3GPP TSG-T plenary meeting #20 Hämeenlinna, Finland, 4-6 May 2003

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 request:

T3 Doc	Spec	CR	Rev	Phase	Subject	Cat	V. old	V. new
T3-030409	51.013	001	-	Rel-5	Update of 51.013 Specification for Release 5	В	4.0.1	5.0.0

Tdoc T3-030409Rev T3-030300

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

	[1]	(void)
	[2]	(void)
	[3]	3GPP TS 51.011: " Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface".
	[4]	3GPP TS 11.14 <u>51.014</u> : " Specification of the SIM application toolkit for the Subscriber Identity Module – Mobile Equipment (SIM – ME) interface".
	[5]	3GPP TSGSM 11.17: "Subscriber Identity Module" (SIM) conformance test specification".
ļ	[6]	(void)
	[7]	3GPP TS GSM 43.019 Rel-5: "Subscriber Identity Module Application Programming Interface (SIM API); SIM API for Java Card TM ; Stage 2".
	[8]	3GPP TS 23.048 Rel-45: " Security Mechanisms for the (U)SIM application toolkit; Stage 2"
I	[9]	ISO/IEC 7816-3 (1997) " Identification cards - Integrated circuit(s) cards with contacts, Part 3: Electronic signals and transmission protocols".
	[10]	3GPP TS 42.019: "Subscriber Identity Module Application Programming Interface (SIM API); Service description; Stage 1".
	[11]	SUN Java Card Specification "Java Card 2.1 API Specification".
	[12]	SUN Java Card Specification "Java Card 2.1 Runtime Environment Specification".
	[13]	SUN Java Card Specification "Java Card 2.1 VM Architecture Specification".
		SUN Java Card Specifications can be downloaded at http://java.sun.com/products/javacard
	[14]	ETSI TS 101 220 "Integrated Circuit Cards (ICC); ETSI numbering system for telecommunication; Application providers (AID)".
	[15]	3GPP TSGSM 11.10-1: "Digital cellular telecommunication system (Phase 2+); Mobile Station (MS) conformance specification; Part 1: Conformance specification".

4 Test Environment

4.1 Applicability

The tests defined in this specification shall be performed taking into account the services supported by the card as specified in the EF_{SST} file.

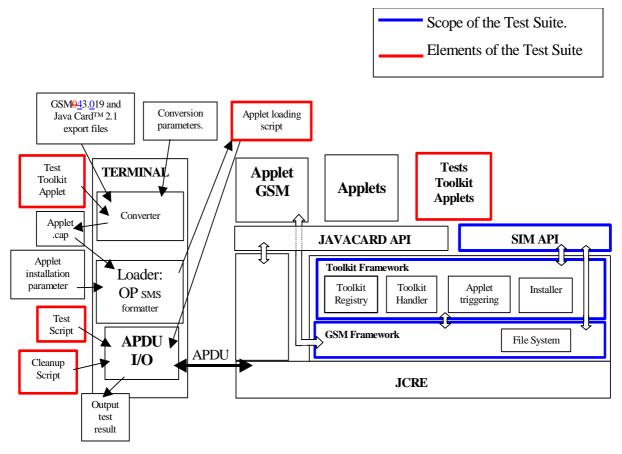
This specification contains tests that test interoperability at the API level. This specification does not currently contain tests for interoperability at the SIM API framework and at the byte code level. These are for further study.

The tests defined in this specification are applicable to cards implementing TS 03.19 [7] version 7.4.0 TS 43.019 [7] unless otherwise stated.

The tests defined in this specification require that the card support the concatenation process with 2 concatenated SMS. Therefore the envelope handler shall support 280 bytes of data.

4.2 Test environment description

The general architecture for the test environment is:

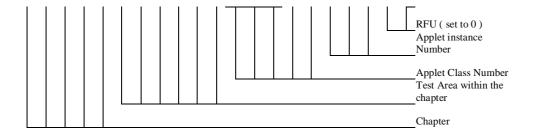


Note: This diagram shows the test architecture required to test interoperability at both API and bytcode level. The latter is currently not included in the current specification. The diagram is for information.

4.6.2 Specific Test Applet Name for Framework

Specific applet test name (bits b4-b24):

b4 b5 b6 b7 b8 b9 b10 b11 b12 b13 b14 b15 b16 b17 b18 b19 b20 b21 b22 b23 b24



for Chapter (5 bits)

00001 Toolkit Installation Parameters

00010 Minimum Handler Availability

00011 Handler Integrity

00100 Applet Triggering

00101 Proactive Command Sending

00110 Framework Security

00111 Envelope Response Posting

01000 File System Context

01001 Exception Handling

01010 Other parts transferred to framework from API

01011 Concatenation processing

other are RFU

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

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

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

6 API Test Plan

6.2 Package sim.toolkit

6.2.2 Interface ToolkitInterface

6.2.2.1.1 Conformance Requirement:

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

6.2.2.1.1.1 Normal execution

CRRN1: This interface must be implemented by a Toolkit applet (which extends the javacard.framework.Applet class) so that it can be triggered by the Toolkit Handler according to the registration information.

CRRN2: The Toolkit applet will have to implement the processToolkit shared method so that the following events can be notified:

Event	Description	
EVENT_PROFILE_DOWNLOAD	Terminal Profile command reception	
EVENT_FORMATTED_SMS_PP_ENV	Formatted envelope SMS-PP Data Download	
	reception	
EVENT_FORMATTED_SMS_PP_UPD	Formatted Update Record EF SMS	
EVENT_FORMATTED_SMS_CB	Formatted envelope Cell Broadcast Data	
	Download command reception	
EVENT_UNFORMATTED_SMS_PP_ENV	Unformatted Envelope SMS-PP Data Download	
	reception	
EVENT_UNFORMATTED_SMS_PP_UPD	Unformatted Update Record EF SMS	
EVENT_UNFORMATTED_SMS_CB	Unformatted Cell Broadcast Data Download	
	command reception	
EVENT_MENU_SELECTION	Envelope Menu Selection command reception	
EVENT_MENU_SELECTION_HELP_REQUEST	Envelope Menu Selection Help Request	
	command reception	
EVENT_CALL_CONTROL_BY_SIM	Envelope Call Control by SIM command	
	reception	
EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM	Envelope MO Short Message Control by SIM	
	command reception	
EVENT_TIMER_EXPIRATION	Envelope Timer Expiration	
EVENT_EVENT_DOWNLOAD_MT_CALL	Envelope Event Download - MT call	
EVENT_EVENT_DOWNLOAD_CALL_CONNECTED Envelope Event Download - Call connected		
EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED Event Download - Call disconnected		
EVENT_EVENT_DOWNLOAD_LOCATION_STATUS	Envelope Event Download - Location status	
EVENT_EVENT_DOWNLOAD_USER_ACTIVITY	Envelope Event Download - User activity	
EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE	Envelope Event Download - Idle screen available	
EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS	Envelope Event Download - Card Reader Status	
EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION	Envelope Event Download – Language Selection	
EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION	Envelope Event Download – Browser	
	Termination	
_EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE	Envelope Event Download - Data Available	
EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS	Envelope Event Download - Channel Status	
_EVENT_FIRST_COMMAND_AFTER_SELECT	First command performed after select GSM	
	application or ATR	
EVENT_STATUS_COMMAND	Status APDU command event	
EVENT_UNRECOGNIZED_ENVELOPE	Unrecognized Envelope command reception	

6.2.4 Class EnvelopeHandler

6.2.4.3 Method getSecuredDataLength

Test Area Reference: API_2_ENH_GSDL

6.2.4.3.1 Conformance Requirement:

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

6.2.4.3.1.1 Normal execution

CRRN1: The method shall return the length of the secured dataSecured Data from the Command Packet in the SMS

TPDU (simple or concatenated) or Cell Broadcast Page Simple TLV contained in a SMS TPDU TLV.the

Envelope handler

CRRN2: The length is from the first SMS TPDU TLV or Cell Broadcast Page Simple TLV.

CRRN3: The length should not include padding bytes.

CRRN4: The method can be used if the event is EVENT_FORMATTED_SMS_PP_ENV and if the SMS TP-UD is formatted according to GSM03.48.TS 23.048 [8].

CRRN5: The method can be used if the event is EVENT_FORMATTED_SMS_PP_UPD and if the SMS TP-UD is formatted according to GSM03.48.TS 23.048 [8].

CRRN6: The method can be used if the event is EVENT_FORMATTED_SMS_CB and if the Cell Broadcast Page is formatted according to TS 23.048 [8].

CRRN7: If the method is successful and if the event is EVENT_FORMATTED_SMS_PP_ENV, the selected TLV should be the SMS TPDU TLV.

CRRN8: If the method is successful and if the event is **EVENT FORMATTED SMS PP UPD**, the selected TLV should be the SMS TPDU TLV.

<u>CRRN9: If the method is successful and if the event is EVENT_FORMATTED_SMS_CB</u>, the selected TLV should be the Cell Broadcast Page TLV.

Parameters error

No requirements

6.2.4.3.1.2 Context errors

CRRC1: The method shall thrown ToolkitException (UNAVAILABLE_ELEMENT) in case of unavailable SMS TPDU TLV element or Cell Broadcast Page Simple TLV

CRRC2: The method shall thrown ToolkitException (UNAVAILABLE_ELEMENT) in case of missing Secured Data. wrong data format.

6.2.4.3.2 Test suite files

Specific triggering:

- FORMATTED SMS CB
- UNFORMATTED SMS CB
- FORMATTED SMS PP UPD
- UNFORMATED SMS PP ENV

 ——For Formatted triggering if CC/RC/DS is used, the security parameters are the one used for downloading applications.

Test Script: API_2_ENH_GSDL_1.scr

Test Applet: API_2_ENH_GSDL_1.java

Load Script: API_2_ENH_GSDL_1.ldr

Cleanup Script: API_2_ENH_GSLD_1.clr

Parameter File: API_2_ENH_GSLD_1.parAPI_2_ENH_GSDL_1.clr

Parameter File: API 2 ENH GSDL 1.par

6.2.4.3.3 Test procedure

ld	Description	API Expectation	APDU Expectation
	FORMATTED SMS PP ENV Triggering		

4	Task with FORMATTED ONG DD FNV and TD	D = 4 = 00 A	
4	Test with FORMATTED_SMS_PP_ENV and TP- OA length of 2	Returns 0x2A	
1		Deturne 0x000A	
1	Test with FORMATTED_SMS_PP_ENV and TP-	Returns 0x002A	
2	OA length of 2 Test with TP-OA length of 6	Returns 0x2A	
2	Test with TP-OA length of 6	Returns 0x002A	
3	Test with TP-OA length of 12	Returns 0x002A	
	Test with TP-OA length of 12		
3		Returns 0x002A	
4	Test with RC/CC/DS length of 0	Returns 0x10	
4	Test with RC/CC/DS length of 0	Returns 0x0010	
5	Test with RC/CC/DS length of 8	Returns 0x10	
<u>5</u>	Test with RC/CC/DS length of 8	Returns 0x0010	
6	Test with PCNTR = 0	Returns 0x10	
<u>6</u>	Test with PCNTR = 0	Returns 0x0010	
7	Test with PCNTR = 7	Returns 0x05	
<u>7</u>	Test with PCNTR = 7	Returns 0x0005	
8	Test with SecuredDataLength = 00	Returns 0x00	
<u>8</u>	Test with Secured Data Length = 00	Returns 0x0000	
9	Test with UserDataLength = 0x33	Returns 0x33	
9	Test with Secured Data Length = 0x33	Returns 0x0033	
10	Test with UserDataLength = 0x 6C	Returns 0x 6C	
<u>10</u>	Test with Secured Data Length = 0x6C (UDL =	Returns 0x006C	
	<u>0x7F)</u>		
	Test with UserDataLength = 0x 6D	Returns 0x 6D	
<u>11</u>	Test with Secured Data Length = 0x6D (UDL =	Returns 0x006D	
	<u>0x80)</u>		
12	Test with UserDataLength = maximum length:	Returns 0x 79	
	0x79		
<u>12</u>	Test with Secured Data Length = maximum length	Returns 0x0079	
	for one envelope: 0x79 (UDL = 0x8C)		
13	Verify it is the first TPDU TLV:	Returns 0x05	
	Send a SMS PP with 2 TPDU TLV and inside two		
	different secured data lengths: 5 and 10		
14	Same test as 1 but with	Returns 0x2A	
	FORMATTED_SMS_PP_UPD		
13	Verify it is the first TPDU TLV:	Returns 0x0005	
	Send a SMS PP with 2 TPDU TLV and inside two		
	different secured data lengths: 5 and 10		
14	Test with secured data length = 0x7F (2	Returns 0x007F	
	concatenated envelopes are needed)		
<u>15</u>	Test with secured data length = 0x80 (2	Returns 0x0080	
	concatenated envelopes are needed)		
<u>16</u>		Returns 0x00FA	
	for 2 concatenated envelopes : 0xFA		
<u>17</u>	Test with FORMATTED SMS_PP_ENV	getValueByte returns 0x0040	
	Verify after call of the method the current TLV is		
	the TPDU TLV:		
	findTLV device identities, getSecuredDataLength		
	and then getValueByte to verify that the current		
	and then getValueByte to verify that the current TLV is the TPDU TLV		
	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering		
15	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with	Returns 0x2A	
	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD		
	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 1 but with	Returns 0x002A	
<u>18</u>	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 1 but with FORMATTED_SMS_PP_UPD	Returns 0x002A	
	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 1 but with FORMATTED_SMS_PP_UPD Same test as 3 but with		
18 16	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 1 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD	Returns 0x002A Returns 0x2A	
18 16	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 1 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD Same test as 2 but with Same test as 2 but with	Returns 0x002A	
18 16 19	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 1 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD	Returns 0x002A Returns 0x002A Returns 0x002A	
18 16	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 1 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 4 but with	Returns 0x002A Returns 0x2A	
18 16 19 17	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 1 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 2 but with FORMATTED_SMS_PP_UPD	Returns 0x002A Returns 0x002A Returns 0x002A Returns 0x10	
18 16 19 17	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 1 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 4 but with FORMATTED_SMS_PP_UPD Same test as 4 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD	Returns 0x002A Returns 0x002A Returns 0x002A	
18 16 19 17 20	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 1 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 4 but with FORMATTED_SMS_PP_UPD Same test as 4 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD	Returns 0x002A Returns 0x002A Returns 0x002A Returns 0x10 Returns 0x002A	
18 16 19 17	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 1 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 4 but with FORMATTED_SMS_PP_UPD Same test as 4 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD Same test as 5 but with FORMATTED_SMS_PP_UPD	Returns 0x002A Returns 0x002A Returns 0x002A Returns 0x10	
18 16 19 17 20 48	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 1 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 4 but with FORMATTED_SMS_PP_UPD Same test as 4 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD Same test as 5 but with FORMATTED_SMS_PP_UPD	Returns 0x002A Returns 0x002A Returns 0x002A Returns 0x10 Returns 0x002A Returns 0x002A	
18 16 19 17 20 48	and then getValueByte to verify that the current TLV is the TPDU TLV FORMATTED SMS PP UPD Triggering Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 1 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD Same test as 2 but with FORMATTED_SMS_PP_UPD Same test as 4 but with FORMATTED_SMS_PP_UPD Same test as 4 but with FORMATTED_SMS_PP_UPD Same test as 3 but with FORMATTED_SMS_PP_UPD Same test as 5 but with FORMATTED_SMS_PP_UPD	Returns 0x002A Returns 0x002A Returns 0x002A Returns 0x10 Returns 0x002A	

19	Same test as 6 but with- FORMATTED SMS PP UPD	Returns 0x10
<u>22</u>	Same test as 5 but with	Returns 0x0010
20	FORMATTED SMS PP UPD Same test as 7 but with	Returns 0x05
20	FORMATTED_SMS_PP_UPD	110 tall 10 ta
<u>23</u>	Same test as 6 but with FORMATTED SMS PP UPD	Returns 0x0010
21	Same test as 8 but with- FORMATTED SMS PP UPD	Returns 0x00
<u>24</u>	Same test as 7 but with FORMATTED SMS PP UPD	Returns 0x0005
22	Same test as 9 but with	Returns 0x33
<u>25</u>	FORMATTED_SMS_PP_UPD Same test as 8 but with	Returns 0x0000
23	FORMATTED SMS PP UPD Same test as 10 but with	Returns 0x 6C
26	FORMATTED_SMS_PP_UPD Same test as 9 but with	Returns 0x0033
24	FORMATTED_SMS_PP_UPD Same test as 11 but with	Returns 0x 6D
	FORMATTED_SMS_PP_UPD	
<u>27</u>	Same test as 10 but with FORMATTED SMS PP UPD	Returns 0x006C
25	Same test as 12 but with FORMATTED_SMS_PP_UPD	Returns 0x 79
<u>28</u>	Same test as 11 but with FORMATTED_SMS_PP_UPD	Returns 0x006D
26	Same test as 13 but with- FORMATTED_SMS_PP_UPD	Returns 0x05
<u>29</u>	Same test as 12 but with FORMATTED SMS PP UPD	Returns 0x0079
27	Same test as 4 but with FORMATTED_SMS_CB	Returns 0x10
30	Same test as 13 but with	Returns 0x0005
31	FORMATTED SMS PP UPD Test with secured data length = 0x7F (2	Returns 0x007F
	concatenated envelopes are needed)	
<u>32</u>	Test with secured data length = 0x80 (2 concatenated envelopes are needed)	Returns 0x0080
<u>33</u>	<u>Test with secured data length = maximum length</u> for 2 concatenated envelopes : 0xFA	Returns 0x00FA
<u>34</u>	Test with FORMATTED_SMS_PP_UPD_ Verify after call of the method the current TLV is_	getValueByte returns 0x0040
	the TPDU TLV:	
	findTLV device identities, getSecuredDataLength and then getValueByte to verify that the current	
	TLV is the TPDU TLV FORMATTED SMS CB Triggering	
28	Same test as 5 but with FORMATTED SMS_CB	Returns 0x10
	Same test as 4 but with FORMATTED_SMS_CB	Returns 0x0010
29	Same test as 6 but with FORMATTED_SMS_CB	Returns 0x10
	Same test as 5 but with FORMATTED_SMS_CB	Returns 0x0010
30	Same test as 7 but with FORMATTED_SMS_CB	Returns 0x05
<u>37</u>	Same test as 6 but with FORMATTED SMS CB	Returns 0x0010
	Same test as 8 but with FORMATTED_SMS_CB	Returns 0x00
	Same test as 7 but with FORMATTED SMS CB	Returns 0x0005
32	Same test as 9 but with FORMATTED_SMS_CB	Returns 0x33
3 <u>9</u> 33	Same test as 8 but with FORMATTED_SMS_CB Same test as 12 but with maximum length: 0x42.	Returns 0x0000 Returns 0x 42
	and FORMATTED_SMS_CB	
<u>40</u>	Same test as 9 but with FORMATTED_SMS_CB	Returns 0x0033
34	Test with FORMATTED_SMS_PP_ENV	getValueByte returns 0x40
1	Verify after call of the method the current TLV is	
	the TPDU TLV: findTLV device identities, getSecuredDataLength	
	and then getValueByte to verify that the current TLV is the TPDU TLV	
	TEVIORIO II DO I EV	

35	Test with FORMATTED_SMS_CB	getValueByte returns 0x58
00	Verify after call of the method the current TLV is	get value by te returns 6x66
	the Cell Broadcast Page TLV:	
	findTLV device identities, getSecuredDataLength	
	and then getValueByte to verify that the current-	
	TLV is the Cell Broadcast Page TLV	
<u>41</u>	Same test as 12 but with maximum secured data	Returns 0x0042
	length: 0x42, and FORMATTED_SMS_CB	
36	Send an envelope SMS CB,	ToolkitException-
	getSecuredDataLength	UNAVAILABLE ELEMENT
42	Test with FORMATTED_SMS_CB	getValueByte returns 0x00
	Verify after call of the method the current TLV is	
	the Cell Broadcast Page TLV:	
	findTLV device identities, getSecuredDataLength	
	and then getValueByte to verify that the current	
	TLV is the Cell Broadcast Page TLV	
	Error tests	
43	Send an envelope SMS CB,	ToolkitException
	getSecuredDataLength	UNAVAILABLE ELEMENT
37	Send an envelope SMS PP unformatted	ToolkitException -
		UNAVAILABLE_ELEMENT
44	Send an envelope SMS PP unformatted	ToolkitException
		UNAVAILABLE ELEMENT

6.2.4.3.4 Test Coverage

This method has only been tested with call control and the tests shall be improved during 03.48 tests.

CRR number	Test case number
N1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
	11, 12, 13
N2	13
N3	6, 7
N4	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
	11, 12, 13, 25
N5	14, 15, 16, 17, 18, 19, 20,
	21, 22, 23, 24, 25, 26
N6	27, 28, 29, 30, 31, 32, 33 34 35
N7	34
N8	35
C1	36
C2	37
<u>N1</u>	<u>1 to 42</u>
<u>N2</u>	<u>13, 30</u>
<u>N3</u>	6, 7, 23, 24, 37, 38
<u>N4</u>	<u>1 to 17</u>
<u>N5</u>	<u>18 to 34</u>
<u>N6</u>	35 to 42
<u>N7</u>	<u>17</u>
<u>N8</u>	<u>34</u>
<u>N9</u>	<u>42</u>
<u>C1</u>	<u>43</u>
<u>C2</u>	<u>44</u>

6.2.4.4 Method getSecuredDataOffset

Test Area Reference: API_2_ENH_GSDO

6.2.4.4.1 Conformance Requirement:

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

6.2.4.4.1.1 Normal execution

CRRN1: The method shall return the offset of the secured data first byte contained in a SMS TPDU TLV.

CRRN2: The offset is from the first SMS TPDU TLV.

CRRN3: The method can be used if the event is EVENT_FORMATTED_SMS_PP_ENV and if the SMS TP-UD is formatted according to GSM03.48.TS 23.048 [8].

CRRN4: The method can be used if the event is EVENT_FORMATTED_SMS_PP_UPD and if the SMS TP-UD is formatted according to GSM03.48.TS 23.048 [8].

CRRN5: The method can be used if the event is EVENT_FORMATTED_SMS_CB and if the Cell Broadcast Page is formatted according to <u>TS 23.048</u> [8].

CRRN6: If the method is successful and if the event is EVENT_FORMATTED_SMS_PP_ENV, the selected TLV should be the SMS TPDU TLV.

CRRN7: If the method is successful and if the event is <u>EVENT_FORMATTED_SMS_PP_UPD</u>, the selected <u>TLV</u> should be the <u>SMS_TPDU_TLV</u>.

<u>CRRN8: If the method is successful and if the event is EVENT_FORMATTED_SMS_CB</u>, the selected TLV should be the Cell Broadcast Page TLV.

Parameters error

No requirements CRNN9: If the Secured Data length is zero the value returned shall be the offset of the first byte following the TS 23.048 [8] Command Packet structure.

6.2.4.4.1.2 Context errors

CRRC1: The method shall thrown ToolkitException (UNAVAILABLE_ELEMENT) in case of unavailable SMS TPDU TLV element.

CRRC2: The method shall thrown ToolkitException (UNAVAILABLE_ELEMENT) in case of missing Secured Data.wrong data format

6.2.4.4.2 Test suite files

Specific triggering:

- FORMATTED SMS CB
- UNFORMATTED SMS CB
- FORMATTED SMS PP UPD
- UNFORMATED SMS PP ENV

- For Formatted triggering if CC/RC/DS is used, the security parameters are the one used for downloading applications.

