2.1 Conformance Requirements

6.3.7.2.1.1 Normal Execution

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

6.3.6.2.1.2 Parameters error

No requirements.

6.3.6.2.1.3 Context Errors

No requirements.

6.3.7.2.2 Test Area Files

Test Script:	FWK_ERP_EMCN_1.scr
Test Applet:	FWK_ERP_EMCN_1.java
	FWK_ERP_EMCN_2.java
	FWK_ERP_EMCN_3.java
Load Script:	FWK_ERP_EMCN_1.ldr
Cleanup Script:	FWK_ERP_EMCN_1.clr
Parameter File:	FWK_ERP_EMCN_1.par

6.3.7.2.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
1	**Applet1 is registered on the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM; Applet2 is registered and triggered on the EVENT_MENU_SELECTION.** 1-Applet2 invokes the method send()and no fetch is performed) 2-Envelope(MO-SM control) is sent to the SIM 3-Applet1 calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming TP_Destination_Address and any RP_Destination_Address of the Service Center into +11 22 33 44 4-A Fetch command is sent to the SIM 5-A Terminal Response command is sent to the SIM 6-Delete ~~applet1 ~~Applet1 & ~~applet2~~Applet2 7-Install ~~applet3~~Applet3	Applet2 is suspended Applet1 is triggered. The Applet's execution shall continue.	The SIM answers 9Fxx to the Envelope(MO-Short Message Control) The TP_Destination_Address is retrieved with a GetResponse command. The SIM answers to the Get Response command with status words 91xx.
2	**~~Applet 3~~Applet3 is registered on both the events EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM and EVENT_MENU_SELECTION.** 1-Applet3 invokes the method send()and no fetch is performed) 2-Envelope(MO-SM control) is sent to the SIM 3-Applet3 calls the method EnvelopeResponseHandler.postASBERTLV() to change any incoming TP_Destination_Address and any RP_Destination_Address of the Service Center into +11 22 33 44. 4-A Fetch command is sent to the SIM 5-A Terminal Response command is sent to the SIM	~~Applet 3~~Applet3 is suspended on the send() method Applet3 is triggered on the EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM. The Applet3's execution shall continue.	The SIM answers 9Fxx to the Envelope(MO-Short Message Control) The TP_Destination_Address is retrieved with a GetResponse command. The SIM answers to the Get Response command with status words 91xx.

6.3.7.2.4 Test Coverage

CRR Number	Test Case Number
CRRN1	1, 2

6.3.8 Toolkit Installation

6.3.8.6 Access Domain

Test Area Reference: FWK_TIN_ACDO

6.3.8.6.1 Conformance Requirements

6.3.8.6.1.1 Normal execution

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

6.3.8.6.1.2 Parameters errors

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

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

6.3.8.6.1.3 Context errors

No requirements.

6.3.8.6.2 Test suite files

Test Script:	FWK_TIN_ACDO_1.scr
Test Applet:	FWK_TIN_ACDO_1.java
	FWK_TIN_ACDO_2.java
	FWK_TIN_ACDO_3.java
Load Script:	FWK_TIN_ACDO_1.ldr
Cleanup Script:	FWK_TIN_ACDO_1.clr
Parameter File:	FWK_TIN_ACDO_1.par

6.3.8.6.3 Test Procedure

Id	Description	API/Framework Expectation	APDU Expectation
0	Install (install) applet1 with: - Length of Access Domain field value is '1' - Access Domain Parameter value is '00' (full access to the GSM File System) Install (install) applet2 with:		

Id	Description	API/Framework Expectation	APDU Expectation
	- Length of Access Domain field value is '1' - Access Domain Parameter value is 'FF' (No access to the GSM File System) Install (install) applet3 with: - Length of Access Domain field value is '1' - Access Domain Parameter value is '00' (full access to the GSM File System)		

1	**readBinary/readRecord method with full Access Domain Parameter**
	
	1- Select EF-TARU file whose Read access condition is ALWAYS
	Perform the readBinary method:
	fileOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
	