Test Script: API_2_ENH_GSDO_1.scr

Test Applet: API_2_ENH_GSDO_1.java

Load Script: API_2_ENH_GSDO_1.ldr

Cleanup Script: API_2_ENH_GSDO_1.clr

Parameter File: API_2_ENH_GSDO_1.par

6.2.4.4.3 Test Procedure

ld	Description	API Expectation	APDU Expectation
	FORMATTED SMS PP ENV triggering	1 1 2 3	
4	Test with TP-OA length of 2	Returns 0x21	
1	Test with TP-OA length of 2 and RC/CC/DS length is 0	Returns 0x21	
2	Test with TP-OA length of 6	Returns 0x23	
2	Test with TP-OA length of 6 and RC/CC/DS length is 0	Returns 0x23	
3	Test with TP-OA length of 12	Returns 0x26	
<u>3</u>	Test with TP-OA length of 12 and RC/CC/DS length is 0	Returns 0x26	
4	Test with RC/CC/DS length of 0	Returns 0x21	
4	Test with RC/CC/DS length of 0 and TP-OA length is 2	Returns 0x21	
5	Test with RC/CC/DS length of 8	Returns 0x29	
<u>5</u>	Test with RC/CC/DS length of 8 and TP-OA length is 2		
6	Send a SMS PP with 2 TPDU TLV and inside two different secured data offsets	Returns 0x24 (the first offset)	
7	Same test as 1 but with- FORMATTED_SMS_PP_UPD	Returns 0x21	
7	Same test as 1 but without any secured data	Returns 0x21	
8	Same test as 2 but with FORMATTED_SMS_PP_UPD	Returns 0x23	
<u>8</u>	Test with FORMATTED_SMS_PP ENV	Returns 0x40	
	Verify after call of the method the current TLV is		
	the TPDU TLV: findTLV device identities, getSecuredDataOffset		
	and then getValueByte to verify that the current		
	TLV is the TPDU TLV		
9	Same test as 3 but with- FORMATTED SMS PP UPD	Returns 0x26	
9	Same test as 1, but with a concatenated SMS (2 Short Messages and maximum Secured Data	Returns 0x21	
	<u>Length = 0x00FA)</u> FORMATTED SMS PP UPR triggering		
10	Same test as 4 but with	Returns 0x21	
	FORMATTED_SMS_PP_UPD		
	Same test as 1 but with FORMATTED_SMS_PP_UPD	Returns 0x21	
	Same test as 5 but with- FORMATTED_SMS_PP_UPD	Returns 0x29	
11	Same test as 2 but with FORMATTED_SMS_PP_UPD	Returns 0x23	
12	Same test as 6 but with FORMATTED_SMS_PP_UPD	Returns 0x24 (the first offset)	
<u>12</u>	Same test as 3 but with FORMATTED_SMS_PP_UPD	Returns 0x26	
<u>13</u>	Same test as 4 but with FORMATTED SMS PP UPD	Returns 0x21	
14	Same test as 5 but with FORMATTED SMS PP UPD	Returns 0x29	
<u>15</u>	Same test as 6 but with	Returns 0x24 (the first offset)	
13	FORMATTED_SMS_PP_UPD Test with FORMATTED_SMS_PP_ENV	Returns 0x40	
10	Verify after call of the method the current TLV is	INCIAITIS ONTO	
	the TPDU TLV:		
	findTLV device identities, getSecuredDataOffset		
	and then getValueByte to verify that the current- TLV is the TPDU TLV		
<u>16</u>	Same test as 7 but with FORMATTED_SMS_PP_UPD	Returns 0x21	
<u>17</u>	Test with FORMATTED_SMS_PP UPD	Returns 0x40	
	Verify after call of the method the current TLV is		

	Li TDDUTIN	
	the TPDU TLV:	
	findTLV device identities, getSecuredDataOffset	
	and then getValueByte to verify that the current	
	TLV is the TPDU TLV	
14	Same test as 4 but with FORMATTED_SMS_CB	Returns 0x16
<u>18</u>		Returns 0x21
	Short Messages and maximum Secured Data	
	$\underline{\text{Length} = 0x00FA})$	
	FORMATTED SMS CB triggering	
19	Same test as 4 but with FORMATTED_SMS_CB	Returns 0x16
15	Same test as 5 but with FORMATTED_SMS_CB	Returns 0x1E
20	Same test as 5 but with FORMATTED SMS CB	Returns 0x1E
16	Test with FORMATTED SMS CB	Returns 0x58
	Verify after call of the method the current TLV is	
	the Cell Broadcast Page TLV:	
	findTLV device identities, getSecuredDataOffset	
	and then getValueByte to verify that the current	
	TLV is the Cell Broadcast Page TLV	
21	Same test as 7 but with FORMATTED SMS CB	Returns 0x16
17	Send an UNFORMATTED SMS CB envelope,	ToolkitException-
	getSecuredDataOffset	UNAVAILABLE_ELEMENT
22	Test with FORMATTED_SMS_CB	Returns 0x00
	Verify after call of the method the current TLV is	
	the Cell Broadcast Page TLV:	
	findTLV device identities, getSecuredDataOffset	
	and then getValueByte to verify that the current	
	TLV is the Cell Broadcast Page TLV	
	UNFORMATTED Triggering	
23	Send an UNFORMATTED SMS CB envelope,	ToolkitException
	<u>getSecuredDataOffset</u>	UNAVAILABLE ELEMENT
18	Send an UNFORMATTED SMS PP envelope	ToolkitException-
	·	UNAVAILABLE_ELEMENT
24	Send an UNFORMATTED SMS PP envelope,	<u>ToolkitException</u>
	getSecuredDataOffset	UNAVAILABLE_ELEMENT
19	Send an FORMATTED SMS-PP envelope with no-	Returns 0x21
	secured data, getSecuredDataOffset	

6.2.4.4.4 Test Coverage

This method has only been tested with call control and the tests shall be improved during 03.48 tests.

CRR number	Test case number
N1	1, 2, 3, 4, 5, 6, 19
N2	6, 12
N3	1, 2, 3, 4, 5, 6, 19
N 4	7, 8, 9, 10, 11
N5	14,15
N6	13
N7	16
C1	17
C2	18
<u>N1</u>	<u>1 to 22.</u>
<u>N2</u>	<u>6, 15.</u>
<u>N3</u>	<u>1 to 9.</u>
<u>N4</u>	<u>10 to 18.</u>
<u>N5</u>	<u>19, 20, 21, 22</u>
<u>N6</u>	<u>8</u>
<u>N7</u>	<u>17</u>
<u>N8</u>	<u>22</u>
<u>N9</u>	<u>7, 16, 21.</u>
<u>C1</u> <u>C2</u>	<u>23</u>
<u>C2</u>	<u>24</u>

6.2.4.5 Method getTheHandler

Test Area Reference: API_2_ENH_GTHD

6.2.4.5.1 Conformance Requirements

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

6.2.4.5.1.1 Normal execution

CRRN1: The method shall return the single system instance of the EnvelopeHandler class.

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

Parameters error

No requirements

6.2.4.5.1.2 Context errors

CRRC1: The method shall thrown ToolkitException (HANDLER_NOT_AVAILABLE) if the handler is busy.

6.2.4.5.2 Test suite files

Test Script: API_2_ENH_GTHD_1.scr

Test Applet: API_2_ENH_GTHD_1.java

Load Script: API_2_ENH_GTHD_1.ldr

Cleanup Script: API_2_ENH_GTHD_1.clr

Parameter File: API_2_ENH_GTHD_1.par

6.2.4.5.3 Test Procedure

ld	Description	API Expectation	APDU Expectation
1	getTheHandler twice	The returned objects shall be the	
		same	
2	Verify that getTheHandler returns an-	The reference returned shall be an	
	EnvelopeHandler	EnvelopeHandler (checkcast)	
	GetTheHandler		
2	Verify that getTheHandler returns an	The reference returned shall be an	
	<u>EnvelopeHandler</u>	EnvelopeHandler (check cast)	
	<u>GetTheHandler</u>		
3	Verify the returned value is not null	The reference returned shall not be	
	GetTheHandler	null.	

6.2.4.5.4 Test Coverage

CRR number	Test case number
N1	1, 2, 3
N2	To be checked in
	Framework tests and
	insert here cross
	reference
C1	To be checked in
	Framework tests and
	insert here cross
	reference

6.2.4.6 Method getTPUDLOffset

Test Area Reference: API_2_ENH_GTPO

6.2.4.6.1 Conformance Requirement:

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

6.2.4.6.1.1 Normal execution

CRRN1: The method shall return the TPUDL offset in a SMS TPDU TLV.

CRRN2: The offset is from the first SMS TPDU TLV.

CRRN3: The method can be used if the event is EVENT_FORMATTED_SMS_PP_ENV.

CRRN4: The method can be used if the event is EVENT_FORMATTED_SMS_PP_UPD.

CRRN5: The method can be used if the event is EVENT_UNFORMATTED_SMS_PP_ENV.

CRRN6: The method can be used if the event is EVENT_UNFORMATTED_SMS_PP_UPD.

CRRN7: If the method is successful, the selected TLV should be the SMS TPDU TLV.

Parameters error

No requirements

6.2.4.6.1.2 Context errors

 $CRRC1: The \ method \ shall \ thrown \ ToolkitException \ (UNAVAILABLE_ELEMENT) \ in \ case \ of \ unavailable \ SMS \ TPDU \ TLV \ element.$

CRRC2: The method shall thrown ToolkitException (UNAVAILABLE_ELEMENT) if the TPUDL field does not exist.

6.2.4.6.2 Test suite files

Specific triggering:

FORMATTED SMS PP UPD
UNFORMATTED SMS PP UPD

- FORMATTED SMS PP UPD

- UNFORMATTED SMS PP UPD

UNFORMATTED SMS PP ENV

— <u>UNFORMATTED</u> SMS CB

Test Script: API_2_ENH_GTPO_1.scr

Test Applet: API_2_ENH_GTPO_1.java

Load Script: API_2_ENH_GTPO_1.ldr

Cleanup Script: API_2_ENH_GTPO_1.clr

Parameter file: API_2_ENH_GTPO_1.par

6.2.4.6.3 Test procedure

ld	Description	API Expectation	APDU Expectation
Iu	FORMATTED SMS PP ENV triggering	AFI Expectation	APDO Expectation
1	Test with TP-OA length of 2	Returns 0x0D	
-	Test with TP-OA length of 6		
		Returns 0x0F Returns 0x12	
4	Test with TP-OA length of 12 Send a SMS PP with 2 TPDU TLV and inside two	Returns 0x12 (the first offset)	
4		Returns 0x10 (the first offset)	
4	different UDL offsets Send a SMS PP with 2 TPDU TLV and inside two	Deturns Ord O (the first offeet)	
4	different UDL offsets	Returns 0x10 (the first offset)	
_	Same test as 1 but with	Returns 0x0D	
5	FORMATTED_SMS_PP_UPD	Returns 0x0D	
_	Same test as 1, but with a concatenated SMS (2	Deturne OvOD	
<u>5</u>	Short Messages and maximum Secured Data	Returns 0x0D	
	Length = 0x00FA)		
6	Same test as 2 but with	Returns 0x0F	
0	FORMATTED_SMS_PP_UPD	NOTATION OF THE PROPERTY OF TH	
6	Verify after call of the method the current TLV is the	Peturne 0v40	
O	TPDU TLV:	INGIUITIS UX40	
	findTLV device identities, getTPUDLOffset and then		
	getValueByte to verify that the current TLV is the		
	TPDU TLV		
	FORMATTED SMS PP UPD triggering		
7	Same test as 3 but with	Returns 0x12	
· ·	FORMATTED SMS PP UPD	Trotaine oxiz	
7	Same test as 1 but with	Returns 0x0D	
_	FORMATTED_SMS_PP_UPD		
8	Same test as 4 but with	Returns 0x10 (the first offset)	
	FORMATTED_SMS_PP_UPD	, ,	
8	Same test as 2 but with	Returns 0x0F	
	FORMATTED_SMS_PP_UPD		
9	Same test as 3 but with	Returns 0x12	
	FORMATTED SMS PP UPD		
9	Same test as 1 but with	Returns 0x0D	
	UNFORMATTED_SMS_PP_UPD		
10	Same test as 2 but with	Returns 0x0F	
	UNFORMATTED_SMS_PP_UPD		
<u>10</u>	Same test as 4 but with	Returns 0x10 (the first offset)	
	FORMATTED_SMS_PP_UPD		
11	Same test as 3 but with	Returns 0x12	
	UNFORMATTED_SMS_PP_UPD		
<u>11</u>	Same test as 7, but with a concatenated SMS (2	Returns 0x0D	
	Short Messages and maximum Secured Data		
	Length = 0x00FA)		
	UNFORMATTED SMS PP UPD triggering		
12	Same test as 4 but with	Returns 0x12 (the first offset)	
	UNFORMATTED_SMS_PP_UPD		
<u>12</u>	Same test as 1 but with	Returns 0x0D	
	UNFORMATTED_SMS_PP_UPD		

40	Same test as 1 but with	Returns 0x0D
13		Returns UXUL
	UNFORMATTED_SMS_PP_ENV	
<u>13</u>	Same test as 2 but with	Returns 0x0F
	UNFORMATTED SMS PP UPD	
14	Same test as 2 but with	Returns 0x0F
	UNFORMATTED_SMS_PP_ENV	
14	Same test as 3 but with	Returns 0x12
	UNFORMATTED SMS PP UPD	
15	Same test as 3 but with	Returns 0x12
	UNFORMATTED SMS PP ENV	
15	Same test as 4 but with	Returns 0x12 (the first offset)
	UNFORMATTED_SMS_PP_UPD	- 1010 1110 0X12 (1110 1110 1110 11)
16	Same test as 4 but with	Returns 0x10 (the first offset)
'0	UNFORMATTED_SMS_PP_ENV	restarts over (the mot effect)
16	Same test as 12, but with a concatenated SMS (2	Returns 0x0D
10	Short Messages and maximum User Data Length =	INCIGITIS OXOD
	0x010C)	
-		
4-7	UNFORMATTED SMS PP ENV triggering	D + 0 0D
17	Same test as 1 but with	Returns 0x0D
	UNFORMATTED_SMS_PP_ENV	
<u>18</u>	Same test as 2 but with	Returns 0x0F
	<u>UNFORMATTED_SMS_PP_ENV</u>	
<u>19</u>	Same test as 3 but with	Returns 0x12
	UNFORMATTED SMS PP ENV	
17	Verify after call of the method the current TLV is the	Returns 0x40
	TPDU TLV:	
	findTLV device identities, getTPUDLOffset and then	
	getValueByte to verify that the current TLV is the	
	TPDU TLV	
18	Send an envelope SMS CB, getTPUDLOffset	ToolkitException-
		UNAVAILABLE_ELEMENT
20	Same test as 4 but with	Returns 0x10 (the first offset)
1	UNFORMATTED SMS PP_ENV	
21	Same test as 17, but with a concatenated SMS (2	Returns 0x0D
=-	Short Messages and maximum User Data Length =	
	0x010C)	
-	SMS CB triggering	
22	Send an envelope SMS CB, getTPUDLOffset	ToolkitException
	Cond an envelope office ob, get it obtoliset	UNAVAILABLE ELEMENT
		OTAVAITADET TETRIFIAT

6.2.4.6.4 Test Coverage

CRR number	Test case number
N1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
	11, 12, 13, 14, 15, 16, 17
N2	4
<u>N1</u>	<u>1 to 21.</u>
<u>N2</u>	<u>4, 10, 15, 20.</u>
N3	1, 2, 3, 4, 17
N4	5, 6, 7, 8
N5	13, 14, 15, 16
<u>N3</u>	<u>1, 2, 3, 4, 5, 6</u>
<u>N4</u>	<u>7, 8, 9, 10, 11,</u>
<u>N5</u>	<u>12, 13, 14, 15, 16</u>
N6	9, 10, 11, 12
N7	17
C1	18
<u>N6</u>	<u>17, 18, 19, 20, 21</u>
N7 C1 C2	<u>6</u>
<u>C1</u>	<u>22</u>
C2	Don't no how to test
<u>C2</u>	Not applicable

6.2.4.7 Method getLength

Test Area Reference: API_2_ENH_GLEN

6.2.4.7.1 Conformance Requirement

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

public short getLength()

throws ToolkitException

6.2.4.7.1.1 Normal execution

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

Parameter Error

No requirements

6.2.4.7.1.2 Context errors

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

6.2.4.7.2 Test Suite files

Specific triggering: None

Test Script: API_2_ENH_GLEN_1.scr

Test Applet: API_2_ENH_GLEN_1.java

Load Script: API_2_ENH_GLEN_1.ldr

Cleanup Script: API_2_ENH_GLEN_1.clr

Parameter File: API_2_ENH_GLEN_1.par

6.2.4.7.3 Test Procedure

ld	Description	API Expectation	APDU Expectation
1	Send an envelope SMS PP with BER length of - 0x31	Result of getLength() is 0x0031	•
2	Send an envelope SMS PP with BER length of - 0x7F	Result of getLength() is 0x007Fh	
3	Send an envelope SMS PP with BER length of - 81 80	Result of getLength() is 0x0080h	
4	Send an envelope SMS PP with BER length of 81 FC	Result of getLength() is 0x00FCh	
<u>4</u>	Send an envelope SMS PP with BER length of 81 FC (maximum length for a single SMS)	Result of getLength() is 0x00FCh	
<u>5</u>	Send formatted SMS with BER length of 0x00FF, using 2 concatenated SMS	Result of getLength() is 0x00FFh	
<u>6</u>	Send formatted SMS with BER length of 0x0100, using 2 concatenated SMS	Result of getLength() is 0x0100h	
7	Send formatted SMS with maximum user data length (0x10D) (BER length:0x012F), using 2 concatenated SMS	Result of getLength() is 0x012Fh	

6.2.4.7.4 Test Coverage

CRR number Test case numbe	
N1	1, 2, 3, 4
<u>N1</u>	<u>1, 2, 3, 4, 5, 6, 7</u>
C1	Does not apply for
	EnvelopeHandler

6.2.4.8 Method copy

Test Area Reference: API_2_ENH_COPY_BSS

6.2.4.8.1 Conformance Requirement

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

6.2.4.8.1.1 Normal execution

CRRN1: copies the simple TLV list contained in the handler to the destination byte array.

CRRN2: returns dstOffset + dstLength.

6.2.4.8.1.2 Parameter errors

CRRP1: if dstBuffer is null a NullPointerException is thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative, an ArrayIndexOutOfBoundsException is thrown.

CRRP3: if dstLength is greater than the length of the simple TLV List, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

6.2.4.8.1.3 Context errors

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

6.2.4.8.2 Test Suite files

Specific triggering: None

Test Script: API_2_ENH_COPY_BSS_1.scr

Test Applet: API_2_ENH_COPY_BSS_1.java

Load Script: API_2_ENH_COPY_BSS_1.ldr

Cleanup Script: API_2_ENH_COPY_BSS_1.elr

Parameter File: API_2_ENH_ COPY _ BSS _1.par API_2_ENH_ COPY_BSS_1.clr

Parameter File: API_2_ENH_COPY_BSS_1.par

6.2.4.8.3 Test Procedure

ld	Description	API Expectation	APDU Expectation
1	NULL as parameter to dstBuffer	NullPointerException is thrown	•
	•	·	
2	dstOffset ≥ dstBuffer.length	ArrayIndexOutOfBoundsException is	
	dstBuffer.length = 5	thrown	
	<pre>dstOffset = 5 dstLength = 1</pre>		
3	dstOffset < 0	ArrayIndexOutOfBoundsException is	
3	dstBuffer.length = 5	thrown	
	dstOffset = -1	unown	
	dstLength = 1		
4	dstLength > dstBuffer.length	ArrayIndexOutOfBoundsException is	
	<pre>dstBuffer.length = 5 dstOffset = 0</pre>	thrown	
	dstollset = 0 dstLength = 6		
5	DstOffset + dstLength > dstBuffer.length	ArrayIndexOutOfBoundsException is	
	DstBuffer.length = 5	thrown	
	DstOffset = 3		
<u></u>	DstLength = 3	1 1 1 0 1015	
6	dstLength < 0	ArrayIndexOutOfBoundsException is	
	<pre>dstBuffer.length = 5 dstOffset = 0</pre>	thrown	
	dstLength = -1		
7	DstLength > length of the simple TLV list	ToolkitException.OUT_OF_TLV_BO	
	DstBuffer.length = 48	UNDARIES is thrown	
	DstOffset = 0		
_	DstLength = 48	Desult of conv() is 0V0047	
8	Successful call, dstBuffer is the whole buffer DstBuffer.length = 47	Result of copy() is 0X0047	
	DstOffset = 0		
	DstLength = 47		
9	Compare the buffer	Result of arrayCompare() is 0	
10	Successful call, dstBuffer is part of a buffer	Result of copy() is 0X0032	
	DstBuffer.length = 50		
	<pre>dstOffset = 3 dstLength = 47</pre>		
11	Compare the whole buffer	Result of arrayCompare() is 0	
12	Successful call, dstBuffer is part of a buffer	Result of copy() is 0X0009	
	dstBuffer.length = 15	130	
	dstOffset = 3		
12	dstLength = 6 Compare the whole buffer	Result of arrayCompare() is 0	
13 14	Successful call, dstBuffer is part of a buffer	Result of copy() is 0X0104	
'4	dstBuffer.length = 260	Tresuit of copy() is unutual	
	dstOffset = 257		
	dstLength = 3		
15	Compare the whole buffer	Result of arrayCompare() is 0	
16	Successful call, copy with length =0	Result of copy() is 0x104	
	dstBuffer.length = 260 dstOffset = 260		
	dstOffset = 260 dstLength = 0		
	Send a Formatted SMS PP with the maximum		
	user data length = 0x010D, using 2		
	concatenated envelopes		
<u>17</u>	Successful call, copy with length =299	Result of copy() is 0x12B	
	dstBuffer.length = 299		
	dstOffset = 0		
	dstLength = 299		

6.2.4.8.4 Test Coverage

CRR number	Test case number
N1	9, 11, 13, 15
N2	8, 10, 12, 14, 16
<u>N2</u>	<u>8, 10, 12, 14, 16, 17</u>
P1	1
P2	2, 3, 4, 5, 6
P3	7
C1	Does not apply for
	EnvelopeHandler

6.2.4.9 Method findTLV

Test Area Reference: API_2_ENH_FINDBB

6.2.4.9.1 Conformance Requirement

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

6.2.4.9.1.1 Normal execution

Looks for the indicated occurrence of a TLV element from the beginning of the TLV list (handler buffer):

CRRN1: the method is successful if the required occurrence exists then the corresponding TLV becomes current.

CRRN2: if the method is successful then it returns TLV_FOUND_CR_SET when Comprehension Required flag is set.

CRRN3: if the method is successful then it returns TLV_FOUND_CR_NOT_SET when Comprehension Required flag is not set.

CRRN4: if the required occurrence of the TLV element does not exist, the current TLV is no longer defined and -TLV_NOT_FOUND is returned.

CRRN5: The search method is comprehension required flag independent.

6.2.4.9.1.2 Parameter errors

CRRP1: if an input parameter is not valid (e.g. occurrence = 0) an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException BAD_INPUT_PARAMETER.

6.2.4.9.1.3 Context errors

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

6.2.4.9.2 Test Suite files

Specific triggering: None

Test Script: API_2_ENH_ENH_FINDBB_1.scr

Test Applet: API_2_ENH_ENH_FINDBB_1.java

Load Script: API_2_ENH_ENH_FINDBB_1.ldrAPI_2_ENH_FINDBB_1.java

Load Script: API 2 ENH FINDBB 1.ldr

Cleanup Script: API_2_ENH_FINDBB_1.clr

Parameter File: API_2_ENH_FINDBB_1.par

6.2.4.9.3 Test Procedure

ld	Description	API Expectation	APDU Expectation
	Trig the applet with SMS PP including one more		
	tag 02 and one TAG 04		
1	Invalid input parameter	ToolkitException.BAD_INPUT_P	
	Occurrence = 0	ARAMETER is thrown	
2	Search 1st TLV	Result is TLV_FOUND_CR_SET	
	Tag = 02h Occurrence = 1		
3	Call the getValueLength() method	Result is 0x02	
4	Search 2nd TLV	Result is TLV FOUND CR SET	
-	Tag = 06h	INESUR IS TEV_FOOND_ON_OET	
	Occurrence = 1		
5	Call the getValueLength() method	Result is 0x05h	
6	Select a TLV (tag 02h)		
	Search a wrong tag	Result is TLV_NOT_FOUND	
	Tag = 03h		
	Occurrence = 1		
7	Call the getValueLength() method	ToolkitException.UNAVAILABLE	
	0	_ELEMENT is thrown.	
8	Search a tag with wrong occurrence Tag = 02h	Result is TLV_NOT_FOUND	
	Occurrence = 3		
9	Call the getValueLength() method	ToolkitException.UNAVAILABLE	
	Gan and gott and zongmy mounds	ELEMENT is thrown.	
10	Search the TLV	Result is	
	Tag = 02h	TLV_FOUND_CR_NOT_SET	
	Occurrence = 2		
11	Search the TLV	Result is	
	Tag = 04h	TLV_FOUND_CR_NOT_SET	
12	Occurrence = 1 Search tag 81h	Result is TLV_FOUND_CR_SET	
72	Tag = 86h	NOSCILIS TEV_I COND_CIN_CET	
	Occurrence = 1		
<u>12</u>	Search tag 86h	Result is TLV_FOUND_CR_SET	
	Tag = 86h		
	Occurrence = 1		
13	Search tag 84h	Result is	
	Tag = 84h Occurrence = 1	TLV_FOUND_CR_NOT_SET	
L	Occurrence = 1		

6.2.4.9.4 Test Coverage

CRR number	Test case number
N1	3, 5
N2	2, 4
N3	10, 11
N4	6, 7,8, 9
N5	12, 13
P1	1
C1	Does not apply for EnvelopeHandler

6.2.4.10 Method getValueLength

Test Area Reference: API_2_ENH_GVLE-

6.2.4.10.1 Conformance Requirement

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

6.2.4.10.1.1 Normal execution

CRRN1: gets and returns the binary length of the value field for the last TLV element which has been found in the handler.

6.2.4.10.1.2 Parameter errors

No requirements.

6.2.4.10.1.2 Context errors

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

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.2.4.10.2 Test Suite files

Specific triggering: None

Test Script: ___API_2_ENH_GVLE_1.scr

Test Applet: API_2_ENH_GVLE_1.java

Load Script: API_2_ENH_GVLE_1.ldr

Cleanup Script: API_2_ENH_GVLE_1.clr

Parameter File: API_2_ENH_GVLE_1.java

Load Script: API_2_ENH_GVLE_1.ldr

Cleanup Script: API_2_ENH_GVLE_1.ldr

Cleanup Script: API_2_ENH_GVLE_1.ldr

Parameter File: API_2_ENH_GVLE_1.par

6.2.4.10.3 Test Procedure

ld	Description	API Expectation	APDU Expectation
	Fill the SMS PP with TLV: Tag 33, Length C8		
1	getValueLength()	ToolkitException.UNAVAILABLE _ELEMENT is thrown	
2	Search TLV 02h		
	getValueLength()	Result is 0X0002	
3	Search TLV 0Bh		
	getValueLength()	Result is 0X0024	
4	Search TLV 33h		
	getValueLength()	Result is 0X00C8	
	Send Formatted SMS PP with the maximum		
	user data length = 0x010D, using 2		
	concatenated envelopes		
<u>5</u>	Search SMS TPDU TAG		
	<pre>getValueLength()</pre>	Result is 0X0120	

6.2.4.10.4 Test Coverage

	CRR number		Test case number	
N1		2, 3, 4		
·	<u>N1</u>		<u>2, 3, 4, 5</u>	
	C1		Does not apply for	
			EnvelopeHandler	
	C2		1	

6.2.4.11 Method getValueByte

Test Area Reference: API_2_ENH_GVBYS-

6.2.4.11.1 Conformance Requirement

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

6.2.4.11.1.1 Normal execution

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

6.2.4.11.1.2 Parameter errors

CRRP1: if valueOffset is out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

6.2.4.11.1.3 Context errors

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

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.2.4.11.2 Test Suite files

Specific triggering: None

Test Script: _____API_2_ENH_GVBYS_1.scr

Test Applet: ____API_2_ENH_GVBYS_1.java

Load Script: ___API_2_ENH_GVBYS_1.dr

Cleanup Script: ___API_2_ENH_GVBYS_1.elr

Parameter File: ___API_2_ENH_GVBYS_1.java

Load Script: ___API_2_ENH_GVBYS_1.ldr

Cleanup Script: ___API_2_ENH_GVBYS_1.ldr

Cleanup Script: ___API_2_ENH_GVBYS_1.clr

Parameter File: ___API_2_ENH_GVBYS_1.par

6.2.4.11.3 Test Procedure

ld	Description	API Expectation	APDU Expectation
	Fill the SMS PP with TLV: Tag 33, Length C8		
	Value 01 02		
1	getValueByte(0)	ToolkitException.UNAVAILABLE	
		_ELEMENT is thrown	
2	Search TLV 02h		
	getValueByte(2)	ToolkitException.OUT_OF_TLV_	
		BOUNDARIES is thrown	
3	Search TLV 02h		
	getValueByte(1)	Result is 0x81	
4	Search TLV 02h (Device Identities TLV)		
	<pre>getValueByte(0)</pre>	Result is 83h (Source)	
5	Search TLV 33h		
	getValueByte(7E)	Result is 0x7F	
6	Search TLV 33h		
	getValueByte(80)	Result is 0x81	
7	getValueByte(7F)	Result is 0x80	
8	Search TLV B3h		
	getValueByte(C7)	Result is 0xC8	
	Send Formatted SMS PP with the maximum		
	user data length = 0x010D, using 2		
	concatenated envelopes		
9	Search SMS TPDU TAG		
	<pre>getValueByte(0x011F)</pre>	Result is 0xFA	

6.2.4.11.4 Test Coverage

	CRR number		Test case number	
N1		3, 4, 5, 6, 7, 8		
	<u>N1</u>		<u>3, 4, 5, 6, 7, 8, 9</u>	
	P1		2	
	C1		Does not apply for	
			EnvelopeHandler	
	C2		1	

6.2.4.12 Method copyValue

Test Area Reference: API_2_ENH_CPYVS_BSS-

6.2.4.12.1 Conformance Requirement

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

6.2.4.12.1.1 Normal execution

CRRN1: copies a part of the last TLV element which has been found, into a destination. buffer.

 $CRRN2: returns\hbox{--} dstOffset + dstLength.$

6.2.4.12.1.2 Parameter errors

CRRP1: if dstBuffer is null NullPointerException is thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative ArrayIndexOutOfBoundsException is thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

6.2.4.12.1.3 Context errors

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

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.2.4.12.2 Test Suite files

Specific triggering: None

Test Script:	API_2_ENH_CPYVS_BSS_1.scr
Test Applet:	API_2_ENH_CPYVS_BSS_1.java
Load Script:	API_2_ENH_CPYVS_BSS_1.ldr
Cleanup Script:	API_2_ENH_CPYVS_BSS_1.clr
Parameter File:	API_2_ENH_CPYVS_BSS_1.java
Load Script:	API_2_ENH_CPYVS_BSS_1.ldr
Cleanup Script:	API_2_ENH_CPYVS_BSS_1.clr
Parameter File:	API_2_ENH_CPYVS_BSS_1.par

6.2.4.12.3 Test procedure

ld	Description	API Expectation	APDU Expectation
1	Search TLV 02h		-
	copyValue() with a null dstBuffer	NullPointerException is thrown	
2	Search TLV 0Bh		
	dstOffset ≥ dstBuffer.length	ArrayIndexOutOfBoundsExceptio	
	dstBuffer.length = 5	n is thrown	
	dstOffset = 5		
	dstLength = 1		
3	dstOffset < 0	ArrayIndexOutOfBoundsExceptio	
	dstBuffer.length = 5	n is thrown	
	dstOffset = -1		
	dstLength = 1		
4	dstLength >dstBuffer.length	ArrayIndexOutOfBoundsExceptio	
	dstBuffer.length = 5	n is thrown	
	dstOffset = 0		
_	dstLength = 6	A da da da F	
5	dstOffset + dstLength >dstBuffer.length	ArrayIndexOutOfBoundsExceptio	
	<pre>dstBuffer.length = 5 dstOffset = 3</pre>	n is thrown	
	dstLength = 3		
6	dstLength < 0	ArrayIndexOutOfBoundsExceptio	
	dstBuffer.length = 5	n is thrown	
	dstOffset = 0	II lo unown	
	dstLength = -1		
7	Search TLV 06h		
	valueOffset ≥ TLV Length	ToolkitException.OUT_OF_TLV_	
	valueOffset = 6	BOUNDARIES is thrown	
	dstBuffer.length = 15		
	dstOffset = 0		
	dstLength = 1		
8	valueOffset < 0	ToolkitException.OUT_OF_TLV_	
	valueOffset = -1	BOUNDARIES is thrown	
	dstBuffer.length = 15		

	dstOffset = 0		
9	dstLength = 1	TablishEverntion OUT OF TIV	
9	dstLength > TLV length valueOffset = 0	ToolkitException.OUT_OF_TLV_	-
	dstBuffer.length = 15	BOUNDARIES is thrown	
	dstOffset = 0		
	dstLength = 7		
10	valueOffset + dstLength > TLV length	ToolkitException.OUT_OF_TLV_	
10	valueOffset = 2	BOUNDARIES is thrown	-
	dstBuffer.length = 15	BOUNDARIES IS UIIOWII	
	dstOffset = 0		
	dstLength = 5		
11	Search TLV 01h		
	copyValue()	ToolkitException.UNAVAILABLE	
		ELEMENT is thrown on the	
		copyValue() method	
12	Search TLV 06h	copy value() method	
12	Successful call	Described company (alread) is 0,40000	
		Result of copyValue() is 0x0006	
	<pre>valueOffset = 0 dstBuffer.length = 6</pre>		
	dstOffset = 0		
	dstLength = 6		
13	Compare buffer	Result is 00h	
10	buffer = 81 11 22 33 44 F5	result is con	
	initialize dstBuffer		
dstP	buffer = 55 55 55		
14	initialise dstBuffer		
	dstBuffer = 55 55 55		
	Successful call	Result of copyValue() is 0x0007	
	valueOffset = 1		
	dstBuffer.length = 20		
	dstOffset = 3		
	dstLength = 4		
15	Compare buffer	Result is 00h	
	buffer =		
	55 55 55 11 22		
	33 44 55 55 55		
	55 55 55 55 55		
16	55 55 55 55	Posult of copy\/alug() is 20	
16	55 55 55 55 55 Successful call, copy with length =0	Result of copyValue() is 20	
16	Successful call, copy with length =0 dstBuffer.length = 20	Result of copyValue() is 20	
16	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20	Result of copyValue() is 20	
16	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0	Result of copyValue() is 20	
16	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user	Result of copyValue() is 20	
16	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated	Result of copyValue() is 20	
	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes	Result of copyValue() is 20	
16 <u>17</u>	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG		
	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes	Result of copyValue() is 20 Result of copyValue() is 0x010D	
	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call		
	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0		
	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D	Result of copyValue() is 0x010D	
	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer		
17	Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer buffer = 0348 header and secured data (01)	Result of copyValue() is 0x010D	
<u>17</u>	Send Formatted SMS with the maximum user data length = 0x010D using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D dstOffset = 0 dstLength = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer buffer = 0348 header and secured data (01 FA)	Result of copyValue() is 0x010D	
17	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer buffer = 0348 header and secured data (01 FA) Initialise dstBuffer	Result of copyValue() is 0x010D	
<u>17</u>	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer buffer = 0348 header and secured data (01 FA) Initialise dstBuffer dstBuffer = 55 55 55	Result of copyValue() is 0x010D Result is 00h	
<u>17</u>	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer buffer = 0348 header and secured data (01 FA) Initialise dstBuffer dstBuffer = 55 55 55 Successful call	Result of copyValue() is 0x010D	
<u>17</u>	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer buffer = 0348 header and secured data (01 FA) Initialise dstBuffer dstBuffer = 55 55 55 Successful call valueOffset = 0x0111	Result of copyValue() is 0x010D Result is 00h	
<u>17</u>	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer buffer = 0348 header and secured data (01 FA) Initialise dstBuffer dstBuffer = 55 55 55 Successful call valueOffset = 0x0111 dstBuffer.length = 0x010D	Result of copyValue() is 0x010D Result is 00h	
<u>17</u>	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer buffer = 0348 header and secured data (01 FA) Initialise dstBuffer dstBuffer = 55 55 55 Successful call valueOffset = 0x0111 dstBuffer.length = 0x010D dstOffset = 0x0111 dstBuffer.length = 0x010D dstOffset = 0x0100	Result of copyValue() is 0x010D Result is 00h	
17 18 19	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer buffer = 0348 header and secured data (01 FA) Initialise dstBuffer dstBuffer = 55 55 55 Successful call valueOffset = 0x0111 dstBuffer.length = 0x010D dstDeffset = 0x0111 dstBuffer.length = 0x010D dstOffset = 0x0100 dstOffset = 0x0100 dstLength = 0x000D	Result of copyValue() is 0x010D Result is 00h Result of copyValue() is 0x010D	
<u>17</u>	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer buffer = 0348 header and secured data (01 FA) Initialise dstBuffer dstBuffer.length = 0x0111 dstBuffer.length = 0x010D dstOffset = 0x0111 dstBuffer.length = 0x010D dstOffset = 0x0110 dstOffset = 0x0100 dstOffset = 0x0100 dstOffset = 0x000D Compare buffer	Result of copyValue() is 0x010D Result is 00h	
17 18 19	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer buffer = 0348 header and secured data (01 FA) Initialise dstBuffer dstBuffer.length = 0x0111 dstBuffer.length = 0x010D dstOffset = 0x0111 dstBuffer = 55 55 55 Successful call valueOffset = 0x0111 dstBuffer.length = 0x010D dstOffset = 0x0100 dstOffset = 0x0100 dstOffset = 0x000D Compare buffer buffer =	Result of copyValue() is 0x010D Result is 00h Result of copyValue() is 0x010D	
17 18 19	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer buffer = 0348 header and secured data (01 FA) Initialise dstBuffer dstBuffer.length = 0x0111 dstBuffer.length = 0x010D dstOffset = 0x0111 dstBuffer.length = 0x010D dstOffset = 0x0110 dstOffset = 0x0100 dstOffset = 0x0100 dstOffset = 0x000D Compare buffer	Result of copyValue() is 0x010D Result is 00h Result of copyValue() is 0x010D	
17 18 19	Successful call, copy with length =0 dstBuffer.length = 20 dstOffset = 20 dstLength = 0 Send Formatted SMS with the maximum user data length = 0x010D, using 2 concatenated envelopes Search SMS TPDU TAG Successful call valueOffset = 0x11 dstBuffer.length = 0x010D dstOffset = 0 dstLength = 0x010D Compare buffer buffer = 0348 header and secured data (01 FA) Initialise dstBuffer dstBuffer.length = 0x0111 dstBuffer.length = 0x010D dstOffset = 0x0111 dstBuffer = 55 55 55 Successful call valueOffset = 0x0111 dstBuffer.length = 0x010D dstOffset = 0x0100 dstOffset = 0x0100 dstOffset = 0x000D Compare buffer buffer =	Result of copyValue() is 0x010D Result is 00h Result of copyValue() is 0x010D	

6.2.4.12.4 Test Coverage

CRR number	Test case number
N1	13, 15
<u>N1</u>	<u>13, 15, 18, 20</u>
N2	12, 14, 16
<u>N2</u>	<u>12, 14, 16, 17, 19</u>
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for
	EnvelopeHandler
C2	11

6.2.4.13 Method compareValue

Test Area Reference: API_2_ENH_CPRVS_BSS-

6.2.4.13.1 Conformance requirement

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

6.2.4.13.1.1 Normal execution

Compares the last found TLV element with a buffer:

CRRN1: returns 0 if identical.

CRRN2: returns -1 if the first miscomparing byte in simple TLV List is less than that in compareBuffer.

CRRN3: returns 1 if the first miscomparing byte in simple TLV List is greater than that in compareBuffer.

6.2.4.13.1.2 Parameter errors

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset or compareLength or both would cause access outside array bounds, or if compareLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

6.2.4.13.1.3 Context errors

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

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.2.4.13.2 Test Suite files

Specific triggering: None

Test Script: ____API_2_ENH_CPRVS_BSS_1.scr

Test Applet: API_2_ENH_CPRVS_BSS_1.java

Load Script: API_2_ENH_CPRVS_BSS_1.ldr

Cleanup Script: API_2_ENH_CPRVS_BSS_1.clr

Parameter File: API_2_ENH_CPRVS_BSS_1.java

Load Script: API_2_ENH_CPRVS_BSS_1.java

Load Script: API_2_ENH_CPRVS_BSS_1.clr

Cleanup Script: API_2_ENH_CPRVS_BSS_1.clr

Parameter File: API_2_ENH_CPRVS_BSS_1.par

6.2.4.13.3 Test procedure

Search TLV 02h CompareValue() with a null compareBuffer NullPointerException is thrown	ld	Description	API Expectation	APDU Expectation
CompareValue() with a mull compareBuffer NullPointerException is thrown	1		·	•
CompareOffset ≥ compareBuffer.length compareBuffer.length = 5		compareValue() with a null compareBuffe	NullPointerException is thrown	
compareBuffer length = 5 compareOffset = 5 compareLength = 1 3	2	Search TLV 0Bh		
CompareBuffer.length = 5		compareBuffer.length = 5		
compareBuffer.length = 5 compareDeffset = -1 compareDeffset = 1 compareDeffset = 0 compareDeffset = 0 compareDeffset + compareDeffset compareDeffset = 3 compareDeffset = 3 compareDeffset = 0 comp				
compareDiffert = 0 compareOffset = 0 compareOffset = 0 compareLength = 6 CompareDiffset + compareLength	3	<pre>compareBuffer.length = 5 compareOffset = -1 compareLength = 1</pre>	n is thrown	
ScompareBuffer.length CompareDuffer.length CompareOffset 2 Search TLV 106h	•	<pre>compareBuffer.length = 5 compareOffset = 0 compareLength = 6</pre>	n is thrown	
compareBuffer.length = 5 n is thrown compareOffset = 0 compareIngth = -1 7 Search TLV 06h valueOffset ≥ TLV Length ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown valueOffset = 6 BOUNDARIES is thrown compareDuffer.length = 15 CompareDuffset = 0 compareDuffer.length = 15 ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown valueOffset = -1 BOUNDARIES is thrown compareDuffer.length = 15 BOUNDARIES is thrown compareDuffer.length = 15 CompareDuffset = 0 compareDuffer.length = 7 ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown 10 ValueOffset + compareLength > TLV length valueOffset = 0 compareDuffer.length = 15 ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown 10 ValueOffset + compareLength > TLV length valueOffset = 0 compareDuffer.length = 15 ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown 11 Search TLV 01h Result is TLV_NOT_FOUND compareValue() ToolkitException.UNAVAILABLE_ELEMENT is thrown	5	>compareBuffer.length compareBuffer.length = 5 compareOffset = 3	n is thrown	
valueOffset ≥ TLV Length valueOffset = 6 BOUNDARIES is thrown compareBuffer.length = 15 CompareOffset = 0 compareLength = 1 ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown 8 valueOffset < 0	6	<pre>compareBuffer.length = 5 compareOffset = 0 compareLength = -1</pre>		
valueOffset = 6 compareBuffer.length = 15 compareOffset = 0 compareLength = 1 8	7	Search TLV 06h		
8		<pre>valueOffset = 6 compareBuffer.length = 15 compareOffset = 0</pre>		
9	8	<pre>valueOffset < 0 valueOffset = -1 compareBuffer.length = 15 compareOffset = 0</pre>		
10 valueOffset + compareLength > TLV length valueOffset = 2 compareBuffer.length = 15 compareOffset = 0 compareLength = 5 11 Search TLV 01h compareValue() Search TLV 06h ToolkitException.OUT_OF_TLV_ BOUNDARIES is thrown Result is TLV_NOT_FOUND ToolkitException.UNAVAILABLE _ELEMENT is thrown	9	<pre>compareLength > TLV length valueOffset = 0 compareBuffer.length = 15 compareOffset = 0</pre>		
11 Search TLV 01h Result is TLV_NOT_FOUND compareValue() ToolkitException.UNAVAILABLE _ELEMENT is thrown 12 Search TLV 06h	10	<pre>valueOffset + compareLength > TLV lengt valueOffset = 2 compareBuffer.length = 15 compareOffset = 0</pre>		
compareValue() ToolkitException.UNAVAILABLE _ELEMENT is thrown 12 Search TLV 06h	11		Result is TLV_NOT_FOUND	
			ToolkitException.UNAVAILABLE	
Initialize compare Duffer	12	Search TLV 06h		
minanze compareburier		Initialize compareBuffer	<u> </u>	

		81 11 22 33 44 F5	
		Compare buffers	Result is 00h
		valueOffset = 0	
		compareOffset = 0	
		_	
		compareLength = 6	
13		Initialize compareBuffer	
	comp	pareBuffer =	
ļl.,	7F 1	1 22 33 44 F5	
	13	Initialise compareBuffer	
		compareBuffer =	
		7F 11 22 33 44 F5	
I ,		Compare buffers with same parameters	Result is -1
14		Initialize compareBuffer	TCSUIT 13 - 1
44		Initialize compareburier	
	comp	pareBuffer =	
ļl.,	83 1	1 22 33 44 F5	
	<u>14</u>	Initialise compareBuffer	
		compareBuffer =	
		83 11 22 33 44 F5	
		Compare buffers with same parameters	Result is -1
15	1	Initialize compareBuffer	
11 10	a	oareBuffer =	
	COIII)	paresuffer = 55 55 81 11 22 23 44 F5	
	55 5	75 55 61 11 22 33 11 F5	
ļ1 ,	55 5)3 33 33	
	<u>15</u>	Initialise compareBuffer	
		<pre>compareBuffer =</pre>	
		55 55 55 81 11 22 33 44 F5	
		55 55 55 55	
		Compare buffers	Result is 00h
		valueOffset = 1	
		compareOffset = 4	
		compareLength = 5	
16		Initialize compareBuffer	
	comp	pareBuffer =	
	55 5	55 55 81 10 22 33 44 F5	
		55 55 55 55	
i	16	Initialise compareBuffer	
	<u></u>	compareBuffer =	
		55 55 55 81 10 22 33 44 F5	
		55 55 55 55 55	
'		Compare buffers with same parameters	Result is +1
17		Initialize compareBuffer	1.00 care 10 1 1
++	a	nntianze comparebutter	
	E E	55 55 81 12 22 33 44 F5	
	000	5 5 5 5 5 5	
¦'	000	Initialise compareBuffer	
	<u>17</u>		
		<pre>compareBuffer =</pre>	
		55 55 55 81 12 22 33 44 F5	
1		55 55 55 55	D. His A
		Compare buffers with same parameters	Result is -1
ſ	18	Successful call, compareValue with length =0	Result of compareValue() is 0
		CompareBuffer.length = 15	"
		CompareOffset = 15	
		CompareLength = 0	
		Send Formatted SMS PP with the maximum	
		user data length = 0x010D, using 2	
		concatenated envelopes	
}			
		Search SMS TPDU TAG	
		Initialise compareBuffer	
		compareBuffer = 0348 header and formatted	
		data(01 02 FA)	
	<u>19</u>	Compare buffers	Result is 00h
		<pre>valueOffset = 0x11</pre>	
		<pre>compareOffset = 0</pre>	
		$\underline{\text{compareLength} = 0x010D}$	
		compareBufferLength = 0x010D	
] [20	Compare buffers	Result is 00h
		<pre>valueOffset = 0x0111</pre>	
		<pre>compareOffset = 0x0100</pre>	
		compareLength = 0x000D	
			•
		compareBufferLength = 0x010D	

6.2.4.13.4 Test Coverage

CRR number	Test case number
N1	12, 15
<u>N1</u>	<u>12, 15, 19, 20</u>
N2	13, 16, 18
N3	14, 17
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for
	EnvelopeHandler
C2	11

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

Test Area Reference: API_2_ENH_FACYB_BS-

6.2.4.14.1 Conformance requirement

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

6.2.4.14.1.1 Normal execution

CRRN1: looks for the first occurrence of a TLV element from the beginning of a TLV list and copy its value into a destination buffer.

CRRN2: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN3: if the method is successful then the corresponding TLV becomes current and dstOffset + length of the copied value is returned.

CRRN4: The search method is comprehension required flag independent.

6.2.4.14.1.2 Parameter errors

CRRP1: if dstBuffer is null NullPointerException shall be thrown.

CRRP2: if dstOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

6.2.4.14.1.3 Context errors

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

6.2.4.14.2 Test Suite files

Specific triggering: None

Test Script: _____API_2_ENH_FACYB_BS_1.scr

Test Applet: API_2_ENH_FACYB_BS_1.java

Load Script: API_2_ENH_FACYB_BS_1.ldr

Cleanup Script: API_2_ENH_FACYB_BS_1.clr

Parameter File: API 2 ENH FACYB BS 1.java

Load Script: API 2 ENH FACYB BS 1.ldr

Cleanup Script: API 2 ENH FACYB BS 1.clr

Parameter File: API 2 ENH FACYB BS 1.par

6.2.4.14.3 Test procedure

	ld	Description		API Expectation APDU Exp		tation
		Fill the SMS PP with TLV: Tag 02 Value 22 44		-	_	
		Tag 33, Length C4 Value 01 02				
	1			NullPointerException is thrown		
	2	dstOffset ≥ dstBuffer.length		ArrayIndexOutOfBoundsExcepti	io	
		tag = 06h dstBuffer.length = 06		n is thrown		
		dstOffset = 06				
	3	dstOffset < 0		ArrayIndexOutOfBoundsExcepti	io	
		dstBuffer.length = 06		n is thrown		
		dstOffset = -1		A		
	4	<pre>length > dstBuffer.length dstBuffer.length = 05</pre>		ArrayIndexOutOfBoundsExcepti n is thrown	10	
		dstOffset = 0		II IS UIIOWII		
	5	DstOffset + length >dstBuffer.length		ArrayIndexOutOfBoundsExcepti	io	
		DstBuffer.length = 06		n is thrown		
		DstOffset = 1				
	6	Select a TLV (tag 02h)				
	U	findAndCopyValue()		ToolkitException.UNAVAILABLE	=	
		tag = 03h		_ELEMENT is thrown	-	
		Call the getValueLength() method	Tooll	kitException.UNAVAILABLE		
				MENT is thrown.		
	<u>7</u>	Call the getValueLength() method		ToolkitException.UNAVAILABLE	E	•
				ELEMENT is thrown.		
7		Successful call		ult of findAndCopyValue () is		
	Tag Dat F	= Uon	0006			
	Date:	Offset = 0				
	8	Successful call		Result of findAndCopyValue () is	<u>S</u>	
		$\frac{\text{Tag} = 06h}{\text{Tag}}$		<u>0006</u>		
		<pre>DstBuffer.length = 06 DstOffset = 0</pre>				
8			Resu	ılt is 00h		
1.	buf f	er = 81 11 22 33 44 F5				
	<u>9</u>	Compare buffer		Result is 00h		
		buffer = 81 11 22 33 44 F5 initialize dstBuffer				
9	datF	unitalize astbuller Buffer = 55 55 55				
1	10	initialise dstBuffer		l .		
		dstBuffer = 55 55 55				
		Successful call		Result of findAndCopyValue () is	S	
		<pre>dstBuffer.length = 12 dstOffset = 2</pre>		0008		
10			Resi	ılt is 00h		
	buf f	fer =				
	55 5	5 5 81 11 22 33 44 F5 55 55 55 55				
	11	Compare huffer	ı	Popult is 00h		
	<u>11</u>	Compare buffer buffer =		Result is 00h		
		55 55 81 11 22 33 44 F5 55 55 55 55				
44				ult of findAndCopyValue () is		
	tag	= 02h Suffer.length = 2	0002	!		
		Offset = 0				
.	<u>12</u>	Successful call		Result of findAndCopyValue () is	<u>S</u>	,
		tag = 02h		0002		
		<pre>dstBuffer.length = 2 dstOffset = 0</pre>				
12		Compare buffer	Resi	ılt is 00h	l	
	1	paro barror				

	buff	cr = 83 81		
	13	Compare buffer		Result is 00h
		buffer = 83 81		
13		Successful call (with tag 82h)	Resu	ult of findAndCopyValue () is
	tag	= 82h	0002	
	dstB	uffer.length = 02		
	dst0	ffset = 0		
	14	Successful call (with tag 82h)		Result of findAndCopyValue () is
		tag = 82h		0002
		<pre>dstBuffer.length = 02</pre>		
		<u>dstOffset = 0</u>		
14		Compare buffer	Resu	ılt is 00h
	buf f	er = 83 81		
	<u>15</u>	Compare buffer		Result is 00h
		<u>buffer = 83 81</u>		
15		Successful call (with tag B3h)	Resu	ult of findAndCopyValue () is
		= B3h	00C/	4
	dstB	uffer.length = C4		
1 .	dst0	ffset = 0	L.,	
	<u>16</u>	Successful call (with tag B3h)		Result of findAndCopyValue () is
		tag = B3h		<u>00C4</u>
		<pre>dstBuffer.length = C4</pre>		
		dstOffset = 0		
16		Compare buffer	Resu	Ilt is 00h
		er = 01 02 C4		
	<u>17</u>	Compare buffer		Result is 00h
ŀ		buffer = 01 02 C4		
		Send Formatted SMS PP with the maximu	m_	
		user data length = 0x010D, using 2		
		concatenated envelopes		
	<u>18</u>	Successful call (with SMS TPDU TAG)		Result of findAndCopyValue () is
		tag = 0Bh		<u>0x011E</u>
		dstBuffer.length = 0x011E		
ļ		dstOffset = 0		D. H.I. COL
	<u>19</u>	Compare buffer		Result is 00h
	<pre>buffer = 0348 Header + secured data (01</pre>		1_	
	000	02 FA)		D 16 (C 14 10 1/1 0)
	<u>20</u>	Successful call (with SMS TPDU TAG)		Result of findAndCopyValue () is
	tag = 0Bh			<u>0x021E</u>
		<pre>dstBuffer.length = 0x0220 dstOffset = 0x0100</pre>		
-	24			Decult in 00h
	<u>21</u>	Compare buffer	,	Result is 00h
		buffer = 0348 Header + secured data (0	<u> </u>	
		02 FA)		

6.2.4.14.4 Test Coverage

CRR number		Test case number	
N1		8, 10, 12	
N2		6	
N3		7, 9, 11	
N4		13, 14, 15, 16	
<u>N1</u>		<u>9, 11, 13</u>	
<u>N2</u>		<u>6, 7</u>	
<u>N3</u>		<u>8, 10, 12</u>	
<u>N4</u>		<u>14, 15, 16, 17, 18, 19, 20,</u>	
		<u>21</u>	
P1		1	
P2		2, 3, 4, 5	
C1	•	Does not apply for	
		EnvelopeHandler	

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

Test Area Reference: API_2_ENH_FACYBS_BSS-

6.2.4.15.1 Conformance requirement

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

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

6.2.4.15.1.1 Normal execution

CRRN1: looks for the indicated occurrence of a TLV element from the beginning of a TLV list and copy its value into a destination buffer.