	2- Select EF-SMS file whose Read access condition is CHV1
	Perform the readRecord method:
	
	recNumber = 1
	mode = REC_ACC_MODE_ABSOLUTE_CURRENT
	recOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
	
	3- Select EF-TRAC file whose Read access condition is CHV2
	Perform the readBinary method:
	fileOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
	
	4- Select EF-SUME file Read access condition is ADM0
	Perform the readBinary method:
	fileOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
	
	5- Select EF-TNR file whose Read access condition is NEVER
	Perform the readBinary method:
	fileOffset = 0
	resp = abRead[]
	respOffset = 0
	respLength = 3
		1 to 4- no exception is thrown
		
		5- SIMViewException AC_NOT_FULFILLED is thrown	

Id	Description	API/Framework Expectation	APDU Expectation
2	**updateBinary/updateRecord method with full Access Domain Parameter**	1 to 4- no exception is thrown	
3	**invalidate method with full Access Domain Parameter** 1- Select EF-TNR file whose Invalidate access condition is ALWAYS Perform the invalidate method 2- Select EF-TIAC file whose Invalidate access condition is CHV1 Perform the invalidate method 3- Select EF-ADN file whose Invalidate access condition is CHV2 Perform the invalidate method 4- Select EF-SUME file Invalidate access condition is ADM0 Perform the invalidate method 5- Select EF-CNIV file whose Invalidate access condition is NEVER Perform the invalidate method	1 to 4- no exception is thrown 5- SIMViewException AC_NOT_FULFILLED is thrown	

Id	Description	API/Framework Expectation	APDU Expectation
4	**rehabilitate method with full Access Domain Parameter** 1- Select EF-TNR file whose Rehabilitate access condition is ALWAYS Perform the rehabilitate method 2- Select EF-IMSI file whose Rehabilitate access condition is CHV1 Perform the rehabilitate method 3- Select EF-ADN file whose Rehabilitate access condition is CHV2 Perform the rehabilitate method 4- Select EF-SUME file Rehabilitate access condition is ADM0 Perform the rehabilitate method 5- Select EF-CNRI file whose Rehabilitate access condition is NEVER Perform the rehabilitate method	1 to 4- no exception is thrown 5- SIMViewException AC_NOT_FULFILLED is thrown	

5	**increase method with full Access Domain Parameter**
	
	1- Select EF-CNR~~U~~ file whose Increase access condition is ALWAYS
	Perform the increase method:
	incr = abIncreaseValue[]
	incrOffset = 0
	resp = abRead[]
	respOffset = 0
	
	2- Select EF-ACM file whose Increase access condition is CHV1
	Perform the increase method:
	incr = abIncreaseValue[]
	incrOffset = 0
	resp = abRead[]
	respOffset = 0
	
	3- Select EF-CIAC file whose Increase access condition is CHV2
	Perform the increase method:
	incr = abIncreaseValue[]
	incrOffset = 0
	resp = abRead[]
	respOffset = 0
	
	4- Select EF-CIAA file Increase access condition is ADM0
	Perform the increase method:
	incr = abIncreaseValue[]
	incrOffset = 0
	resp = abRead[]
	respOffset = 0
	
	5- Select EF-CNU~~R~~ file whose Increase access condition is NEVER
	Perform the increase method
		1 to 4- no exception is thrown
		
		5- SIMViewException AC_NOT_FULFILLED is thrown	

Id	Description	API/Framework Expectation	APDU Expectation

6	**readBinary method with no Access Domain Parameter**
	`Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Select EF-TARU file whose Read access condition is ALWAYS`
	`Perform the readBinary method:`
	`fileOffset = 0`
	`resp = abRead[]`
	`respOffset = 0`
	`respLength = 3`
	`t`
		SIMViewException AC_NOT_FULFILLED is thrown	

7	**updateRecord method with no Access Domain Parameter**
	`Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Select EF-SMS file whose Update access condition is CHV1`
	`Perform the updateRecord method:`
	`fileOffset = 0`
	`resp = abUpdate[]`
	`respOffset = 0`
	`respLength = 3`
		SIMViewException AC_NOT_FULFILLED is thrown	