32

CRRN2: -if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN3: if the method is successful then the corresponding TLV becomes current and dstOffset + dstLength is returned.

CRRN4: The search method is comprehension required flag independent.

6.2.4.15.1.2 Parameter errors

CRRP1: if dstBuffer is null NullPointerException shall be thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, dstLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

6.2.4.15.1.3 Context errors

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

6.2.4.15.2 Test Suite files

Test Script:	API_2_ENH_FACYBS_BSS_1.scr
Test Applet:	API_2_ENH_FACYBS_BSS_1.java
Load Script:	API_2_ENH_FACYBS_BSS_1.ldr
Cleanup Script:	API_2_ENH_FACYBS_BSS_1.clr
Parameter File:	API_2_ENH_FACYBS_BSS_1.java
Load Script:	API_2_ENH_FACYBS_BSS_1.ldr
Cleanup Script:	API_2_ENH_FACYBS_BSS_1.clr
Parameter File:	API_2_ENH_FACYBS_BSS_1.par

6.2.4.15.3 Test procedure

ld	Description	API Expectation	APDU Expectation
	Fill the SMS PP with TLV: Tag 02 Value 22 44		
	Tag 33, Length C4 Value 01 02		
1	findAndCopyValue() with a null dstBuffer	NullPointerException is thrown	

-	2	<pre>dstOffset ≥ dstBuffer.length tag = 06h, occurrence = 1 valueOffset = 0 dstBuffer.length = 5</pre>		ArrayIndexOutOfBoundsExceptio n is thrown	
		dstOffset = 5			
-	3	<pre>dstLength = 1 dstOffset < 0 dstBuffer.length = 5 dstOffset = -1</pre>		ArrayIndexOutOfBoundsExceptio n is thrown	
-	4	<pre>dstLength = 1</pre>		ArrayIndexOutOfBoundsExceptio n is thrown	
-	5	<pre>dstOffset + dstLength >dstBuffer.length dstBuffer.length = 5 dstOffset = 3 dstLength = 3</pre>)	ArrayIndexOutOfBoundsExceptio n is thrown	
	6	<pre>dstLength < 0 dstBuffer.length = 5 dstOffset = 0 dstLength = -1</pre>		ArrayIndexOutOfBoundsException is thrown	
_	7	<pre>valueOffset ≥ Value Length tag = 06h, occurrence = 1 valueOffset = 6 dstBuffer.length = 15 dstOffset = 0 dstLength = 1</pre>		ToolkitException.OUT_OF_TLV_ BOUNDARIES is thrown	
	8	<pre>valueOffset < 0 valueOffset = -1 dstBuffer.length = 15 dstOffset = 0 dstLength = 1</pre>		ToolkitException.OUT_OF_TLV_ BOUNDARIES is thrown	
	9	<pre>dstLength > Value length valueOffset = 0 dstBuffer.length = 15 dstOffset = 0 dstLength = 7</pre>		ToolkitException.OUT_OF_TLV_ BOUNDARIES is thrown	
<u>-</u>	10	<pre>valueOffset + dstLength > Text String lengt valueOffset = 2 dstBuffer.length = 15 dstOffset = 0 dstLength = 5</pre>	th	ToolkitException.OUT_OF_TLV_ BOUNDARIES is thrown	
	11	Select a TLV (tag 02h)			
		<pre>findAndCopyValue() tag = 06h occurrence = 2</pre>		ToolkitException.UNAVAILABLE _ELEMENT is thrown	
				kitException.UNAVAILABLE_ MENT is thrown.	
	<u>12</u>	Call the getValueLength() method		ToolkitException.UNAVAILABLE	I
12		Successful call	Paci	<u>ELEMENT is thrown.</u> ult of find∧ndCopyValue() is 6	
12	tag	= 06h, occurrence = 1	1 (03)	ant of infartiacopy value() is o	
	valu	neOffset = 0 Ouffer.length = 06			
	dstC	Offset = 0			
<u> </u>	dstI	cength = 06		Result of findAndCopyValue() is	
	<u>13</u>	Successful call tag = 06h, occurrence = 1 valueOffset = 0 dstBuffer.length = 06 dstOffset = 0		6	
13		•	Resi	ult is 00h	
l 「	buff 14	Cer = 81 11 22 33 44 F5 Compare buffer		Result is 00h	
4.4		buffer = 81 11 22 33 44 F5			
14	datE	initialize dstBuffer			
' [<u>15</u>	initialise dstBuffer dstBuffer = 55 55 55		I I	I
ŀ		Successful call		Result of findAndCopyValue () is	
		tag = 06h, occurrence = 1		0007	

	3 055 1 0	
	valueOffset = 2	
	dstBuffer.length = 12	
	dstOffset = 3	
	dstLength = 04	D It ': 001
15	Compare buffer	Result is 00h
	buffer =	
	55 55 55 22 33 44 F5 55 55 55 55 55	
	<u>Compare buffer</u>	Result is 00h
	<pre>buffer =</pre>	
	55 55 55 22 33 44 F5 55 55 55 55 55	
16	Successful call	Result of findAndCopyValue() is
	tag = 02h, occurrence = 1	0002
	valueOffset = 0	
	dstBuffer.length = 12	
	dstOffset = 0	
	dstLength = 2	
	17 Successful call	Result of findAndCopyValue() is
	tag = 02h, occurrence = 1	0002
	valueOffset = 0	
	dstBuffer.length = 12	
	dstOffset = 0	
	$\overline{dstLength} = 2$	
17	Compare buffer	Result is 00h
	buffer = 83 81 55 55	
j ' [18 Compare buffer	Result is 00h
	buffer = 83 81 55 55	
18	Successful call	Result of findAndCopyValue() is
'	tag = 02h, occurrence = 2	0002
	valueOffget = 0	000 <u>2</u>
	dstBuffer.length = 12	
	dstOffset = 0	
	dstLength = 2	
	19 Successful call	Result of findAndCopyValue() is
	tag = 02h, occurrence = 2	0002
	valueOffset = 0	<u> </u>
	dstBuffer.length = 12	
	dstOffset = 0	
	dstLength = 2	
19	Compare buffer	Result is 00h
	buffer = 22 44 55 55	
	20 Compare buffer	Result is 00h
	buffer = 22 44 55 55	
20	Successful call (with tag 82h)	Result of find/AndCopyValue () is
	tag = 82h	0002
	occurrence = 1	
	valueOffset = 0	
	valueOffset = 0 dstBuffer.length = 12	
	<pre>dstBuffer.length = 12 dstOffset = 0</pre>	
	<pre>dstBuffer.length = 12 dstOffset = 0 dstLength = 02</pre>	
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21 Successful call (with tag 82h)	Result of findAndCopyValue () is
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21 Successful call (with tag 82h) tag = 82h	Result of findAndCopyValue () is 0002
	<pre>dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21</pre>	
	<pre>dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21</pre>	
	<pre>dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21</pre>	
	<pre>dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21</pre>	
	<pre>dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21</pre>	0002
24	<pre>dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21</pre>	
24	<pre>dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21</pre>	Result is 00h
24	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	0002
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result is 00h
24	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result of findAndCopyValue () is
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result is 00h
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result of findAndCopyValue () is
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result of findAndCopyValue () is
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result of findAndCopyValue () is
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result of findAndCopyValue () is
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result is 00h Result of findAndCopyValue () is 0002
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result of findAndCopyValue () is 0002 Result of findAndCopyValue () is 0002
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result is 00h Result of findAndCopyValue () is 0002
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result of findAndCopyValue () is 0002 Result of findAndCopyValue () is 0002
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result of findAndCopyValue () is 0002 Result of findAndCopyValue () is 0002
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result of findAndCopyValue () is 0002 Result of findAndCopyValue () is 0002
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result of findAndCopyValue () is 0002 Result of findAndCopyValue () is 0002
	dstBuffer.length = 12 dstOffset = 0 dstLength = 02 21	Result is 00h Result of findAndCopyValue () is 0002 Result of findAndCopyValue () is 0002

	Buff	Eer = 22 44 55 55	
	24	Compare buffer	Result is 00h
		Buffer = 22 44 55 55	
24	Suc	, , , , , , , , , , , , , , , , , , , ,	sult of findAndCopyValue () is-
		=0 12	
	DstI	Suffer.length = 12	
	dst	Gength = 0	
	25	Successful call, findAndCopyValue with	Result of findAndCopyValue () is
	==	length =0	12
		DstBuffer.length = 12	<u></u>
		dstOffset = 12	
		dstLength = 0	
		Send Formatted SMS PP with the maximum	
		user data length = 0x010D, using 2	
		concatenated envelopes	D
	<u>26</u>	Successful call tag = 0Bh, occurrence = 1	Result of findAndCopyValue() is 0x010D
		valueOffset = 0x11	<u>0x010D</u>
		dstBuffer.length = 0x010D	
		dstOffset = 0	
		dstLength = 0x010D	
	<u>27</u>	Compare buffer	Result is 00h
		buffer = 0348 Header + secured data (01 02 FA)	
	28	initialise dstBuffer	
		dstBuffer = 55 55 55	
		Successful call	Result of findAndCopyValue () is
		tag = 0Bh, occurrence = 1	0x010D
		<pre>valueOffset = 0x0111</pre>	
		dstBuffer.length = 0x010D dstOffset = 0x0100	
		$\frac{\text{dstOliset} = 0x0100}{\text{dstLength} = 0x0D}$	
	29	Compare buffer	Result is 00h
	==	buffer =	
		55 55 55 55 EE EF F0 F1 F2 F3 F4 F5 F6	
		F7 F8 F9 FA	

6.2.4.15.4 Test Coverage

CRR number	Test case number
N1	13, 15, 17, 19
N2	11
N3	12, 14, 16, 18, 24
N4	20, 21, 22, 23
<u>N1</u>	<u>14, 15, 17, 19, 20</u>
<u>N2</u>	<u>11, 12</u>
<u>N3</u>	<u>13, 15, 17, 19, 25</u>
<u>N4</u>	<u>21, 22, 23, 24, 26, 27, 28,29</u>
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for EnvelopeHandler

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

Test Area Reference: API_2_ENH_FACRB_BS-

6.2.4.16.1 Conformance requirement

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

throws java.lang.NullPointerException,
 java.lang.ArrayIndexOutOfBoundsException,
 ToolkitException

6.2.4.16.1.1 Normal execution

Looks for the first occurrence of a TLV element from beginning of a TLV list and compare its value with a buffer:

CRRN1: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN2: if the method is successful then the corresponding TLV becomes current.

CRRN3: if identical returns 0.

CRRN4: if the first miscomparing byte in simple TLV is less than that in compareBuffer returns -1.

CRRN5: if the first miscomparing byte in simple TLV is greater than that in compareBuffer returns 1.

CRRN6: The search method is comprehension required flag independent.

6.2.4.16.1.2 Parameter errors

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset would cause access outside array bounds ArrayIndexOutOfBoundsException shall be thrown.

6.2.4.16.1.3 Context errors

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

6.2.4.16.2 Test Suite files

Test Script: ____API_2_ENH_FACRB_BS_1.scr

Test Applet: API_2_ENH_FACRB_BS_1.java

Load Script: API_2_ENH_FACRB_BS_1.ldr

Cleanup Script: API_2_ENH_FACRB_BS_1.clr

Parameter File: API_2_ENH_FACRB_BS_1.java

Load Script: API_2_ENH_FACRB_BS_1.ldr

Cleanup Script: API_2_ENH_FACRB_BS_1.clr

Parameter File: API_2_ENH_FACRB_BS_1.clr

Parameter File: API_2_ENH_FACRB_BS_1.par

6.2.4.16.3 Test procedure

ld	Description	API Expectation	APDU Expectation
	Fill the SMS PP with TLV: Tag 02 Value 22 44		
	Tag 33, Length C4 Value 01 02		
1	findAndCompareValue() with a null dstBuffer	NullPointerException is thrown	
2	compareOffset ≥ compareBuffer.length	ArrayIndexOutOfBoundsExceptio	
	tag = 06h	n is thrown	
	compareBuffer.length = 12		
	compareOffset = 12		
3	compareOffset < 0	ArrayIndexOutOfBoundsExceptio	
	compareBuffer.length = 12	n is thrown	
	compareOffset = -1		
4	length > compareBuffer.length	ArrayIndexOutOfBoundsExceptio	
	compareBuffer.length = 05	n is thrown	
	compareOffset = 0		

	5	compareOffset + length >		ArrayIndexOutOfBoundsExcepti	0
	_				~
		compareBuffer.length		n is thrown	
		compareBuffer.length = 12			
		compareOffset = 7			
	_				
	6	Select a TLV (tag 02h)			
		findAndCompareValue()		ToolkitException.UNAVAILABLE	=
		tag = 03h			-
		1 2		_ELEMENT is thrown	
		Call the getValueLength() method	Tool	kitException.UNAVAILABLE	
		can the gettardezongm() memod			
			ELE	MENT is thrown.	
	7	Call the getValueLength() method		ToolkitException.UNAVAILABLE	
	<u> </u>	our the gerrand conging mother			=
				ELEMENT is thrown.	
7		Initialize compareBuffer			
	aomr	pareBuffer = 81 11 22 33 44 F5			
	COME				
	<u>8</u>	Initialise compareBuffer			
		compareBuffer = 81 11 22 33 44 F5			
1				Desult is 00h	
		Compare buffers		Result is 00h	
		tag = 06h			
		compareOffset = 0			
	<u> </u>		D	dr != 00	
8		Verify current TLV	Resi	30 si tlu	
	get\	/alueLength()			
1	9	Verify current TLV	•	Result is 06	<u> </u>
	3			I COULT TO UU	
1.	<u> </u>	<pre>getValueLength()</pre>			
9		Initialize compareBuffer	1		
11 ~	aom~	pareBuffer = 81 11 22 33 44 F4	Ì		
	COIII		l	ļ	,
	<u>10</u>	Initialise compareBuffer			
	1	compareBuffer = 81 11 22 33 44 F4			
ı				Description of	
		Compare buffers with same parameters	<u> </u>	Result is +1	
10	T	Initialize compareBuffer			
11.0	l	pareBuffer = 81 11 22 33 44 F6			
	comp				
	11	Initialise compareBuffer			
		compareBuffer = 81 11 22 33 44 F6			
ı				D 11: 1	
		Compare buffers with same parameters	•	Result is -1	
11		Initialize compareBuffer			
11		Tittidii 20 Compai CDanoi			
	com	parebuller =			
	55 5	55 81 11 22 33 44 F5 55 55 55 55			
	12	Initialise compareBuffer			
	12	compareBuffer =			
		55 55 81 11 22 33 44 F5 55 55 55 55			
		Compare buffers		Result is 00h	
		compareOffset = 2		1 toodit to ooti	
	ļ			<u></u>	
12		Initialize compareBuffer			
	COM	pareBuffer =			
		55 83 81 55 55 55 55 55 55 55 55			
ļ I					
	13	Initialise compareBuffer			
		compareBuffer =			
		55 55 83 81 55 55 55 55 55 55 55			
I					
		Compare buffers		Result is 00h	
		compareOffset = 2			
13	1				
13		Initialize compareBuffer			
		pareBuffer =			
	55 5	55 83 80 55 55 55 55 55 55 55			
1'		Initialise compareBuffer			
	<u>14</u>				
		<pre>compareBuffer =</pre>			
1		55 55 83 80 55 55 55 55 55 55 55			
•		Compare buffers		Result is +1	
				INCOURT IO TI	
	<u> </u>	compareOffset = 2			
14		Initialize compareBuffer	I		
		pareBuffer =	I		
ļ l		55 83 82 55 55 55 55 55 55 55	l		
	15	Initialise compareBuffer			
	1	compareBuffer =			
1		55 55 83 82 55 55 55 55 55 55 55			
		Compare buffers		Result is -1	
		compareOffset = 2			
1 -				<u> </u>	
15		Initialize compareBuffer	I		
	com	pareBuffer =	I		
		31 55 55 55 55 55 55 55 55 55	I		
1					
	<u>16</u>	Initialise compareBuffer			
	I	compareBuffer =			
		83 81 55 55 55 55 55 55 55 55 55 55			
ı	-			D	
		Successful call (with tag 02h)		Result is 00h	
	1	tag = 02h			

		compareBuffer.length = 12	
		compareOffset = 0	
16		Initialize compareBuffer	
11	Comp	pareBuffer = 01 02 C4	
1' 1	17	Initialise compareBuffer	
		CompareBuffer = 01 02 C4	
		Successful call (with tag B3h)	Result is 00h
		Tag = B3h	
		CompareBuffer.length = C4	
		CompareOffset = 0	
		Send Unformatted SMS PP with the maximu	<u>ım</u>
		user data length = 0x010C, using 2	
		concatenated envelopes	
		Initialise compareBuffer	
		CompareBuffer = 0340 Header + user data	_
		(00 01 02 FF 01 OC)	
	<u>18</u>	Successful call (with SMS TPDU TAG)	Result is 00h
		Tag = 0Bh	
		<pre>CompareBuffer.length = 0x011E</pre>	
		CompareOffset = 0	
		Initialise compareBuffer	
		<pre>CompareBuffer = 55 55 55 CompareBuffer from offset 0x0100= 0340</pre>	
		Header + user data (00 01 02 FF 01	
		OC)	
1	19	Successful call (with SMS TPDU TAG)	Result is 00h
	10	Tag = 0Bh	TOOGIL TO COLL
		CompareBuffer.length = 0x220	
		CompareOffset = 0x0100	

6.2.4.16.4 Test Coverage

CRR number		Test case number	
N1		6	
N2		8	
N3		7, 11, 12	
N 4		9, 13	
N5		10, 14	
N6		15, 16	
<u>N1</u>		<u>6,7</u>	
<u>N2</u>		<u>9</u>	
<u>N3</u>		<u>8, 12, 13, 18, 19</u>	
<u>N4</u>		<u>10, 14</u>	
<u>N5</u>		<u>11, 15</u>	
<u>N6</u>		<u>16, 17</u>	
P1		1	
P2	•	2, 3, 4, 5	
C1		Does not apply for	
		EnvelopeHandler	

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

Test Area Reference: API_2_ENH_FACRBBS_BSS-

6.2.4.17.1 Conformance requirement

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

java.lang.ArrayIndexOutOfBoundsException,				
	ToolkitException			
	java.lang.ArrayIndexOutOfBoundsException,			
	ToolkitException			

6.2.4.17.1.1 Normal execution

Looks for the indicated occurrence of a TLV element from the beginning of a TLV list and compare its value with a buffer:

CRRN1: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN2: if the method is successful then the corresponding TLV becomes current.

CRRN3: if identical- 0 is returned.

CRRN4: if the first miscomparing byte in simple TLV is less than that in compareBuffer -1 is returned.

CRRN5: if the first miscomparing byte in simple TLV is greater than that in compareBuffer 1 is returned

CRRN6: The search method is comprehension required flag independent.

6.2.4.17.1.2 Parameter errors

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset or compareLength or both would cause access outside array bounds, or if compareLength is negative ArrayIndexOutOfBoundsException shall be thrown.

CRRP3: if valueOffset, compareLength or both are out of the current TLV an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

CRRP4: if an input parameter is not valid (e.g. occurence = 0) an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException BAD_INPUT_PARAMETER.

6.2.4.17.1.3 Context errors

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

6.2.4.17.2 Test Suite files

Test Script:	API_2_ENH_FACRBBS_BSS_1.scr
Test Applet:	API_2_ENH_FACRBBS_BSS_1.java
Load Script:	API_2_ENH_FACRBBS_BSS_1.ldr
Cleanup Script:	API_2_ENH_FACRBBS_BSS_1.clr
Parameter File:	API_2_ENH_FACRBBS_BSS_1.java
Load Script:	API_2_ENH_FACRBBS_BSS_1.ldr
Cleanup Script:	API 2 ENH FACRBBS BSS 1.clr
Parameter File:	API_2_ENH_FACRBBS_BSS_1.par

6.2.4.17.3 Test procedure

ld	Description	API Expectation	APDU Expectation
	Fill the SMS PP with TLV: Tag 02 Value 22 44		
	Tag 33, Length C4 Value 01 02		
1	findAndCompareValue() with a null	NullPointerException is thrown	
	compareBuffer		
2	compareOffset ≥ compareBuffer.length	ArrayIndexOutOfBoundsExceptio	

		tag = 06h, occurrence = 1		n is thrown	
		valueOffset = 0			
		compareBuffer.length = 6			
		compareOffset = 6			
		compareLength = 1			
	3	compareOffset < 0		ArrayIndexOutOfBoundsExceptio	
		compareBuffer.length = 6		n is thrown	
		compareOffset = -1			
		compareLength = 1			
	4	compareLength >compareBuffer.length	1	ArrayIndexOutOfBoundsExceptio	
		compareBuffer.length = 5		n is thrown	
		compareOffset = 0			
		compareLength = 6			
	5	compareOffset + compareLength		ArrayIndexOutOfBoundsExceptio	
		>compareBuffer.length		n is thrown	
		compareBuffer.length = 5			
		compareOffset = 3			
		compareLength = 3			
	6	compareLength < 0		ArrayIndexOutOfBoundsExceptio	
		compareBuffer.length = 5		n is thrown	
		compareOffset = 0			
		compareLength = -1			
	7	valueOffset ≥ Value Length		ToolkitException.OUT_OF_TLV_	
		tag = 06h, occurrence = 1		BOUNDARIES is thrown	
		valueOffset = 6			
		<pre>compareBuffer.length = 15 compareOffset = 0</pre>			
		compareLength = 1			
	8	valueOffset < 0		ToolkitException.OUT_OF_TLV_	
	•	valueOffset = -1		BOUNDARIES is thrown	
		compareBuffer.length = 15		BOOMB/ WILLO IO WILLOWIT	
		compareOffset = 0			
		compareLength = 1			
	9	compareLength > Value length		ToolkitException.OUT_OF_TLV_	
		<pre>valueOffset = 0</pre>		BOUNDARIES is thrown	
		compareBuffer.length = 15			
		compareOffset = 0			
		compareLength = 7			
	10	valueOffset + compareLength > Value length	gth	ToolkitException.OUT_OF_TLV_	
		<pre>valueOffset = 2 compareBuffer.length = 15</pre>		BOUNDARIES is thrown	
		compareBuller.length = 15 compareOffset = 0			
		compareLength = 5			
11		Invalid parameter	Tool	kitException.BAD_INPUT_PA	
	occu	irrence = 0		METER is thrown	
	11	Invalid parameter	1000	ToolkitException.BAD INPUT P	
		occurrence = 0		ARAMETER is thrown	
	12	Select a TLV (tag 02h)		74 O WIE LEICHO GHIOWH	
	12	findAndCompareValue()	Tool	kitException.UNAVAILABLE	
	tag	 06h		MENT is thrown	
	oggi	= 0011 		WIEINT IS UITOWIT	
1	0000	findAndCompareValue()	ı	ToolkitException.UNAVAILABLE	' '
		tag = 06h		ELEMENT is thrown	
		occurrence = 2			
		Call the getValueLength() method	Tool	kitException.UNAVAILABLE	
		3. 0		MENT is thrown.	
	13	Call the getValueLength() method	1	ToolkitException.UNAVAILABLE	<u> </u>
		San and gotterade origin() morriou		_ELEMENT is thrown.	
13		Initialize compareBuffer			T
	come	pareBuffer = 81 11 22 33 44 F5			
1	14	Initialise compareBuffer	1	,	
		compareBuffer = 81 11 22 33 44 F5			
		findAndCompareValue()		Result is 00h	
		tag = 06h, occurrence = 1		. 1000 10 00	
		valueOffset = 0			
		compareOffset = 0			
		compareLength = 6			
14		Verify current TLV	Res	ult is 0006	
	get\	ValueLength()			
	<u>15</u>	Verify current TLV		Result is 0006	
		<pre>getValueLength()</pre>		<u> </u>	<u> </u>
4 -	1	Initialize compareBuffer			
15				1	
15	comp	pareBuffer = 81 11 22 33 44 F4			I

	<u>16</u>	Initialise compareBuffer		
		compareBuffer = 81 11 22 33 44 F4	D. H. L.	
1 40		Compare buffers with same parameters	Result is +1	
16	aomr	Initialize compareBuffer		
	17	Initialise compareBuffer		
	17	compareBuffer = 81 11 22 33 44 F6		
•		Compare buffers with same parameters	Result is -1	
17		Initialize compareBuffer		
	com	pareBuffer =		
		55 55 22 33 44 F5 55 55 55 55		
	<u>18</u>	<u>Initialise compareBuffer</u> compareBuffer =		
		55 55 55 22 33 44 F5 55 55 55		
•		Compare buffers	Result is 00h	
		valueOffset = 2		
		compareOffset = 3		
18	_	compareLength = 4 Initialize compareBuffer		
1170	com	pareBuffer =		
	55 5	55 55 22 33 45 F5 55 55 55 55		
	<u>19</u>	Initialise compareBuffer		
		<pre>compareBuffer = 55 55 55 22 33 45 F5 55 55 55</pre>		
I		Compare buffers with same parameters	Result is -1	
19	Η	Initialize compareBuffer	IVESUIT IS - I	
13	comp	pareBuffer =		
		55 55 22 33 43 F5 55 55 55 55		
	<u>20</u>	Initialise compareBuffer		
		compareBuffer =		
		55 55 55 22 33 43 F5 55 55 55 55 Compare buffers with same parameters	Result is +1	
20	Η	Initialize compareBuffer	IVESUIT IS +1	
20	comp	pareBuffer =		
	83 (31 55 55 55 55 55 55 55 55 55		
	<u>21</u>	Initialise compareBuffer		
		<pre>compareBuffer = 83 81 55 55 55 55 55 55 55 55 55 55</pre>		
I		findAndCompareValue()	Result is 00h	
		tag = 02h, occurrence = 1	1100011100011	
		valueOffset = 0		
		compareOffset = 0		
21	Η-	compareLength = 2 Initialize compareBuffer		
	comp	pareBuffer =		
	22	14 55 55 55 55 55 55 55 55 55		
	<u>22</u>	Initialise compareBuffer		
		<u>compareBuffer =</u> 22 44 55 55 55 55 55 55 55 55 55		
Į		findAndCompareValue()	Result is 00h	
		tag = 02h, occurrence = 2	result is con	
		valueOffset = 0		
		compareOffset = 0		
22	_	compareLength = 2 Initialize compareBuffer		
==	com	oareBuffer =		
	22	15 55 55 55 55 55 55 55 55 55		
	<u>23</u>	Initialise compareBuffer		
		<pre>compareBuffer =</pre>		
I		22 45 55 55 55 55 55 55 55 55 55 55 55 55	Result is -1	
		tag = 02h, occurrence = 2	Result is - i	
		valueOffset = 0		
		compareOffset = 0		
23	Η	compareLength = 2 Initialize compareBuffer		
🕁	COM	mmanze oumparebuner pareBuffer =		
	83 8	31 55 55 55 55 55 55 55 55 55 55		
'	<u>24</u>	Initialise compareBuffer		
		compareBuffer =		
I		83 81 55 55 55 55 55 55 55 55 55 55 55 55 55	Result is 00h	
		tag = 02h, occurrence = 1	IVESUIT IS ONLY	
	1	1	I	- I

		valueOffset = 0				
		compareBuffer.length = 12				
		compareOffset = 0				
		compareLength = 2				
24		Initialize compareBuffer				
ll .	comp	pareBuffer = 01 02 C4				
	25	Initialise compareBuffer				
		compareBuffer = 01 02 C4				
		Successful call (with tag B3h)		Result is 00h		
		tag = B3h, occurrence = 1				
		<pre>valueOffset = 0</pre>				
		compareBuffer.length = 00C4				
		compareOffset = 0				
		compareLength = 00C4				
25	Su		Resu	ult of findAndCompareValue()		
		length =0	is 00	h		
	DstE	Buffer.length = C4				
	Dst(Offset = C4				
		ength = 0				
	<u> 26</u>	Successful call, findAndCompareValue wi	th	Result of findAndCompareVal	<u>ue()</u>	
		<u>length =0</u>		<u>is 00h</u>		
		DstBuffer.length = C4				
		<pre>DstOffset = C4</pre>				
<u> </u>		DstLength = 0				
		Send Formatted SMS PP with the maximu	m			
		user data length = 0x010D, using 2				
		concatenated envelopes				
		Initialise compareBuffer				
		CompareBuffer = 23.048 Header + secure	d			
		data (01 02 FA)				
	27	Successful call (with SMS TPDU TAG)		Result is 00h		
		tag = 0Bh, occurrence = 1				
		valueOffset = 0x11				
		<pre>compareBuffer.length = 0x010D</pre>				
		<pre>compareOffset = 0</pre>				
		compareLength = 0x010D				
		Initialise compareBuffer	_			
		CompareBuffer = 55 55 55 EE EF F0 F				
	20	F2 F3 F4 F5 F6 F7 F8 F9 FA	-	Describie 00h		
	<u>28</u>	Successful call (with SMS TPDU TAG)		Result is 00h		
		tag = 0Bh, occurrence = 1 valueOffset = 0x11				
		compareBuffer.length = 0x010D				
		compareOffset = 0x0100				
		compareLength = 0x0D				
l [COMPATCHCHIGHT - UNUD				

6.2.4.17.4 Test Coverage

CRR number	Test case num	ber
N1	12	
N2	14	
N3	13, 17, 20, 21,	-25
N4	19, 15	
N5	16, 18, 22	
N6	23, 24	
<u>N1</u>	<u>12, 13</u>	
<u>N2</u>	<u>15</u>	
<u>N3</u>	14, 18, 21, 22, 26,	27, 28
<u>N4</u>	<u>16, 20</u>	
<u>N5</u>	<u>17, 19, 23</u>	
<u>N6</u>	<u>24, 25</u>	
P1	1	
P2	2, 3, 4, 5, 6	i
P3	7, 8, 9, 10	
P4	11	
C1	Does not apply	
	EnvelopeHand	dler

6.2.4.18 Method getCapacity

Test Area Reference: API_2_ENH_GCAP

6.2.4.18.1 Conformance Requirement:

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

public byte getCapacity()

6.2.4.18.1.1 Normal execution

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

6.2.4.18.2 Test suite files

Test Script: API 2 ENH GCAP 1.scr
Test Applet: API 2 ENH GCAP 1.java
Load Script: API 2 ENH GCAP 1.ldr
Cleanup Script: API 2 ENH GCAP 1.clr
Parameter File: API 2 ENH GCAP 1.par

6.2.4.18.3 Test Procedure

ld	<u>Description</u>	API Expectation	APDU Expectation
<u>1</u>	EnvelopeHandler available		
	1 - Send envelope SMS-PP Formatted	1 - Applet is triggered	
	2 - The applet calls the getLength() method	2 - No exception is thrown	
	3 - The applet calls getCapacity()method	3 - No exception is thrown; the	
		capacity is greater than the BER	
		TLV Length	

6.2.4.18.4 Test Coverage

CRR number	Test case number
<u>N1</u>	<u>1</u>

6.2.4.19 Method getUserDataLength

Test Area Reference: API_2_ENH_GUDL

6.2.4.19.1 Conformance Requirement:

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

public short getUserDataLength()

6.2.4.19.1.1 Normal execution

CRRN1: The method shall return the length of the User Data contained in the SMS TPDU TLV element.

CRRN2: The length is from the first SMS TPDU TLV element.

CRRN3: If the SMS TPDU TLV element is available, it becomes the selected TLV

CRRN4: The method can be used if the event is EVENT_FORMATTED_SMS_PP_ENV.

CRRN5: The method can be used if the event is EVENT FORMATTED SMS PP UPD.

CRRN6: The method can be used if the event is EVENT UNFORMATED SMS PP ENV.

CRRN7: The method can be used if the event is EVENT_UNFORMATTED_SMS_PP_UDP.

6.2.4.19.1.2 Context errors

CRRC1: The method shall throw UNAVAILABLE ELEMENT in case of unavailable TPDU TLV element.

CRRC2: The method shall throw UNAVAILABLE ELEMENT in case of wrong data format.

6.2.4.19.2 Test suite files

Specific triggering:

- UNFORMATTED SMS PP ENV
- FORMATTED_SMS_PP_UPD
- UNFORMATED_SMS_PP_UPD
- <u>UNRECOGNIZED_ENVELOPE</u>
- For Formatted triggering if CC/RC/DS is used, the security parameters are those used for downloading applications.

Test Script: API_2_ENH_GUDL_1.scr

Test Applet: API_2_ENH_GUDL_1.java

Load Script: API 2 ENH GUDL 1.ldr

Cleanup Script: API_2_ENH_GUDL_1.clr

Parameter File: API_2_ENH_GUDL_1.par

6.2.4.19.3 Test Procedure

ld	Description	API Expectation	APDU Expectation
	FORMATTED SMS PP ENV Triggering		
1	Test with FORMATTED_SMS_PP_ENV and TP-	Returns 0x003D	
	OA length of 2 and user data length of 0x3D		
2	Test with TP-OA length of 12 and user data length	Returns 0x003D	
	of 0x3D		
<u>3</u>	Test with RC/CC/DS length of 0 and secured data	Returns 0x0023	
	length of 0x10		
<u>4</u>	Test with RC/CC/DS length of 8 and secured data	Returns 0x002B	
	length of 0x10		
<u>5</u>	Test with PCNTR = 0, no RC/CC/DS and data	Returns 0x0023	
	length of 0x10		
<u>6</u>	Test with PCNTR = 7, no RC/CC/DS and data	Returns 0x001F	
	length of 0x05		
<u>7</u>	Test with SecuredDataLength = 00 and no	Returns 0x0013	
	RC/CC/DS		
<u>8</u>	Test with UserDataLength = 0x7F	Returns 0x007F	
9	Test with UserDataLength = 0x80	Returns 0x0080	
<u>10</u>	<u>Test with UserDataLength = maximum length</u>	Returns 0x008C	
	(0x8C) for a single SMS		
<u>11</u>	Verify it is the first TPDU TLV:	Returns 0x0018	
	Send a SMS PP with 2 TPDU TLV with two		
	different user data lengths: 0x18 and 0x23		
<u>12</u>	Send envelope SMS-PP Formatted.	GetValueByte() returns 0x40(23.040	
	FindTLV() with TAG_DEVICE_IDENTITIES.	<u>first byte)</u>	

	GetUserDataLength() and then getValueByte()		
	with offset 0		
<u>13</u>	Test with UserDataLength = 0xFF with 2	Returns 0x00FF	
	concatenated SMS		
14	Test with UserDataLength = 0x100 with 2	Returns 0x0100	
	concatenated SMS		
15	Test with UserDataLength = maximum length	Returns 0x010D	
<u></u>	(0x010D) with 2 concatenated SMS	TROCATION ONO TOP	
	FORMATTED SMS PP UPD Triggering		
16	Test with FORMATTED_SMS_PP_UPD and TP-	Returns 0x003D	
10	OA length of 2 and user data length of 0x3D	Returns 0x003D	
47		Deturne 0x000D	
<u>17</u>	Test with TP-OA length of 12 and user data length	Returns 0x003D	
	of 0x3D		
<u>18</u>		Returns 0x0023	
	length of 0x10		
<u>19</u>	Test with RC/CC/DS length of 8 and secured data	Returns 0x002B	
	length of 0x10		
<u>20</u>	Test with PCNTR = 0, no RC/CC/DS and data	Returns 0x0023	
	length of 0x10		
21	Test with PCNTR = 7, no RC/CC/DS and data	Returns 0x001F	
L	length of 0x05	<u> </u>	
22	Test with SecuredDataLength = 00 and no	Returns 0x0013	
	RC/CC/DS		
23	Test with UserDataLength = 0x7F	Returns 0x007F	
	Test with UserDataLength = 0x80	Returns 0x0080	
25	Test with UserDataLength = maximum	Returns 0x008C	
20	length(0x8C) for a single SMS	TCCCCTTS OXOGOO	
26	Verify it is the first TPDU TLV:	Returns 0x0018	
20	Send a SMS PP with 2 TPDU TLV with two	Keturis 0x0016	
07	different user data lengths: 0x18 and 0x23	O-t)/-lD-t()t	
<u>27</u>	Send envelope SMS-PP Formatted.	GetValueByte() returns 0x40(23.040	
	FindTLV() with TAG_DEVICE_IDENTITIES.	first byte)	
	GetUserDataLength() and then getValueByte()		
	with offset 0		
<u>28</u>	Test with UserDataLength = 0xFF with 2	Returns 0x00FF	
	Iconcatonated SMS		
L	concatenated SMS		
<u>29</u>	Test with UserDataLength = 0x100 with 2	Returns 0x0100	
<u>29</u>	Test with UserDataLength = 0x100 with 2 concatenated SMS	Returns 0x0100	
<u>29</u> <u>30</u>	Test with UserDataLength = 0x100 with 2 concatenated SMS	Returns 0x0100 Returns 0x010D	
	Test with UserDataLength = 0x100 with 2		
	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS		
30	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering	Returns 0x010D	
30	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and		
<u>30</u>	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D	Returns 0x010D Returns 0x003D	
30	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length	Returns 0x010D Returns 0x003D	
30 31 32	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D	Returns 0x010D Returns 0x003D Returns 0x003D	
30 31 32 33	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00	Returns 0x000D Returns 0x0003D Returns 0x0003D Returns 0x00000	
30 31 32 33 34	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F	Returns 0x010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F	
30 31 32 33 34 35	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80	Returns 0x010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080	
30 31 32 33 34	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length:	Returns 0x010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F	
30 31 32 33 34 35 36	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS	Returns 0x010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080 Returns 0x008C	
30 31 32 33 34 35	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV:	Returns 0x010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080	
30 31 32 33 34 35 36	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two	Returns 0x010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080 Returns 0x008C	
30 31 32 33 34 35 36	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23	Returns 0x010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080 Returns 0x008C	
30 31 32 33 34 35 36	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted.	Returns 0x010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080 Returns 0x008C Returns 0x0018 GetValueByte() returns 0x00	
30 31 32 33 34 35 36 37	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted. FindTLV() with TAG_DEVICE_IDENTITIES.	Returns 0x010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080 Returns 0x008C Returns 0x0018	
30 31 32 33 34 35 36 37	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted. FindTLV() with TAG_DEVICE_IDENTITIES. GetUserDataLength() and then getValueByte()	Returns 0x010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080 Returns 0x008C Returns 0x0018 GetValueByte() returns 0x00	
30 31 32 33 34 35 36 37	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED_SMS_PP_ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted. FindTLV() with TAG_DEVICE_IDENTITIES. GetUserDataLength() and then getValueByte()	Returns 0x010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080 Returns 0x008C Returns 0x0018 GetValueByte() returns 0x00	
30 31 32 33 34 35 36 37	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED SMS PP ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted. FindTLV() with TAG DEVICE IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0 (first user data = 0x55)	Returns 0x010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080 Returns 0x008C Returns 0x0018 GetValueByte() returns 0x00	
30 31 32 33 34 35 36 37	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED SMS PP ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted. FindTLV() with TAG DEVICE IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0 (first user data = 0x55) UNFORMATTED SMS PP UPD Triggering	Returns 0x010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080 Returns 0x008C Returns 0x0018 GetValueByte() returns 0x00	
30 31 32 33 34 35 36 37	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED SMS PP ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted. FindTLV() with TAG DEVICE IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0 (first user data = 0x55) UNFORMATTED SMS PP UPD Triggering Test with UNFORMATTED_SMS_PP_UPD and	Returns 0x0010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080 Returns 0x008C Returns 0x0018 GetValueByte() returns 0x00 (23.040 first byte)	
30 31 32 33 34 35 36 37 38	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED SMS PP ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted. FindTLV() with TAG DEVICE IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0 (first user data = 0x55) UNFORMATTED SMS PP UPD Triggering Test with UNFORMATTED SMS PP UPD and TP-OA length of 2, and user data length of 0x3D	Returns 0x0010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080 Returns 0x008C Returns 0x0018 GetValueByte() returns 0x00 (23.040 first byte) Returns 0x003D	
30 31 32 33 34 35 36 37	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED SMS PP ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted. FindTLV() with TAG DEVICE IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0 (first user data = 0x55) UNFORMATTED SMS PP UPD Triggering Test with UNFORMATTED_SMS PP_UPD and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length	Returns 0x0010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080 Returns 0x008C Returns 0x0018 GetValueByte() returns 0x00 (23.040 first byte) Returns 0x003D	
30 31 32 33 34 35 36 37 38 40	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED SMS PP ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted. FindTLV() with TAG DEVICE IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0 (first user data = 0x55) UNFORMATTED SMS PP UPD Triggering Test with UNFORMATTED SMS PP UPD and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D	Returns 0x0010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x008C Returns 0x008C Returns 0x0018 GetValueByte() returns 0x00 (23.040 first byte) Returns 0x003D Returns 0x003D	
30 31 32 33 34 35 36 37 38 40 41	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED SMS PP ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted. FindTLV() with TAG DEVICE IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0 (first user data = 0x55) UNFORMATTED SMS PP UPD Triggering Test with UNFORMATTED SMS PP UPD and TP-OA length of 2, and user data length of 0x3D Test with UserDataLength = 0x00	Returns 0x0010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x0080 Returns 0x008C Returns 0x0018 GetValueByte() returns 0x00 (23.040 first byte) Returns 0x003D Returns 0x003D Returns 0x003D Returns 0x00000	
30 31 32 33 34 35 36 37 38 40 41 42	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED SMS PP ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted. FindTLV() with TAG DEVICE IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0 (first user data = 0x55) UNFORMATTED SMS PP UPD Triggering Test with UNFORMATTED SMS PP UPD and TP-OA length of 2, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F	Returns 0x0010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x008C Returns 0x008C Returns 0x0018 GetValueByte() returns 0x00 (23.040 first byte) Returns 0x003D Returns 0x003D Returns 0x00000 Returns 0x0007F	
30 31 32 33 34 35 36 37 38 40 41 42 43	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED SMS PP ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted. FindTLV() with TAG DEVICE IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0 (first user data = 0x55) UNFORMATTED SMS PP UPD Triggering Test with UNFORMATTED SMS PP UPD and TP-OA length of 2, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x00 Test with UserDataLength = 0x80	Returns 0x0010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x008C Returns 0x008C Returns 0x0018 GetValueByte() returns 0x00 (23.040 first byte) Returns 0x003D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x0007F Returns 0x0007F Returns 0x00080	
30 31 32 33 34 35 36 37 38 40 41 42	Test with UserDataLength = 0x100 with 2 concatenated SMS Test with UserDataLength = maximum length (0x010D) with 2 concatenated SMS UNFORMATTED SMS PP ENV Triggering Test with UNFORMATTED SMS PP ENV and TP-OA length of 2, and user data length of 0x3D Test with TP-OA length of 12, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F Test with UserDataLength = 0x80 Test with UserDataLength = maximum length: 0x8C for a single SMS Verify it is the first TPDU TLV: Send a SMS PP with 2 TPDU TLV with two different user data lengths: 0x18 and 0x23 Send envelope SMS-PP Unformatted. FindTLV() with TAG DEVICE IDENTITIES. GetUserDataLength() and then getValueByte() with offset 0 (first user data = 0x55) UNFORMATTED SMS PP UPD Triggering Test with UNFORMATTED SMS PP UPD and TP-OA length of 2, and user data length of 0x3D Test with UserDataLength = 0x00 Test with UserDataLength = 0x00 Test with UserDataLength = 0x7F	Returns 0x0010D Returns 0x003D Returns 0x003D Returns 0x0000 Returns 0x007F Returns 0x008C Returns 0x008C Returns 0x0018 GetValueByte() returns 0x00 (23.040 first byte) Returns 0x003D Returns 0x003D Returns 0x00000 Returns 0x0007F	

45	Verify it is the first TPDU TLV:	Returns 0x0018	
	Send a SMS PP with 2 TPDU TLV with two		
	different user data lengths: 0x18 and 0x23		
	Send envelope SMS-PP Unformatted.	GetValueByte() returns 0x00	
	FindTLV() with TAG_DEVICE_IDENTITIES.	(23.040 first byte)	
	GetUserDataLength() and then getValueByte()		
	with offset 0		
	UNRECOGNIZED ENVELOPE Triggering		
47	Test with an UNRECOGNIZED_ENVELOPE	ToolkitException	
		UNAVAILABLE ELEMENT	

6.2.4.19.4 Test Coverage

CRR number	Test case number
<u>N1</u>	All test cases excepted:
	<u>53</u>
<u>N2</u>	<u>11, 26, 37, 45</u>
<u>N3</u>	<u>12, 27, 38, 46</u>
<u>N4</u>	<u>1 to 15</u>
<u>N5</u>	<u>16 to 30</u>
<u>N6</u>	31 to 38
<u>N7</u>	39 to 46
<u>C1</u>	47
<u>C2</u>	Not applicable

6.2.4.20 Method getChannelldentifier

Test Area Reference: API_2_ENH_GCID

6.2.4.20.1 Conformance Requirement:

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

public byte getChannelIdentifier()

throws ToolkitException

6.2.4.20.1.1 Normal execution

CRRN1: The method shall return the channel identifier byte value.

CRRN2: The channel identifier byte value returned shall be from the first Channel status TLV element.

CRRN3: If the element is available it becomes the currently selected TLV.

<u>CRRN4</u>: The channel identifier is available for all triggered toolkit applets from the invocation to the termination of their processToolkit method if the EnvelopeHandler is available.

6.2.4.20.1.2 Context errors

<u>CRRC1</u>: The method shall throw ToolkitException (UNAVAILABLE_ELEMENT) if the Channel status TLV is <u>not present.</u>

<u>CRRC2</u>: The method shall throw ToolkitException (OUT_OF_TLV_BOUNDARIES) if the Simple TLV Channel Status length is equal to 0.

6.2.4.20.2 Test suite files

Test Script:	API_2_ENH_GCID_1.scr
Test Applet:	API_2_ENH_GCID_1.java
Load Script:	API_2_ENH_GCID_1.ldr
Cleanup Script:	API 2 ENH GCID 1.clr

Parameter File: API 2 ENH GCID 1.par

6.2.4.20.3 Test Procedure

ld	<u>Description</u>	API Expectation	APDU Expectation
0	1- Applet1 is installed with maximum number of		2- OPEN CHANNEL
_	<pre>channel = 07.</pre>		proactive command is
	2- Applet1 builds proactive commands OPEN CHANNEL	-	fetched
	with init() method in order to open all channels. ProactiveHandler.send() methods are called.		
	ProactiveHandler.send() methods are carred.		TERMINAL RESPONSE is
			issued with Channel Id from
			01 to 07
1	1- Send envelope Event Download Channel Status	1- Applet1 is triggered	
	with channel status TLV:		
	channel status value = 0x8100.		
	2- Call EnvelopeHandler.getChannelIdentifier()	2- Returns 0x01	
	method		
2	1- Send envelope Event Download Channel Status	2- Returns 0x04	
=	with two channel status TLV:	Returns 0x04	
	first value = 0x8400	Totallio oxo i	
	second value = 0x8500.		
	0 0 17 1 1		
	2- Call twice the EnvelopeHandler.getChannelIdentifier() method		
3	1- Send envelope Event Download Channel Status	2- Returns 0x06	
2	with channel status TLV:	2- Returns 0x00	
	Channel Status value = 0x0605	3- GetChannelldentifier()	
		=getValueByte(0)	
	ViewHandler.FindTLV() with Device IdentityTag.	<u>-gervalueByte(0)</u>	
	O Call Brown and an all and an and Channel I don't if i and i		
	<pre>2- Call EnvelopeHandler.getChannelIdentifier() method.</pre>		
	meeriou.		
	3- Compare EnvelopeHandler.getChannelIdentifier()		
	and then ViewHandler.getValueByte(0).		
<u>4</u>	1- Send envelope Menu Selection without Channel	2- A Toolkit exception	
	Status TLV.	UNAVAILABLE ELEMEN	
	2- Call EnvelopeHandler.getChannelIdentifier()	T is thrown.	
	method.		
5	1- Send Envelope Event Download Channel Status	1- Returns 0x06	
	with Channel Status TLV:		
	Channel status value = 0x0600		
	2- Call EnvelopeHandler.getChannelIdentifier()		
6	method. 1- Send unrecognized envelope with a Channel	2. A Tablist expension	
<u>6</u>	Status TLV having a length equal to 0.	2- A Toolkit exception OUT OF TLV BOUNDA	
	TITLE IN MATERIA & TONGON CHARLE CO V.		
	2- Call EnvelopeHandler.getChannelIdentifier()	RIES is thrown.	
	method.		