8	**invalidate method with no Access Domain Parameter**
	`Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Select EF-ADN file whose Invalidate access condition is CHV2`
	`Perform the invalidate method`
		SIMViewException AC_NOT_FULFILLED is thrown	

9	**rehabilitate method with no Access Domain Parameter**
	`Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Select EF-SUME file Rehabilitate access condition is ADM0`
	`Perform the rehabilitate method`
		SIMViewException AC_NOT_FULFILLED is thrown	

10	**increase method with no Access Domain Parameter**
	`Send an Envelope that triggers the applet with the EVENT_UNFORMATTED_SMS_PP_ENV event.`
	`Select EF-CNR file whose Increase access condition is NEVER`
	`Perform the increase method`
		SIMViewException AC_NOT_FULFILLED is thrown
		Applet2 finalizes
		Applet3 restore EF-SUME	

6.3.8.6.4 Test Coverage

NOTE: As Item Position management is not fully specified in the 3GPP TS 43.019 [7] or 3GPP TS 23.048 [8] all possible tests cannot be performed.

CRR number	Test case number
CRRN1	1, 2, 3, 4, 5
CRRP1	Not tested
CRRP2	6, 7, 8, 9, 10

Annex C (normative): Default Prepersonalization

C.1 General Default Prepersonalization

This table shows the default prepersonalization, the file system and the files' content, that the test SIM cards shall contain unless otherwise stated.

Name	Identifier	Default Value	Special Features
EF$_{ICCID}$	2FE2	0F FF FF FF FF FF FF FF FF FF	This value is not compliant with 3GPP TS 51.011 [3]
EF$_{IMSI}$	6F07	FF FF FF FF FF FF FF FF FF	This value is not compliant with 3GPP TS 51.011 [3]
EF$_{LP}$	6F05	01 FF FF FF	
EF$_{Kc}$	6F20	FF FF FF FF FF FF FF FF 07	
EF$_{PLMNsel}$	6F30	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
EF$_{HPLMN}$	6F31	05	
EF$_{ACMmax}$	6F37	00 00 00	Access condition UPDATE: CHV1
EF$_{SST}$	6F38	FF 3F C3 0F 0C 00 FF 0F 00 33	
EF$_{ACM}$	6F39	00 00 00	Access condition UPDATE: CHV1
EF$_{PUCT}$	6F41	FF FF FF 00 00	Access condition UPDATE: CHV1
EF$_{BCCH}$	6F74	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
EF$_{ACC}$	6F78	00 00	
EF$_{FPLMN}$	6F7B	FF FF FF FF FF FF FF FF FF FF FF FF	
EF$_{LOCI}$	6F7E	FF FF FF FF 00 F0 00 00 00 FF 01	
EF$_{AD}$	6FAD	00 FF FF	
EF$_{Phase}$	6FAE	03	
EF$_{FDN}$	6F3B	Default value in all the records: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 5
EF$_{SMSP}$	6F42	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF$_{LND}$	6F44	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF$_{SMSS}$	6F43	FF FF	
EF$_{SMS}$	6F3C	1st record: 00 FF … FF(length 176) 2nd record:00 FF … FF(length 176) 3rd record: 00 FF … FF(length 176)	Records: 3
EF$_{ADN}$	6F3A	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF$_{CCP}$	6F3D	FF FF FF FF FF FF FF FF FF FF FF FF FF FF	
EF$_{MSISDN}$	6F40	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF$_{SDN}$	6F49	FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF	Records: 1
EF$_{SUME}$	6F54	85 0C 54 4F 4F 4C 4B 49 54 20 54 45 53 54 FF FF FF FF	
EF$_{CBMI}$	6F45	FF FF	
EF$_{CBMID}$	6F48	10 80	

EF_{CBMIR}	6F50	10 80 10 9F	
EF_{IMG}	4F20	FF FF FF FF FF FF FF FF FF FF FF	

The default value for the CHV1 shall be "0x31 0x31 0x31 0x31 0xFF 0xFF 0xFF 0xFF" and its state shall be 'disabled' during test applets execution.