6.2.4.20.4 Test Coverage

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

6.2.5 Class EnvelopeResponseHandler

6.2.5.21 Method getCapacity

Test Area Reference: API_2_ERH_GCAP

6.2.5.21.1 Conformance Requirement:

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

Public byte getCapacity()

6.2.5.21.1.1 Normal execution

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

<u>6.2.5.21.1.2</u> Context errors

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

6.2.5.21.2 Test suite files

Test Script: API_2_ERH_GCAP_1.scr

Test Applet: API 2 ERH GCAP 1.java

Load Script: API 2 ERH GCAP 1.ldr

Cleanup Script: API_2_ERH_GCAP_1.clr

Parameter File: API_2_ERH_GCAP_1.par

6.2.5.21.3 Test Procedure

ld	<u>Description</u>	API Expectation	APDU Expectation
1	EnvelopeResponseHandler available		
	1- Send envelope SMS-PP Formatted	1- Applet is triggered	
	2- The applet calls getTheHandler() method	2- No exception is thrown	
	3- The applet calls getCapacity() method on the	3- No exception is thrown	
	EnvelopeResponseHandler		
	4- The applet fills the handler with the maximum	4- No exception is thrown	
	capacity using AppendTLV() method	5- No exception is thrown	
	5- The applet calls clear() method on the		
	EnvelopeResponseHandler	6- HANDLER_OVERFLOW	
	6- The applet fills the handler with the maximum	exception is thrown	
	capacity plus one, using AppendTLV() method		

6.2.5.21.4 Test Coverage

CRR number	Test case number	
<u>N1</u>	1	
<u>C1</u>	Tested in Framework	
	part: FWK MHA ERHD	

6.2.6 Class ProactiveHandler

6.2.7.3 Method initDisplayText

Test Area Reference: API_2_PAH_INDTBB_BSS-

6.2.7.3.1 Conformance requirement

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

short offset,
 short length)
throws java.lang.NullPointerException,
 java.lang.ArrayIndexOutOfBoundsException,
 ToolkitException

6.2.7.3.1.1 Normal execution

CRRN1: The method shall build a DISPLAY TEXT proactive command in the ProactiveHandler, using qualifier, dcs and buffer parameters. Comprehension required flags are set.

-CRRN2: A call to this method clears the handler then initializes initialises it.

CRRN3: No TLV is selected after a call to the method.

CRRN4: The DISPLAY TEXT command is not sent by the method.

CRRN5: The Command Number may take any value between 01h and FEh.

CRRN6: If length is equal to zero, then the Text String TLV inserted in the command is a null text string TLV as defined in TS 11.14 [4].

6.2.7.3.1.2 Parameter errors

CRRP1: The method shall throw NullPointerException if buffer is null.

CRRP2: If offset or length or both would cause access outside array bounds, an ArrayIndexOutOfBoundsException shall be thrown.

6.2.7.3.1.3 Context errors

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

6.2.7.3.2 Test Suite files

Test Script: _____API_2_PAH_INDTBB_BSS_1.scr

Test Applet: API_2_PAH_INDTBB_BSS_1.java

Load Script: API_2_PAH_INDTBB_BSS_1.ldr

Cleanup Script: API_2_PAH_INDTBB_BSS_1.clr

Parameter File: API_2_PAH_INDTBB_BSS_1.java

Load Script: API_2_PAH_INDTBB_BSS_1.ldr

Cleanup Script: API_2_PAH_INDTBB_BSS_1.ldr

Cleanup Script: API_2_PAH_INDTBB_BSS_1.clr

Parameter File: API_2_PAH_INDTBB_BSS_1.par

6.2.7.3.3 Test procedure

ld	Description	API Expectation	APDU Expectation
1	NULL as parameter to buffer	NullPointerException is thrown	, p. 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
	buffer = NULL		
2	offset > buffer.length buffer = "Text"	ArrayIndexOutOfBoundsExceptio	
	offset = 5	n is thrown	
	length = 0		
3	offset < 0	ArrayIndexOutOfBoundsExceptio	
	<pre>buffer = "Text" offset = -1</pre>	n is thrown	
4	length > buffer.length	ArrayIndexOutOfBoundsExceptio	
'	buffer = "Text"	n is thrown	
	offset = 0		
5	length = 5 offset + length > buffer.length	ArrayIndexOutOfBoundsExceptio	
3	buffer = "Text"	n is thrown	
	offset = 3	THO UNIOWIT	
	length = 2	A	
6	length < 0 buffer = "Text"	ArrayIndexOutOfBoundsException is thrown	
	offset = 3	IT IS UTIOWIT	
	length = -1		
7	Successful call, buffer is the whole buffer	No exception is thrown	
	qualifier = 0 dcs = 4		
	buffer = "TextA"		
	offset = 0		
	length = 5 Verify the command number value	Command number between 01h	
	verify the command number value	and FEh	
8	Send the command	and i zii	DISPLAY TEXT Proactive
			command
			1.5.
			qualifier = 00h dcs = 4
			Text = "TextA"
9	Successful call, buffer is part of a buffer with		DISPLAY TEXT Proactive
	the end part		command
	Send the command qualifier = 0		qualifier = 00h
	dcs = 4		dcs = 4
	buffer = "12TextB"		Text = "TextB"
	offset = 2 length = 5		
10	Successful call, buffer is part of a buffer with		DISPLAY TEXT Proactive
	the first part		command
	Send the command		7.1.5.1
	qualifier = 0 dcs = 4		qualifier = 00h dcs = 4
	buffer = "TextC12"		Text = "TextC"
	offset = 0		
11	length = 5		DISPLAY TEXT Proactive
11	Successful call, buffer is part of a buffer Send the command		command
	qualifier = 0		Communa
	dcs = 4		qualifier = 00h
	<pre>buffer = "12TextD34" offset = 2</pre>		dcs = 4 Text = "TextD"
	length = 5		TCAC - TCACD
12	Successful call, qualifier = 81h		DISPLAY TEXT Proactive
	Send the command		command
1	qualifier = 81h dcs = 4		qualifier = 81h
1	buffer = "TextE"		dcs = 4
	offset = 0		Text = "TextE"
40	length = 5		DICDLAY TEXT December
13	Successful call, DCS=0 (7 bits) Send the command		DISPLAY TEXT Proactive command
1	qualifier = 0		Continana
	dcs = 0		qualifier = 00h
	buffer = "TextF"		dcs = 0
	offset = 0		Text = "TextF"

	length = 5		
14	Successfull call, DCS=8 (UCS2)		DISPLAY TEXT Proactive
	Send the command		command
	qualifier = 0		
	dcs = 8		qualifier = 00h
	<pre>buffer = "TextG" offset = 0</pre>		dcs = 8 Text = "TextG"
	length = 5		lext - lextG
15	Call the initDisplayText() method with any		DISPLAY TEXT Proactive
10	value		command
	Then build and send a DISPLAY TEXT		Command
	command		qualifier = 00h
	qualifier = 0		dcs = 4
	dcs = 4		Text = "TextHTextH"
	<pre>buffer = "TextHTextH" offset = 0</pre>		
	length = 10		
	Successful call, text length is null		DISPLAY TEXT Proactive
	Send the command		command
qual	ifier = 0		
des	= 4		qualifier = 00h
of fo	er = "" (not null buffer)		rext String TLV = 8D 00
leng	tth = 0		
<u>16</u>	Successful call, text length is zero		DISPLAY TEXT Proactive
	Send the command		command
	qualifier = 0		
	<pre>dcs = 4 buffer = "TextHTextH"</pre>		<pre>qualifier = 00h Text String TLV = 8D 00</pre>
	offset = 0		Tene berring THV - 0D 00
	length = 0		
17	Select a TLV in the ProactiveHandler	UNAVAILABLE_ELEMENT	
	Call the initDisplayText() method	ToolkitException is thrown by	
	Call the getValueLength() method	getValueLength()	
18	Successful call, buffer length = 7Eh		DISPLAY TEXT Proactive command
	qualifier = 0		Command
	dcs = 4		Text String TLV =
	buffer = "UUU"		8D 7F 04 55 55
	offset = 0 length = 7Eh		
19	Successful call, buffer length = 7Fh		DISPLAY TEXT Proactive
	qualifier = 0		command
	dcs = 4		Text String TLV = 8D 81
	<pre>buffer = "UUU" offset = 0</pre>		80 04 55 55
	offset = 0 length = 7Fh		
20	Successful call, buffer length = 240		DISPLAY TEXT Proactive
	_		command
	Qualifier = 0		
	dcs = 4 buffer = "UUU"		Text String TLV = 8D 81 F1 04 55 55
	offset = 0		0 01 FI 04 00 00
	length = 240		
21	Call the initDisplayText() method with a too	HANDLER_OVERFLOW	
	long buffer	ToolkitException is thrown	
	qualifier = 0 dcs = 4		
	buffer = "XXXX"		
	offset = 0		
	length = 241		
22	Call the initDisplayText() without sending the		No proactive command
	command		shall be sent expected
			status is '9000'

6.2.7.3.4 Test Coverage

CRR number	Test case number
N1	8, 9, 10, 11, 12, 13, 14,
	15, 16, 18, 19, 20
N2	15
N3	17
N4	22
N5	7
<u>N6</u>	<u>16</u>
P1	1
P2	2, 3, 4, 5, 6
C1	21

6.2.7.4 Method initGetInkey

Test Area Reference: API_2_PAH_INGKBB_BSS-

6.2.7.4.1 Conformance requirement

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

6.2.7.4.1.1 Normal execution

CRRN1: The method shall build a GET INKEY proactive command in the ProactiveHandler, using qualifier, dcs and buffer parameters. Comprehension Required flags are set.

—CRRN2: A call to this method clears the handler then initializes initialises it.

CRRN3: No TLV is selected after a call to the method.

CRRN4: The GET INKEY command is not sent by the method.

CRRN5: The Command Number may take any value between 01h and FEh.

CRRN6: If length is equal to zero, then the Text String TLV inserted in the command is a null text string TLV as defined in TS 11.14 [4].

6.2.7.4.1.2 Parameter errors

CRRP1: The method shall throw NullPointerException if buffer is null.

CRRP1: If offset or length or both would cause access outside array bounds, a ArrayIndexOutOfBoundsException shall be thrown.

6.2.7.4.1.3 Context errors

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

6.2.7.4.2 Test Suite files

Test Script: _____API_2_PAH_INGKBB_BSS_1.scr

Test Applet: API 2 PAH INGKBB BSS 1.java

 Load Script:
 API_2_PAH_INGKBB_BSS_1.ldr

 Cleanup Script:
 API_2_PAH_INGKBB_BSS_1.elr

 Parameter File:
 API_2_PAH_INGKBB_BSS_1.java

 Load Script:
 API_2_PAH_INGKBB_BSS_1.ldr

 Cleanup Script:
 API_2_PAH_INGKBB_BSS_1.elr

 Parameter File:
 API_2_PAH_INGKBB_BSS_1.par

6.2.7.4.3 Test procedure

ld	Description	API Expectation	APDU Expectation
1	NULL as parameter to buffer	NullPointerException is thrown	Al Do Expediation
'	buffer = NULL	Train offici Exception is thrown	
2	offset > buffer.length	ArrayIndexOutOfBoundsExceptio	
	buffer = "Text"	n is thrown	
	offset = 5		
3	offset < 0	ArrayIndexOutOfBoundsExceptio	
	<pre>buffer = "Text" offset = -1</pre>	n is thrown	
4	length > buffer.length	ArrayIndexOutOfBoundsExceptio	
4	buffer = "Text"	n is thrown	
	offset = 0	II IS UIIOWII	
	length = 5		
5	offset + length > buffer.length	ArrayIndexOutOfBoundsExceptio	
	<pre>buffer = "Text"</pre>	n is thrown	
	offset = 3		
6	length = 2	ArrayIndexOutOfBoundsExceptio	
٥	buffer = "Text"	n is thrown	
	offset = 3	I I I I I I I I I I I I I I I I I I I	
	length = -1		
7	Successful call, buffer is the whole buffer	No exception is thrown	
	qualifier = 0		
	<pre>dcs = 4 buffer = "TextA"</pre>		
	offset = 0		
	length = 5		
	Verify the command number value	Command number between 01h	
	-	and FEh	
8	Send the command		GET INKEY Proactive
			command
			7.1.5.1
			qualifier = 00h dcs = 4
			Text = "TextA"
9	Successful call, buffer is part of a buffer with		GET INKEY Proactive
	the end part		command
	qualifier = 0		
	dcs = 4		qualifier = 00h
	<pre>buffer = "12TextB" offset = 2</pre>		<pre>dcs = 4 Text = "TextB"</pre>
	length = 5		TEXT - TEXTS
10	Successful call, buffer is part of a buffer with		GET INKEY -Proactive
	the first part		command
	qualifier = 0		
	dcs = 4		qualifier = 00h
	<pre>buffer = "TextC12" offset = 0</pre>		dcs = 4 Text = "TextC"
	length = 5		ICAC - ICACC
11	Successful call, buffer is part of a buffer		GET INKEY -Proactive
	Send the command		command
	qualifier = 0		
	dcs = 4		qualifier = 00h
	<pre>buffer = "12TextD34" offset = 2</pre>		dcs = 4 Text = "TextD"
	length = 5		TEXC = "TEXCD"
12	Successful call, qualifier = 81h		GET INKEY Proactive
	qualifier = 81h		command

		<pre>dcs = 4 buffer = "TextE"</pre>		qualifier = 81h
		offset = 0		dcs = 4 Text = "TextE"
		length = 5		Text = "TextE"
-	13	Successful call, DCS=0 (7 bits) gualifier = 0		GET INKEY Proactive
		dcs = 0		command
		<pre>buffer = "TextF" offset = 0</pre>		qualifier = 00h dcs = 0
		length = 5		Text = "TextF"
•	14	Successful call, DCS=8 (UCS2)		GET INKEY Proactive
	17	qualifier = 0		command
		<pre>dcs = 8 buffer = "TextG"</pre>		qualifier = 00h
		offset = 0 length = 5		dcs = 8 Text = "TextG"
		Tengen - 5		Text = "TextG"
	15	Call the initGetInkey() method with any value Then build and send a GET INKEY command		GET INKEY Proactive command
		qualifier = 0		
		<pre>dcs = 4 buffer = "TextHTextH"</pre>		qualifier = 00h dcs = 4
		offset = 0		Text = "TextHTextH"
16		Successful call, text length is null	GI	ET INKEY Proactive
	eus 1	Send the command	co	mmand
	des	= 4	_	alifier = 00h
	Dull	Ser = ""	Te	xt String TLV = 8D 00
1	leng	gth = 0		OFT INICEV Dropotice
	<u>16</u>	Successful call, text length is zero Send the command		GET INKEY Proactive command
		<pre>qualifier = 0 dcs = 4</pre>		qualifier = 00h
		<pre>buffer = "TextHTextH"</pre>		Text String TLV = 8D 00
		offset = 0 length = 0		
	17	Select a TLV in the ProactiveHandler	UNAVAILABLE_ELEMENT ToolkitException is thrown by	
		Call the initGetInkey() method Call the getValueLength() method	getValueLength()	
	40	Cusassful cell buffer length 75h		GET INKEY Proactive
	18	Successful call, buffer length = 7Eh		command
		qualifier = 0 dcs = 4		Text String TLV =
		buffer = "UUU" offset = 0		8D 7F 04 55 55
		length = 7Eh		
	19	Successful call, buffer length = 7Fh		GET INKEY Proactive command
		qualifier = 0		
		dcs = 4 buffer = "UUU"		Text String TLV = 8D 81 80 04 55 55
		offset = 0 length = 7Fh		
ŀ	20	Successful call, buffer length = 240		GET INKEY Proactive
		Qualifier = 0		command
		dcs = 4 buffer = "UUU"		Text String TLV = 8D 81 F1 04 55 55
		offset = 0		OD OI LI OF 33 33"
	21	length = 240 Call the initGetInkey() method with a too long	HANDLER_OVERFLOW	
		buffer	ToolkitException is thrown	
		qualifier = 0 dcs = 4		
		<pre>buffer = "XXXX" offset = 0</pre>		
	00	length = 241		No manage of the second of the
	22	Call the initGetInkey() without sending the command		No proactive command shall be sent expected
L			1	

	status is '9000'

6.2.7.4.4 Test Coverage

CRR number	Test case number
N1	8, 9, 10, 11, 12, 13, 14,
	15, 16,18, 19, 20
N2	15
N3	17
N4	22
N5	7
<u>N6</u>	<u>16</u>
P1	1
P2	2, 3, 4, 5, 6
C1	21

6.2.7.5 Method initGetInput

Test Area Reference: API_2_PAH_INGPBB_BSSSS-

6.2.7.5.1 Conformance requirement

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

6.2.7.5.1.1 Normal execution

CRRN1: The method shall build a GET INPUT proactive command in the ProactiveHandler, using qualifier, dcs, buffer, minRespLength and maxRespLength parameters. Comprehension Required flags are set.

—CRRN2: A call to this method clears the handler then initializes initialises it.

CRRN3: No TLV is selected after a call to the method.

CRRN4: The GET INPUT command is not sent by the method.

CRRN5: The Command Number may take any value between 01h and FEh.

CRRN6: If length is equal to zero, then the Text String TLV inserted in the command is a null text string TLV as defined in TS 11.14 [4].

6.2.7.5.1.2 Parameter errors

CRRP1: The method shall throw NullPointerException if buffer is null.

CRRP2: If offset or length or both would cause access outside array bounds, a ArrayIndexOutOfBoundsException shall be thrown.

6.2.7.5.1.3 Context errors

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

6.2.7.5.2 Test Suite files

Test Script: _____API_2_PAH_INGPBB_BSSSS_1.scr Test Applet: API_2_PAH_INGPBB_BSSSS_1.java Load Script: API_2_PAH_INGPBB_BSSSS_1.ldr Cleanup Script: API_2_PAH_INGPBB_BSSSS_1.clr Parameter File: API 2 PAH INGPBB BSSSS 1.java Load Script: API_2_PAH_INGPBB_BSSSS_1.ldr Cleanup Script: API_2_PAH_INGPBB_BSSSS_1.clr Parameter File: API_2_PAH_INGPBB_BSSSS_1.par

6.2.7.5.3 Test procedure

1	Description NULL as parameter to buffer	API Expectation	APDU Expectation
2		NullPointerException is thrown	
2	buffer = NULL	·	
	offset > buffer.length	ArrayIndexOutOfBoundsExceptio	
	buffer = "Text"	n is thrown	
	offset = 5		
3	offset < 0	ArrayIndexOutOfBoundsExceptio	
	buffer = "Text"	n is thrown	
	offset = -1		
4	length > buffer.length	ArrayIndexOutOfBoundsExceptio	
	buffer = "Text"	n is thrown	
	offset = 0		
	length = 5		
5	offset + length > buffer.length	ArrayIndexOutOfBoundsExceptio	
	buffer = "Text"	n is thrown	
	offset = 3		
	length = 2	1 1 2 20 2	
6	length < 0	ArrayIndexOutOfBoundsExceptio	
	buffer = "Text"	n is thrown	
	offset = 3		
7	length = -1 Successful call, buffer is the whole buffer	No expention is thrown	
1	Succession can, burier is the whole buller	No exception is thrown	
	qualifier = 0		
	dcs = 4		
	buffer = "TextA"		
	offset = 0		
	length = 5		
	minRespLength = 00h		
	maxRespLength = FFh		
	Verify the command number value Cor	mmand number between 01h	

	verify the command number value	and FEh	
8	Send the command		GET INPUT Proactive command
			qualifier = 00h dcs = 4 Text = "TextA" Min Length = 00h
9	Successful call, buffer is part of a buffer with		Max Length = FFh GET INPUT Proactive
	the end part		command
	Send the command		
	qualifier = 0		qualifier = 00h
	dcs = 4		dcs = 4
	buffer = "12TextB"		Text = "TextB"
	offset = 2		Min Length = 10h
	length = 5		Max Length = FFh
	minRespLength = 10h		
1	maxRespLength = FFh		

				,
	10	Successful call, buffer is part of a buffer with		GET INPUT -Proactive
		the first part		command
		Send the command		
		qualifier = 0		qualifier = 00h
		dcs = 4		dcs = 4
		buffer = "TextC12"		Text = "TextC"
		offset = 0		Min Length = FFh
		length = 5		Max Length = FFh
		minRespLength = FFh		
1	11	maxRespLength = FFh		CET INDIT Properties
	11	Successful call, buffer is part of a buffer Send the command		GET INPUT -Proactive
		qualifier = 0		command
		qualifier = 0 dcs = 4		qualifier = 00h
		buffer = "12TextD34"		dcs = 4
		offset = 2		Text = "TextD"
		length = 5		Min Length = 00h
		minRespLength = 00h		Max Length = 00h
		maxRespLength = 00h		
	12	Successful call, qualifier = 81h		GET INPUT Proactive
		qualifier = 81h		command
		dcs = 4		
		<pre>buffer = "TextE"</pre>		qualifier = 81h
		offset = 0		dcs = 4
		length = 5		Text = "TextE"
		minRespLength = 00h		Min Length = 00h
		maxRespLength = 10h		Max Length = 10h
-	40	C		OFF INDIT Describes
	13	Successful call, DCS=0 (7 bits) qualifier = 0		GET INPUT Proactive
		dcs = 0		command
		buffer = "TextF"		qualifier = 00h
		offset = 0		dcs = 0
		length = 5		Text = "TextF"
		minRespLength = 10h		Min Length = 10h
		maxRespLength = 10h		Max Length = 10h
	14	Successful call, DCS=8 (UCS2)		GET INPUT Proactive
		qualifier = 0		command
		dcs = 8		
		buffer = "TextG"		qualifier = 00h
		offset = 0		dcs = 8
		length = 5		Text = "TextG"
		minRespLength = 00h maxRespLength = FFh		Min Length = 00h
		maxrespheriger - FFII		Max Length = FFh
-	15	Call the initGetInput() method with any value		GET INPUT Proactive
1	15	Then build and send a GET INPUT -command		command
		qualifier = 0		Command
		dcs = 4		qualifier = 00h
		buffer = "TextHTextH"		dcs = 4
		offset = 0		Text = "TextHTextH"
		length = 10		Min Length = 00h
		minRespLength = 00h		Max Length = 10h
1 40		maxRespLength = 10h	105	INDUT Description
16		Successful call, text length is null		
	1	Send the command	COIT	mand
	dea	<u>- 4</u>	mia .	lifier = 00h
	buf f	Cer = ""	Texi	String TLV = 8D 00
	offs	set = 0	Min Min	Length = 00h
	lene	yth = 0	Max	Length = 10h
	minF	RespLength = 00h		
L	maxF	RespLength = 10h		OFT INDUT Desert
	<u>16</u>	Successful call, text length is zero		GET INPUT Proactive
		Send the command		command
		qualifier = 0		gualifier - 00h
		<pre>dcs = 4 buffer = "TextHTextH"</pre>		<pre>qualifier = 00h Text String TLV = 8D 00</pre>
		offset = 0		Min Length = 00h
		length = 0		Max Length = 10h
		minRespLength = 00h		
		maxRespLength = 10h		
	17	Select a TLV in the ProactiveHandler	UNAVAILABLE_ELEMENT	
		Call the initGetInput() method	ToolkitException is thrown by	
L			•	

	Call the getValueLength() method	getValueLength()	
18	Successful call, buffer length = 7Eh		GET INPUT Proactive command
	qualifier = 0		
	dcs = 4		Text String TLV =
	buffer = "UUU"		8D 7F 04 55 55
	offset = 0		Min Length = 00h
	<pre>length = 7Eh minRespLength = 00h</pre>		Max Length = 10h
	maxRespLength = 10h		
19	Successful call, buffer length = 7Fh		GET INPUT Proactive
13	Successful call, buller leligtif = 71 ff		command
	qualifier = 0		Command
	dcs = 4		Text String TLV = 8D 81
	buffer = "UUU"		80 04 55 55
	offset = 0		Min Length = 00h
	length = 7Fh		Max Length = 10h
	minRespLength = 00h		
	maxRespLength = 10h		
20	Successful call, buffer length = 236		GET INPUT Proactive
			command
	Qualifier = 0 dcs = 4		
	dcs = 4 buffer = "UUU"		Text String TLV =
	offset = 0		8D 81 ED 04 55 55
	length = 236		
	minRespLength = 00h		
	maxRespLength = 10h		
21	Call the initGetInput() method with a too long	HANDLER OVERFLOW	
	buffer	ToolkitException is thrown	
	qualifier = 0		
	dcs = 4		
	buffer = "XXXX"		
	offset = 0		
	length = 237		
	minRespLength = 00h		
00	maxRespLength = 10h		No ana active accessor !
22	Call the initGetInput() without sending the		No proactive command
	command		shall be sent expected
			status is '9000'

6.2.7.5.4 Test Coverage

CRR number	Test case number	
N1	8, 9, 10, 11, 12, 13, 14,	
	15,16,18, 19, 20	
N2	15	
N3	17	
N4	22	
N5	7	
<u>N6</u>	<u>16</u>	
P1	1	
P2	2, 3, 4, 5, 6	
C1	21	

6.2.7.6 Method send

Test Area Reference: API_2_PAH_SEND-

6.2.7.6.1 Conformance requirement

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

```
public byte send()
```

6.2.7.6.1.1 Normal execution

CRRN1: The send() method send the current proactive command to the mobile.

CRRN2: The returned byte is equal to general result of the command (first byte of Result TLV in Terminal Response).

CRRN3: The handler remains unchanged after a call to send() method until the use of initXX() or appendTLV().

CRRN4: There is no invocation of select() or deselect() method.

CRRN5: A pending toolkit applet transaction at the method invocation is aborted.

6.2.7.6.1.2 Parameter errors

No requirements.

6.2.7.6.1.3 Context errors

CRRC1: A ToolkitException.UNAVAILABLE_ELEMENT shall be thrown is the Result Simple TLV is missing in Terminal Response.

CRRC2: A ToolkitException.OUT_OF_TLV_BOUNDARIES shall be thrown if the general result byte is missing in the Result Simple TLV in Terminal Response.

<u>CRRC3</u>: A ToolkitException COMMAND NOT ALLOWED shall be thrown if the proactive command to be sent is not allowed by the SIM Toolkit Framework.

<u>CRRC4</u>: A ToolkitException COMMAND_NOT_ALLOWED shall be thrown if one parameter of the proactive command to be sent is not allowed by the SIM Toolkit Framework.

6.2.7.6.2 Test Suite files

Test Script: ____API_2_PAH_SEND_1.scr

Test Applet: API_2_PAH_SEND_1.java

Load Script: API_2_PAH_SEND_1.ldr

Cleanup Script: API_2_PAH_SEND_1.elr

Parameter File: API_2_PAH_SEND_1.java

Load Script: API_2_PAH_SEND_1.ldr

Cleanup Script: API_2_PAH_SEND_1.ldr

Cleanup Script: API_2_PAH_SEND_1.clr

Parameter File: API_2_PAH_SEND_1.par

6.2.7.6.3 Test procedure

ld	Description	API Expectation	APDU Expectation
1	Build and send a DISPLAY TEXT command	7 1 = Apootation	DISPLAY TEXT Proactive
'	qualifier = 00h		command
	dcs = 04h		Johnmana
	<pre>buffer = 'Text'</pre>		
2	Terminal Response with General Result = 00	Result of send() is 00h	
	Result TLV = 03 01 00 (command performed		
3	successfully) Build and send a DISPLAY TEXT command		DISPLAY TEXT Proactive
3	qualifier = 00h		command
	dcs = 04h		Command
	buffer = 'Text'		
4	Terminal Response with General Result = 01,	Result of send() is 01h	
	without Additional information on result		
	Result TLV = 03 01 01 (command performed		
	with partial comprehension)		
5	Build and send a DISPLAY TEXT command		DISPLAY TEXT Proactive
	qualifier = 00h		command
	<pre>dcs = 04h buffer = 'Text'</pre>		
6	Terminal Response with General Result = 01,	Result of send() is 01h	
0	with Additional information on result	Result of Seria() is offi	
	with Additional Information on result		
	Result TLV = 03 02 01 55 (command		
	performed with partial comprehension)		
7	Build and send a DISPLAY TEXT command		DISPLAY TEXT Proactive
	qualifier = 00h		command
	dcs = 04h		
	buffer = 'Text'	D = 1(- (10) : - 001	
8	Terminal Response with General Result = 02	Result of send() is 02h	
	Result TLV = 03 04 02 65 43 21 (Missing		
	information)		
9	Build and send a 7Fh byte command		DISPLAY TEXT Proactive
	(DISPLAY TEXT)		command
	qualifier = 00h		
	dcs = 04h		BER-TLV = D0 7F
	buffer = "UUUUU"		Text String TLV = 8D 74
10	length = 73h Build and send a 80h byte command		DISPLAY TEXT Proactive
10	(DISPLAY TEXT)		command
	qualifier = 00h		Command
	dcs = 04h		BER-TLV = D0 81 80
	buffer = "UUUUU"		Text String TLV = 8D 75
	length = 74h		04 55 55 55
11	Build and send a maximum length command		DISPLAY TEXT Proactive
	(length of the handler should be 253)		command
	DIGDIAY MEYE.		DED ELT - 50 01
	DISPLAY TEXT: Qualifier = 0		BER-TLV = D0 81 FD Text String TLV = 8D 81
	dcs = 4		F1 04 55 55
	buffer = "UUU"		
	offset = 0		
	length = 240		
12	Verify that the Proactive Handler is not		
	modified after a send()		
	Build a DISPLAY TEXT command		
	Comp. Dropothyol londles to account but a com-		
	Copy ProactiveHandler to source byte array		
	Send command		
	Sena commana		
	Conv Proactive Handler to destination buts		
	Copy ProactiveHandler to destination byte		
	array		
	Compare source and destination	Source and destination are	
	Compare source and destination	Source and destination are identical	
13	Build and send a DISPLAY TEXT command	iucitiidai	DISPLAY TEXT Proactive
13			command
L	Verify there is no invocation of -select() or	<u> </u>	COMMINATIO

	deselect() method.		
14	Build and send a DISPLAY TEXT command		DISPLAY TEXT Proactive command
	Terminal Response with 2 Result TLV	Result of send() is 02h	
	1st Result TLV = 03 02 02 12		
	2nd Result TLV = 03 03 03 34 56		
15	Build and send a DISPLAY TEXT command		DISPLAY TEXT Proactive command
	Terminal Response without Result Simple TLV	ToolkitException.UNAVAILABLE _ELEMENT is thrown by send()	
16	Build and send a DISPLAY TEXT command		DISPLAY TEXT Proactive command
	Terminal Response without general result byte in the Simple TLV	ToolkitException.OUT_OF_TLV_ BOUNDARIES is thrown by send()	
	Result TLV = 03 00		

6.2.7.6.4 Test Coverage

CRR number	Test case number
N1	1, 3, 5, 7, 9, 10, 11, 12,
	13, 14
N2	2, 4, 6, 8, 14
N3	12
N4	13
N5	To be checked in
	Framework tests and
	insert here cross
	reference
C1	15
C2	16
<u>C3</u>	checked in the
	Framework test:
	FWK_PCS_PCCO (test
	<u>case 1)</u>
<u>C4</u>	checked in the
	Framework test:
	FWK_PCS_PCCO (test
	cases 2 to 3)

6.2.7.12 Method copyValue

Test Area Reference API_2_PAH_CPYVS_BSS-

6.2.7.12.1 Conformance requirement

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

6.2.7.12.1.1 Normal execution

CRRN1: copies a part of the last TLV element which has been found, into a destination. buffer.

CRRN2: returns -dstOffset + dstLength.

6.2.7.12.1.2 Parameter errors

CRRP1: if dstBuffer is null NullPointerException is thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative ArrayIndexOutOfBoundsException is thrown.

—CRRP3: if valueOffset, dstLength or both are out of the current TLV valueOffset is negative or valueOffset + dstLength > current TLV length, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

6.2.7.12.1.3 Context errors

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

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.2.7.12.2 Test Suite files

Test Script: ____API_2_PAH_CPYVS_BSS_1.scr

Test Applet: API_2_PAH_CPYVS_BSS_1.java

Load Script: API_2_PAH_CPYVS_BSS_1.ldr

Cleanup Script: API_2_PAH_CPYVS_BSS_1.clr

Parameter File: API_2_PAH_CPYVS_BSS_1.java

Load Script: API_2_PAH_CPYVS_BSS_1.ldr

Cleanup Script: API_2_PAH_CPYVS_BSS_1.clr

Parameter File: API_2_PAH_CPYVS_BSS_1.par

6.2.7.12.3 Test procedure

ld	Description	API Expectation	APDU Expectation
	Initialize the handler		
	Select a TLV		
<u>1</u>	Initialise the handler		
	Select a TLV		
	copyValue() with a null dstBuffer	NullPointerException is thrown	
2	initDisplayText() with length = 15		
	Select Text String TLV		
	dstOffset > dstBuffer.length	ArrayIndexOutOfBoundsExceptio	
	dstBuffer.length = 5	n is thrown	
	dstOffset = 6		
	dstLength = 0		
3	dstOffset < 0	ArrayIndexOutOfBoundsExceptio	
	dstBuffer.length = 5	n is thrown	
	dstOffset = -1		
	dstLength = 1		
4	dstLength >dstBuffer.length	ArrayIndexOutOfBoundsExceptio	
	dstBuffer.length = 5	n is thrown	
	dstOffset = 0		
- -	dstLength = 6	1 1 1 0 100	
5	dstOffset + dstLength >dstBuffer.length	ArrayIndexOutOfBoundsExceptio	
	dstBuffer.length = 5	n is thrown	
	<pre>dstOffset = 3 dstLength = 3</pre>		
6	dstLength < 0	ArrayIndexOutOfBoundsExceptio	

r			T	T
		dstBuffer.length = 5	n is thrown	
		dstOffset = 0		
		dstLength = -1		
-	7	initDiamler/Text() with length E		
	7	initDisplayText() with length = 5		
		Select Text String TLV		
		valueOffset > Text String Length	ToolkitException.OUT_OF_TLV_	
		valueOffset = 7	BOUNDARIES is thrown	
		dstBuffer.length = 15	BOOMB/ WILE IS WIIOWII	
		dstOffset = 0		
		dstLength = 0		
	8	[Select Text String TLV]	ToolkitException.OUT_OF_TLV_	
		valueOffset < 0	BOUNDARIES is thrown	
		valueOffset = -1		
		dstBuffer.length = 15		
		dstOffset = 0		
		dstLength = 1		
	9	[Select Text String TLV]	ToolkitException.OUT_OF_TLV_	
	9		TOOIRILEXCEPTION.OUT_OF_TEV_	
		dstLength > Text String length	BOUNDARIES is thrown	
		valueOffset = 0		
		dstBuffer.length = 15		
		dstOffset = 0		
		dstLength = 7		
	10	[Select Text String TLV]	ToolkitException.OUT_OF_TLV_	
	. •	valueOffset + dstLength > Text String	BOUNDARIES is thrown	
		length	DOUNDAINES IS UNOWN	
		valueOffset = 2		
		dstBuffer.length = 15		
		3.		
		dstOffset = 0		
		dstLength = 5		
			<u>.</u>	<u>.</u>
11		Initialize the handler		
· [11	Initialise the handler		
	<u>11</u>	Initialise the handler	ToolkitException LINAVAILABLE	
'	<u>11</u>	Initialise the handler copyValue()	ToolkitException.UNAVAILABLE	
, 		copyValue()	ToolkitException.UNAVAILABLE _ELEMENT is thrown	
' 	12			
		copyValue() initDisplayText() dcs = 4		
		copyValue() initDisplayText()		
		copyValue() initDisplayText() dcs = 4		
		copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV	_ELEMENT is thrown	
		copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call		
,		copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0	_ELEMENT is thrown	
'		copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17	_ELEMENT is thrown	
		copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0	_ELEMENT is thrown	
	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17	_ELEMENT is thrown Result of copyValue() is 17	
		copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer	_ELEMENT is thrown	
	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17	_ELEMENT is thrown Result of copyValue() is 17	
' .	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer	_ELEMENT is thrown Result of copyValue() is 17	
14	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer	_ELEMENT is thrown Result of copyValue() is 17	
	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F	_ELEMENT is thrown Result of copyValue() is 17	
	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer Suffer = 55 55 55	_ELEMENT is thrown Result of copyValue() is 17	
	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 initialise dstBuffer	_ELEMENT is thrown Result of copyValue() is 17	
	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 initialise dstBuffer dstBuffer = 55 55 55	_ELEMENT is thrown Result of copyValue() is 17 Result is 00h	
	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 initialise dstBuffer dstBuffer = 55 55 55 Successful call	_ELEMENT is thrown Result of copyValue() is 17	
	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 initialise dstBuffer dstBuffer = 55 55 55 Successful call valueOffset = 2	_ELEMENT is thrown Result of copyValue() is 17 Result is 00h	
	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 initialise dstBuffer dstBuffer = 55 55 55 Successful call valueOffset = 2 dstBuffer.length = 20	_ELEMENT is thrown Result of copyValue() is 17 Result is 00h	
	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 initialise dstBuffer dstBuffer = 55 55 55 Successful call valueOffset = 2	_ELEMENT is thrown Result of copyValue() is 17 Result is 00h	
	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 initialise dstBuffer dstBuffer = 55 55 55 Successful call valueOffset = 2 dstBuffer.length = 20	_ELEMENT is thrown Result of copyValue() is 17 Result is 00h	
	12	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 initialise dstBuffer dstBuffer = 55 55 55 Successful call valueOffset = 2 dstBuffer.length = 20 dstOffset = 3 dstLength = 12	Result of copyValue() is 17 Result is 00h Result of copyValue() is 15	
	13 13 4stI 14	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 Successful call valueOffset = 2 dstBuffer.length = 20 dstOffset = 3 dstLength = 12 Compare buffer	_ELEMENT is thrown Result of copyValue() is 17 Result is 00h	
	13 13 4stI 14	<pre>copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 initialise dstBuffer dstBuffer = 55 55 55 Successful call valueOffset = 2 dstBuffer.length = 20 dstOffset = 3 dstLength = 12 Compare buffer buffer =</pre>	Result of copyValue() is 17 Result is 00h Result of copyValue() is 15	
	13 13 4stI 14	<pre>copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 initialise dstBuffer dstBuffer = 55 55 55 Successful call valueOffset = 2 dstBuffer.length = 20 dstOffset = 3 dstLength = 12 Compare buffer buffer = 55 55 55 01 02</pre>	Result of copyValue() is 17 Result is 00h Result of copyValue() is 15	
	13 13 4stI 14	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 initialise dstBuffer dstBuffer = 55 55 55 Successful call valueOffset = 2 dstBuffer.length = 20 dstOffset = 3 dstLength = 12 Compare buffer buffer = 55 55 55 01 02 03 04 05 06 07	Result of copyValue() is 17 Result is 00h Result of copyValue() is 15	
	13 13 4stI 14	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 Successful call valueOffset = 2 dstBuffer.length = 20 dstOffset = 3 dstLength = 12 Compare buffer buffer = 55 55 01 02 03 04 05 06 07 08 09 0A 0B 0C	Result of copyValue() is 17 Result is 00h Result of copyValue() is 15	
	13 13 4stI 14	copyValue() initDisplayText() dcs = 4 buffer = 00 01 0F Select Text String TLV Successful call valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17 Compare buffer buffer = 04 00 01 0F initialize dstBuffer suffer = 55 55 55 initialise dstBuffer dstBuffer = 55 55 55 Successful call valueOffset = 2 dstBuffer.length = 20 dstOffset = 3 dstLength = 12 Compare buffer buffer = 55 55 55 01 02 03 04 05 06 07	Result of copyValue() is 17 Result is 00h Result of copyValue() is 15	

6.2.7.12.4 Test Coverage

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

6.2.7.13 Method compareValue

Test Area Reference API_2_PAH_CPRVS_BSS-

6.2.7.13.1 Conformance requirement

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

6.2.7.13.1.1 Normal execution

Compares the last found TLV element with a buffer:

CRRN1: returns 0 if identical.

CRRN2: returns -1 if the first miscomparing byte in simple TLV List is less than that in compareBuffer.

CRRN3: returns 1 if the first miscomparing byte in simple TLV List is greater than that in compareBuffer.

6.2.7.13.1.2 Parameter errors

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset or compareLength or both would cause access outside array bounds, or if compareLength is negative ArrayIndexOutOfBoundsException shall be thrown.

—CRRP3: if valueOffset, dstLength or both are out of the current TLV valueOffset is negative or valueOffset + dstLength > current TLV length, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

6.2.7.13.1.3 Context errors

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

CRRC2: in case of unavailable TLV element an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException UNAVAILABLE_ELEMENT.

6.2.7.13.2 Test Suite files

Test Script: _____API_2_PAH_CPRVS_BSS_1.scr

Test Applet: API_2_PAH_CPRVS_BSS_1.java

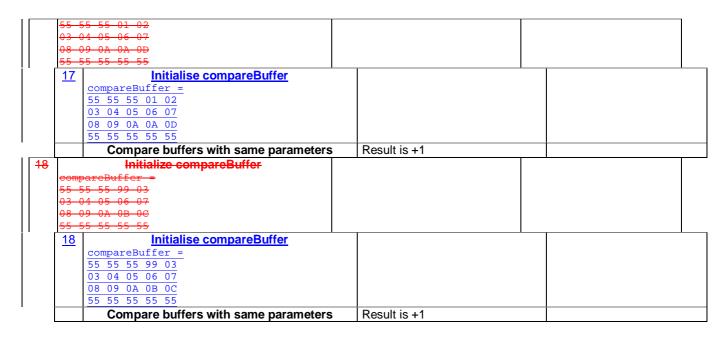
Load Script: API_2_PAH_CPRVS_BSS_1.ldr

Cleanup Script:API_2_PAH_CPRVS_BSS_1.clrParameter File:API_2_PAH_CPRVS_BSS_1.javaLoad Script:API_2_PAH_CPRVS_BSS_1.ldrCleanup Script:API_2_PAH_CPRVS_BSS_1.clrParameter File:API_2_PAH_CPRVS_BSS_1.par

6.2.7.13.3 Test procedure

Id	I Description	API Expectation	APDU Expectation
<u> </u>	Initialize the handler Select a TLV		
1	Select a TLV		
	compareValue() with a null compareBuffer	NullPointerException is thrown	
2	Select Text String TLV		
	<pre>compareOffset > compareBuffer.length compareBuffer.length = 5 compareOffset = 6 compareLength = 0</pre>	ArrayIndexOutOfBoundsException is thrown	
3	<pre>compareOffset < 0 compareBuffer.length = 5 compareOffset = -1 compareLength = 1</pre>	ArrayIndexOutOfBoundsExceptio n is thrown	
4	<pre>compareLength >compareBuffer.length compareBuffer.length = 5 compareOffset = 0 compareLength = 6</pre>	ArrayIndexOutOfBoundsExceptio n is thrown	
5	<pre>>compareBuffer.length compareBuffer.length = 5 compareOffset = 3 compareLength = 3</pre>	ArrayIndexOutOfBoundsException is thrown	
6	compareBuffer.length = 5 compareOffset = 0 compareLength = -1	ArrayIndexOutOfBoundsException is thrown	
7	Select Text String TLV		
	<pre>valueOffset > Text String Length valueOffset = 7 compareBuffer.length = 15 compareOffset = 0 compareLength = 0</pre>	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
8	<pre>[Select Text String TLV] valueOffset < 0 valueOffset = -1 compareBuffer.length = 15 compareOffset = 0 compareLength = 1</pre>	ToolkitException.OUT_OF_TLV_ BOUNDARIES is thrown	
9	[Select Text String TLV] compareLength > Text String length valueOffset = 0 compareBuffer.length = 15 compareOffset = 0 compareLength = 7	ToolkitException.OUT_OF_TLV_ BOUNDARIES is thrown	
10	<pre>()</pre>	ToolkitException.OUT_OF_TLV_BOUNDARIES is thrown	
4	Initialize the handler		
11	Initialise the handler		

		compareValue()		kitException.UNAVAILABLE_		
	<u> </u>	compareValue()		MENT is thrown ToolkitException.UNAVAILABI	IF I	
				ELEMENT is thrown	<u></u>	
	12	initDisplayText()				
		dcs = 4 buffer = 00 01 0F				
1	<u> </u>	Select Text String TLV				
	com	Initialize compareBuffer				
	04 (00 01 0F				
		<pre>Initialise compareBuffer compareBuffer =</pre>				
		04 00 01 0F				
		Compare buffers valueOffset = 0		Result is 00h		
		compareOffset = 0				
		compareLength = 17				
13	Т	Initialize compareBuffer	<u> </u>			
	comp	vareBuffer =				
		05 06 07 08				
	05 (0E :	0 A 0B 0C 0D				
1	13	Initialise compareBuffer				
		compareBuffer =				
		04 00 01 02 03 04 05 06 07 08				
		05 0A 0B 0C 0D 0E 10				
ļ		Compare buffers with same parameters	S	Result is -1		
1 .	_					
14	COM	Initialize compareBuffer				
		00 01 0F				
	<u>14</u>	<pre>Initialise compareBuffer compareBuffer =</pre>				
		03 00 01 0F				
		Compare buffers with same parameters	S	Result is +1		
45	Т	Initialize compareBuffer				
	COM	pareBuffer =				
		94 05 06 07				
	00	99 OA OB OC 55 55 55				
	<u>15</u>	Initialise compareBuffer		l		
		<pre>compareBuffer = 55 55 55 01 02</pre>				
		03 04 05 06 07				
		08 09 0A 0B 0C 55 55 55 55 55				
Ų		Compare buffers		Result is 00h		
		<pre>valueOffset = 2 compareOffset = 3</pre>				
		compareLength = 12				
16		Initialize compareBuffer	<u> </u> 	T		1
	com	vareBuffer =				
	55 !	55 55 02 01 04 05 06 07				
	08 (99 0A 0B 0C				
1	55 ! 16	55 55 55 55 Initialise compareBuffer	<u> </u>			
	10	<pre>compareBuffer =</pre>				
		55 55 55 02 01 03 04 05 06 07				
		08 09 0A 0B 0C				
I		55 55 55 55 55 Compare buffers with same parameters		Result is -1		
		-				
17		Initialize compareBuffer				
11	com	pareBuffer =	1			1



6.2.7.13.4 Test Coverage

CRR number	Test case number
N1	12, 15
N2	13, 16
N3	14, 17, 18
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for
	Proactive Handler
C2	11

6.2.7.15 Method findAndCopyValue(byte tag, byte occurence, short valueOffset, byte[] dstBuffer, short dstOffset, short dstLength)

Test Area Reference API_2_PAH_FACYBBS_BSS-

6.2.7.15.1 Conformance requirement

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

6.2.7.15.1.1 Normal execution

CRRN1: looks for the indicated occurrence of a TLV element from the beginning of a TLV list and copy its value into a destination buffer.

CRRN2: -if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.

CRRN3: if the method is successful then the corresponding TLV becomes current and dstOffset + dstLength is returned.

CRRN4: The search method is comprehension required flag independent.

6.2.7.15.1.2 Parameter errors

CRRP1: if dstBuffer is null NullPointerException shall be thrown.

CRRP2: if dstOffset or dstLength or both would cause access outside array bounds, or if dstLength is negative ArrayIndexOutOfBoundsException shall be thrown.

—CRRP3: if valueOffset, dstLength or both are out of the current TLV valueOffset is negative or valueOffset + dstLength > current TLV length, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

6.2.7.15.1.3 Context errors

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

6.2.7.15.2 Test Suite files

Test Script: ____API_2_PAH_FACYBBS_BSS_1.scr

Test Applet: API_2_PAH_FACYBBS_BSS_1.java

Load Script: API_2_PAH_FACYBBS_BSS_1.ldr

Cleanup Script: API_2_PAH_FACYBBS_BSS_1.elr

Parameter File: API_2_PAH_FACYBBS_BSS_1.java

Load Script: API_2_PAH_FACYBBS_BSS_1.ldr

Cleanup Script: API_2_PAH_FACYBBS_BSS_1.clr

Parameter File: API_2_PAH_FACYBBS_BSS_1.par

6.2.7.15.3 Test procedure

	ld	Description	API Expectation	APDU Expectation
4		Initialize the handler		·
ĺ	1	Initialise the handler		•
•		findAndCopyValue() with a null dstBuffer	NullPointerException is thrown	
	2	initDisplayText() with length = 15		
		<pre>dstOffset > dstBuffer.length tag = 0Dh, occurrence = 1 valueOffset = 0 dstBuffer.length = 5 dstOffset = 6 dstLength = 0</pre>	ArrayIndexOutOfBoundsExceptio n is thrown	
	3	<pre>dstOffset < 0 dstBuffer.length = 5 dstOffset = -1 dstLength = 1</pre>	ArrayIndexOutOfBoundsExceptio n is thrown	
	4	<pre>dstLength >dstBuffer.length dstBuffer.length = 5 dstOffset = 0 dstLength = 6</pre>	ArrayIndexOutOfBoundsExceptio n is thrown	
	5	<pre>dstOffset + dstLength >dstBuffer.length dstBuffer.length = 5 dstOffset = 3 dstLength = 3</pre>	ArrayIndexOutOfBoundsExceptio n is thrown	
	6	<pre>dstLength < 0 dstBuffer.length = 5</pre>	ArrayIndexOutOfBoundsExceptio n is thrown	

	dstOffset = 0		
	dstLength = -1		
7	initDianloyToyt() with longth - F		
7	initDisplayText() with length = 5		
	valueOffset > Text String Length	ToolkitException.OUT_OF_TLV_	
	tag = 0Dh, occurrence = 1	BOUNDARIES is thrown	
	valueOffset = 7		
	dstBuffer.length = 15		
	dstOffset = 0		
	dstLength = 0		
8	valueOffset < 0	ToolkitException.OUT_OF_TLV_	
	valueOffset = -1	BOUNDARIES is thrown	
	dstBuffer.length = 15		
	dstOffset = 0		
	dstLength = 1		
9	dstLength > Text String length	ToolkitException.OUT_OF_TLV_	
	valueOffset = 0	BOUNDARIES is thrown	
	dstBuffer.length = 15		
	dstOffset = 0		
	dstLength = 7		
10	valueOffset + dstLength > Text String length	ToolkitException.OUT_OF_TLV_	
	valueOffset = 2	BOUNDARIES is thrown	
	dstBuffer.length = 15		
	dstOffset = 0		
	dstLength = 5		
11	InitDisplayText()		
· ·	Select a TLV (tag 02h)		
-	findAndCopyValue()	ToolkitException.UNAVAILABLE	
	tag = 0Dh	_ELEMENT is thrown	
	occurrence = 2	ToolkitEveention LINIAVALLADI E	
	Call the getValueLength() method	ToolkitException.UNAVAILABLE	
		_ELEMENT is thrown.	
12	initDisplayText()		
	dcs = 4		
	buffer = 00 01 0F		
	Successful call	Result of findAndCopyValue() is	
	tag = 0Dh, occurrence = 1	17	
	<pre>valueOffset = 0</pre>		
	dstBuffer.length = 17		
	dstOffset = 0		
	dstLength = 17		
13	Compare buffer	Result is 00h	
	buffer = 04 00 01 0F		
1	initialize dstBuffer		
dati	Buffer = 55 55 55		
14	initialise dstBuffer		•
1	dstBuffer = 55 55 55		
	Successful call	Result of findAndcopyValue() is	
	tag = 0Dh, occurrence = 1	15	
	valueOffset = 2	'	
	dstBuffer.length = 20		
	dstOffset = 3		
	dstLength = 12		
15	Compare buffer	Result is 00h	
. •	buffer =		
	55 55 55 01 02		
	03 04 05 06 07		
	08 09 0A 0B 0C		
	55 55 55 55 55		
16	Append a Text String TLV		
'	tag = 0D		
	buffer = 00 11 22 33 44 55 (no specific		
	DCS byte)		
	Successful call	Result of findAndCopyValue() is	
	tag = 0Dh, occurrence = 1	17	
	valueOffset = 0	''	
	dstBuffer.length = 17		
	dstOffset = 0		
	dstLength = 17		
17	Compare buffer	Result is 00h	
''	buffer = 04 00 01 0F	1.Court is out	
	Daller - or oo or or		
	I	[

19	Successful call tag = 0Dh, occurrence = 2 valueOffset = 0 dstBuffer.length = 6 dstOffset = 0 dstLength = 6 Compare buffer buffer = 00 11 22 33 44 55	Result of findAndCopyValue() is 6 Result is 00h
20	<pre>initDisplayText() dcs = 4 buffer = 00 01 0F</pre>	
	Successful call (with tag 8Dh) tag = 8Dh occurrence = 1 valueOffset = 0 dstBuffer.length = 17 dstOffset = 0 dstLength = 17	Result of findAndcopyValue() is 17
21	Compare buffer buffer = 04 00 01 0F	Result is 00h
22	Append tag 0Fh buffer = 00 01 0F	
	Successful call (with tag 8Fh) tag = 8Fh occurrence = 1 valueOffset = 0 dstBuffer.length = 16 dstOffset = 0 dstLength = 16	Result of findAndcopyValue() is 16
23	Compare buffer buffer = 00 01 0F	Result is 00h

6.2.7.15.4 Test Coverage

CRR number	Test case number
N1	13, 15, 17, 19
N2	11
N3	12, 14, 16, 18
N4	20, 21, 22, 23
P1	1
P2	2, 3, 4, 5, 6
P3	7, 8, 9, 10
C1	Does not apply for ProactiveHandler

6.2.7.17 Method findAndCompareValue(byte tag, byte occurence, short valueOffset, byte[] compareBuffer, short compareOffset, short compareLength)

Test Area Reference API_2_PAH_FACRBBS_BSS-

6.2.7.17.1 Conformance requirement

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

6.2.7.17.1.1 Normal execution

Looks for the indicated occurrence of a TLV element from the beginning of a TLV list and compare its value with a buffer:

CRRN1: if no TLV element is found, the UNAVAILABLE_ELEMENT exception is thrown and the current TLV is no longer defined.-

CRRN2: if the method is successful then the corresponding TLV becomes current.

CRRN3: if identical -0 is returned.

CRRN4: if the first miscomparing byte in simple TLV is less than that in compareBuffer -1 is returned.

CRRN5: if the first miscomparing byte in simple TLV is greater than that in compareBuffer 1 is returned.

CRRN6: The search method is comprehension required flag independent.

6.2.7.17.1.2 Parameter errors

CRRP1: if compareBuffer is null NullPointerException shall be thrown.

CRRP2: if compareOffset or compareLength or both would cause access outside array bounds, or if compareLength is negative ArrayIndexOutOfBoundsException shall be thrown.

—CRRP3: if valueOffset, compareLength or both are out of the current TLV valueOffset is negative or valueOffset + dstLength > current TLV length, an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException OUT_OF_TLV_BOUNDARIES.

—CRRP4: if an input parameter is not valid (e.g. occurrence = 0) an instance of ToolkitException shall be thrown. The reason code shall be ToolkitException BAD_INPUT_PARAMETER.

6.2.7.17.1.3 Context errors

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

6.2.7.17.2 Test Suite files

Test Script:	_API_2_PAH_FACRBBS_BSS_1.scr
Test Applet:	API_2_PAH_FACRBBS_BSS_1.java
Load Script:	API_2_PAH_FACRBBS_BSS_1.ldr
Cleanup Script:	API_2_PAH_FACRBBS_BSS_1.elr
Parameter File:	API_2_PAH_FACRBBS_BSS_1.java
Load Script:	API 2 PAH FACRBBS BSS 1.ldr
Cleanup Script:	API_2_PAH_FACRBBS_BSS_1.clr
Parameter File:	API_2_PAH_FACRBBS_BSS_1.par

6.2.7.17.3 Test procedure

ld	Description	API Expectation	APDU Expectation
	Initialize the handler		
1	Initialise the handler		
	findAndCompareValue() with a null	NullPointerException is thrown	
	compareBuffer	·	
	·		
2	initDisplayText() with length = 15		
	compareOffset > compareBuffer.length	ArrayIndexOutOfBoundsException)
	tag = 0Dh, occurrence = 1	n is thrown	
	<pre>valueOffset = 0</pre>		
	compareBuffer.length = 5		
	compareOffset = 6		
3	<pre>compareLength = 0 compareOffset < 0</pre>	ArrayIndayOutOfPaundaEvaantia	.
3	compareBuffer.length = 5	ArrayIndexOutOfBoundsException is thrown	'
	compareOffset = -1	II IS UIIOWII	
	compareLength = 1		
4	compareLength >compareBuffer.length	ArrayIndexOutOfBoundsException)
	<pre>compareBuffer.length = 5</pre>	n is thrown	
	compareOffset = 0		
_	compareLength = 6	1	
5	compareOffset + compareLength	ArrayIndexOutOfBoundsException)
	<pre>>compareBuffer.length compareBuffer.length = 5</pre>	n is thrown	
	compareOffset = 3		
	compareLength = 3		
6	compareLength < 0	ArrayIndexOutOfBoundsException	
	compareBuffer.length = 5	n is thrown	
	compareOffset = 0		
	compareLength = -1		
	initDianles Tout() with leastly 5		
7	initDisplayText() with length = 5	T. History Co. Cliff Of Thy	
	<pre>valueOffset > Text String Length tag = 0Dh, occurrence = 1</pre>	ToolkitException.OUT_OF_TLV_	-
	valueOffset = 7	BOUNDARIES is thrown	
	compareBuffer.length = 15		
	compareOffset = 0		
	compareLength = 0		
8	valueOffset < 0	ToolkitException.OUT_OF_TLV_	-
	valueOffset = -1	BOUNDARIES is thrown	
	<pre>compareBuffer.length = 15 compareOffset = 0</pre>		
	compareLength = 1		
9	compareLength > Text String length	ToolkitException.OUT_OF_TLV_	
	<pre>valueOffset = 0</pre>	BOUNDARIES is thrown	
	<pre>compareBuffer.length = 15</pre>		
	compareOffset = 0		
10	compareLength = 7 valueOffset + compareLength > Text String	ToolkitException.OUT_OF_TLV_	
īŪ	length	BOUNDARIES is thrown	•
	valueOffset = 2	20011D/ II (IEO IO II II OWII	
	compareBuffer.length = 15		
	compareOffset = 0		
	compareLength = 5		
11	Invalid parameter	ToolkitEveenties BAD INDUT D	
11	Invalid parameter	ToolkitException.BAD_INPUT_P	
	Occurrence = 0	ARAMETER is thrown	
12	InitDisplayText()	<u> </u>	1
	Select a TLV (tag 02h)		
	findAndCompareValue()	ToolkitException.UNAVAILABLE	
	tag = 0Dh	_ELEMENT is thrown	
	occurrence = 2	_	
	Call the getValueLength() method	ToolkitException.UNAVAILABLE	
		_ELEMENT is thrown.	
13	initDisplayText()		
	dcs = 4		
	buffer = 00 01 0F Initialize compareBuffer	<u> </u>	
go»	nntianze comparebuner		
$-\sigma m$	PALCEALICE -		

		Initialise compareBuffer compareBuffer =	
		6 of AndCompareValue()	Result is 00h
		<pre>findAndCompareValue() tag = 0Dh, occurrence = 1</pre>	Result is our
		valueOffset = 0	
		<pre>compareOffset = 0 compareLength = 17</pre>	
	14	Verify current TLV	Result is 17
	14	getValueLength()	Result is 17
15		Initializa compare Puffer	
10	comp	Initialize compareBuffer	
	04 (15		
	10	<pre>compareBuffer =</pre>	
l		Compare buffers with same parameters	Result is -1
			Treatment 1
16	comr	Initialize compareBuffer	
ļl ,		00 01 0F	
	<u>16</u>	<pre>Initialise compareBuffer compareBuffer =</pre>	
		03 00 01 OF	D. His 4
		Compare buffers with same parameters	Result is +1
17		Initialize compareBuffer	
	COME	pareButter =	
	03 (04 05 06 07 09 08 0B 0C	
		55 55 55 55 55 55	
	<u>17</u>	<pre>Initialise compareBuffer compareBuffer =</pre>	
		55 55 55 01 02	
		03 04 05 06 07 08 09 0A 0B 0C	
		55 55 55 55 55 Compare buffers	Result is 00h
		valueOffset = 2	Result is 0011
		<pre>compareOffset = 3 compareLength = 12</pre>	
18	COME	Initialize compareBuffer	
	55 5	55 55 02 01 04 05 06 07	
	08 (09 0A 0B 0C	
	55 5 18	55 55 55 55 Initialise compareBuffer	
	10	compareBuffer =	
		55 55 55 02 01 03 04 05 06 07	
		08 09 0A 0B 0C 55 55 55 55 55	
ı		Compare buffers with same parameters	Result is -1
19		Initialize compareBuffer	
	COME	• • • • • • • • • • • • • • • • • • • •	
	03 (04 05 06 07	
		99 0A 0A 0D 55 55 55 55	
	<u>19</u>	Initialise compareBuffer	
		compareBuffer = 55 55 55 01 02	
		03 04 05 06 07 08 09 0A 0A 0D	
		55 55 55 55 55	
		Compare buffers with same parameters	Result is +1
	20	append a Text String TLV	

	tag = 0Dh		
, _	buffer = 00 11 22 33 44 55		
Gom	I nitialize compareBuffer		
	00 01 0F		
	Initialise compareBuffer		
	<pre>compareBuffer = 04 00 01 0F</pre>		
1	findAndCompareValue()	Result is 00h	
	tag = 0Dh, occurrence = 1		
	<pre>valueOffset = 0 compareOffset = 0</pre>		
	compareLength = 17		
	1.77.17		
21	Initialize compareBuffer		
00	- 11 22 33 44 55		
<u>21</u>	Initialise compareBuffer		
	<pre>compareBuffer = 00 11 22 33 44 55</pre>		
'	findAndCompareValue()	Result is 00h	
	<pre>tag = 0Dh, occurrence = 2 valueOffset = 0</pre>		
	<pre>compareOffset = 0</pre>		
	compareLength = 6		
	Initializa compana Dreffer		
22	Initialize compareBuffer		
	- 11 22 33 44 66		
<u>22</u>			
	<pre>compareBuffer = 00 11 22 33 44 66</pre>		
'	findAndCompareValue()	Result is –1	
	<pre>tag = 0Dh, occurrence = 2 valueOffset = 0</pre>		
	compareOffset = 0		
	compareLength = 6		
23	initDisplayText()		
23	dcs = 4		
	buffer = 00 01 0F		
Com	Initialize compareBuffer		
	Initialise compareBuffer		
	CompareBuffer = 04 00 01 0F	D. Hri eer	
	Successful call (with tag 8Dh) tag = 8Dh, occurrence = 1	Result is 00h	
	valueOffset = 0		
	compareBuffer.length = 17		
	<pre>compareOffset = 0 compareLength = 17</pre>		
24	Append tag 0Fh		
	buffer = 00 01 0F		
Com	Initialize compareBuffer		
	Initialise compareBuffer		
	compareBuffer = 00 01 0F	Popult is 00h	
	Successful call (with tag 8Fh) tag = 8Fh, occurrence = 1	Result is 00h	
	valueOffset = 0		
	<pre>compareBuffer.length = 16 compareOffset = 0</pre>		
	compareLength = 16		
25	Initialize compareBuffer		
25	pareBuffer =0099 02 0F Initialise compareBuffer		
	compareBuffer =0099 02 0F		
	findAndCompareValue()	Result is +1	
	<pre>tag = 0Dh, occurrence = 1 valueOffset = 0</pre>		
	compareOffset = 0		
	compareLength = 17		

6.2.7.17.4 Test Coverage

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

6.2.7.24 Method getCapacity

Test Area Reference: API_2_PAH_GCAP

6.2.7.24.1 Conformance Requirement

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

public byte getCapacity()

6.2.7.24.1.1 Normal execution

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

6.2.7.24.2 Test suite files

Test Script: API_2_PAH_GCAP_1.scr

Test Applet: API_2_PAH_GCAP_1.java

Load Script: API_2_PAH_GCAP_1.ldr

Cleanup Script: API_2_PAH_GCAP_1.clr

Parameter File: API_2_PAH_GCAP_1.par

6.2.7.24.3 Test Procedure

ld	<u>Description</u>	API Expectation	APDU Expectation
1	ProactiveHandler available		
	1- Send envelope SMS-PP Formatted	1- Applet is triggered	
	2- The applet calls getTheHandler()	2- No exception is thrown	
	3- The applet calls getCapacity() on the	3- No exception is thrown, the	
	ProactiveHandler	capacity shall not be null	
	4- The applet fills the handler with the maximum	4- No exception is thrown	
	capacity, using appendTLV() method		
	5- The applet calls clear() on the proactive handler	5- No exception is thrown	
	6- The applet fills the handler with the maximum		
	capacity plus one, using appendTLV() method	6- HANDLER OVERFLOW	
		exception is thrown	

6.2.7.24.4 Test Coverage

CRR number	Test case number
<u>N1</u>	<u>1</u>

6.2.7.25 Method initCloseChannel

Test Area Reference: API_2_PAH_ICCHB

6.2.5.21.1 Conformance requirement

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

public void initCloseChannel(byte bChannelIdentifier)

6.2.7.25.1.1 Normal execution

<u>CRRN1: The method shall build a Close Channel Proactive command, using Channel Identifier. Comprehension Required flags are set.</u>

CRRN2: A call to this method clears the handler then initialises it with Close Channel Proactive command.

CRRN3: After the method invocation, no TLV is selected.

CRRN4: The CLOSE CHANNEL Proactive command is not sent by the method.

6.2.7.25.2 Test suite files

Test Script:	API_2_PAH_ICCHB_1.scr
Test Applet:	API_2_PAH_ICCHB_1.java
Load Script:	API 2 PAH ICCHB 1.ldr
Cleanup Script:	API 2 PAH ICCHB 1.clr
Parameter File:	API_2_PAH_ICCHB_1.par

6.2.7.25.3 Test procedure

ld	<u>Description</u>	API Expectation	APDU Expectation
0	Applet1 is installed with maximum number		
	of channel = 01.		
1	Call initCloseChannel() method 1- Call ProactiveHandler.init() method to Open a Channel. Call the ProactiveHandler.send() method.	2- Applet1 is triggered.	1- OPEN CHANNEL proactive command is fetched.
	dari die rraddri maiarer bena() medioa.		TERMINAL RESPONSE of

	2- Send an EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS Envelope.	5- Applet1 is not triggered.	OPEN CHANNEL is sent to the SIM with Channel Id = 01.
	3- Call the ProactiveHandler.initCloseChannel() method with Channel Id = 01. 4- Call the ProactiveHandler.send()		4- CLOSE CHANNEL proactive command is fetched.
	method. 5- Send an EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS Envelope.		TERMINAL RESPONSE of CLOSE CHANNEL is sent to the SIM.
2	Call the initCloseChannel () method with any value then build and send a CLOSE CHANNEL command	5- Applet1 is not triggered.	1- OPEN CHANNEL proactive command is fetched.
	1- Call ProactiveHandler.init() to Open a Channel and ProactiveHandler.send() methods. 2- ProactiveHandler.initCloseChannel() with Channel Id = 2		TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM with Channel Id = 01.
	3- ProactiveHandler.initCloseChannel() with the Channel Id = 1. 4- call the send() method.		4- CLOSE CHANNEL proactive command is fetched.
	5- Send an EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS Envelope.		TERMINAL RESPONSE of CLOSE CHANNEL is sent to the SIM.
<u>3</u>	Select a TLV in the ProactiveHandler Call the initCloseChannel () method 1- Call ProactiveHandler.init() method to	3- UNAVAILABLE_ELEMENT ToolkitException is thrown by getValueLength() method.	1- OPEN CHANNEL proactive command is fetched.
	open a Channel and call the		TERMINIAL DECRONICE -4
	ProactiveHandler.send() method. Select 1st TLV of the Proactive Handler. 2- Call		TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM with Channel Id = 01.
	Select 1st TLV of the Proactive Handler.		OPEN CHANNEL is sent to the SIM with Channel Id =
	Select 1st TLV of the Proactive Handler. 2- Call ProactiveHandler.initCloseChannel() method with Channel Id = 01. 3- Call the ViewHandler.getValueLength()		OPEN CHANNEL is sent to the SIM with Channel Id = 01. 4- CLOSE CHANNEL proactive command is
4	Select 1st TLV of the Proactive Handler. 2- Call ProactiveHandler.initCloseChannel() method with Channel Id = 01. 3- Call the ViewHandler.getValueLength() method. 4- Call ProactiveHandler.send() method. Call the initCloseChannel() without sending the command	3- Applet1 is triggered.	OPEN CHANNEL is sent to the SIM with Channel Id = 01. 4- CLOSE CHANNEL proactive command is fetched. TERMINAL RESPONSE of CLOSE CHANNEL is sent
4	Select 1st TLV of the Proactive Handler. 2- Call ProactiveHandler.initCloseChannel() method with Channel Id = 01. 3- Call the ViewHandler.getValueLength() method. 4- Call ProactiveHandler.send() method. Call the initCloseChannel() without sending	3- Applet1 is triggered.	OPEN CHANNEL is sent to the SIM with Channel Id = 01. 4- CLOSE CHANNEL proactive command is fetched. TERMINAL RESPONSE of CLOSE CHANNEL is sent to the SIM. 1- OPEN CHANNEL proactive command is

6.2.5.21.1 Test Coverage

CRR number	Test case number
<u>N1</u>	<u>1</u>
<u>N2</u>	<u>2</u>
<u>N3</u>	<u>3</u>
N4	2, 4

6.2.8 Class ProactiveResponseHandler

6.2.8.20 Method getCapacity

Test Area Reference: API_2_PRH_GCAP

6.2.8.20.1 Conformance Requirement

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

public byte getCapacity()

6.2.8.20.1.1 Normal execution

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

6.2.8.20.2 Test suite files

Test Script: API 2 PRH GCAP 1.scr

Test Applet: API 2 PRH GCAP 1.java

Load Script: API 2 PRH GCAP 1.ldr

Cleanup Script: API 2 PRH GCAP 1.clr

Parameter File: API 2 PRH GCAP 1.par

6.2.8.20.3 Test Procedure

ld	<u>Description</u>	API Expectation	APDU Expectation
1	ProactiveResponseHandler available		
	1- Send envelope SMS-PP Formatted	1- Applet is triggered	0.04.77
	2- The applet sends a proactive command 3- Fetch the proactive command and send		2- 91 XX 3- The proactive command
	Terminal Response		is fetched
	4- The applet calls method getCapacity() method	4-No exception is thrown	
	5- The applet calls method getLength() method	5- The Capacity result is greater or	
		equal to getLength() result	

6.2.8.20.4 Test Coverage

CRR number	Test case number
<u>N1</u>	<u>1</u>

6.2.8.21 Method getChannelIdentifier

Test Area Reference: API_2_PRH_GCID

6.2.8.21.1 Conformance Requirement:

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

public byte getChannelIdentifier()

throws ToolkitException

6.2.8.21.1.1 Normal execution

CRRN1:The method shall return the channel identifier byte value.

CRRN2:The channel identifier byte value returned shall be from the first Channel status TLV element.

CRRN3: If the element is available it becomes the currently selected TLV.

6.2.8.21.1.2 Context errors

<u>CRRC1</u>: The method shall throw ToolkitException (UNAVAILABLE_ELEMENT) if the Channel status TLV is <u>not present.</u>

<u>CRRC2</u>: The method shall throw ToolkitException (OUT_OF_TLV_BOUNDARIES) if the Simple TLV Channel <u>Status length is equal to 0.</u>

6.2.8.21.2 Test suite files

Parameter File:

Test Script: API 2 PRH GCID_1.scr
Test Applet: API 2 PRH GCID_1.java
Load Script: API 2 PRH GCID_1.ldr
Cleanup Script: API 2 PRH GCID_1.clr

API 2 PRH GCID 1.par

6.2.8.21.3 Test Procedure

ld	Description	API Expectation	APDU Expectation
0	Applet1 is installed with maximum number		
1	of channel = 01. Channel status TLV is not present		1- DISPLAY TEXT
			Proactive command is
	1- Build and send a DISPLAY TEXT command		fetched.
	2- Call		
	<pre>ProactiveResponseHandler.getChannelIdentif ier() method.</pre>		TERMINAL RESPONSE
		2- UNAVAILABLE ELEMENT ToolkitException is thrown	with no Channel status TLV available.
2	Channel status TLV with a length equal to 0	TOOIRIEXCEPTION IS THOWN	1- OPEN CHANNEL
-	1- Build and send a OPEN CHANNEL proactive		Proactive command is
	command		fetched.
	2- Call	2- OUT_OF_TLV_BOUNDARIES	
	<pre>ProactiveResponseHandler.getChannelIdentif ier() method.</pre>	ToolkitException is thrown	TERMINAL RESPONSE
			with Channel status TLV
3	Get channel identifier value	2- Returns 0x01	length equal to 0. 1- OPEN CHANNEL
_			Proactive Command is
	1- Call ProactiveHandler.init() and ProactiveHandler.send() methods to open a		fetched.
	channel.		TERMINAL RESPONSE is
	2- Call		issued with channel status
	ProactiveResponseHandler.getChannelIdentif		<u>value = 0x8100.</u>
	<pre>ier() method.</pre>		
	3- Call (1) (1) (1) (1)		
	ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods.		
4	Get channel identifier value with 2 TLV	2- Returns 0x01	1- OPEN CHANNEL Proactive Command is
	1- Call ProactiveHandler.init()and		fetched.
	ProactiveHandler.send() methods to open a channel		TERMINIAL DECRONCE :-
			TERMINAL RESPONSE is issued with channel status
	2- Call_ ProactiveResponseHandler.getChannelIdentif		value = 0x8100 and 0x8200.
	ier()		
	3- Call		
	ProactiveHandler.initCloseChannel() and		
<u>5</u>	ProactiveHandler.send() methods. Channel status TLV is currently selected TLV	2- Returns 0x03	1- OPEN CHANNEL
			Proactive Command is
	1- Call ProactiveHandler.init() and ProactiveHandler.send() methods to open a	3- Check getChannelIdentifier()	fetched.
	channel.	=getValueByte(0)	TERMINAL RESPONSE is
	ViewHandler.FindTLV with Device Identity Tag.		issued with channel status
			$\underline{\text{value} = 0x0305}_{-}$
	<pre>2- Call_ ProactiveResponseHandler.getChannelIdentif</pre>		
	ier() method.		
	3- Compare		
	ProactiveResponseHandler.getChannelIdentif		
	<pre>ier() and then ViewHandler.getValueByte(0) methods.</pre>		

6.2.8.21.4 Test Coverage

CRR number	Test case number
<u>N1</u>	<u>3</u>
<u>N2</u>	<u>4</u>
<u>N3</u>	<u>5</u>
<u>C1</u>	<u>1</u>

<u>C2</u> <u>2</u>

6.2.8.22 Method copyChannelData

Test Area Reference: API 2 PRH CCHD BSS

6.2.8.22.1 Conformance Requirement:

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

6.2.8.22.1 Normal execution

CRRN1: The method shall copy a part of the Channel data string field.

<u>CRRN2</u>: The Channel data string field value returned shall be the first Channel data TLV element of the current response data field.

CRRN3: If the element is available it becomes the currenly selected TLV.

CRRN4: Returns dstOffset + dstLength.

6.2.8.22.2 Parameters error

CRRP1: If dstBuffer is null, a NullPointerException is thrown.

<u>CRRP2</u>: If dstOffset or dstLength parameter is negative an ArrayIndexOutOfBoundsException exception is thrown and no copy is performed.

<u>CRRP3</u>: If dstOffset+dstLength is greater than dstBuffer.length, the length of the dstBuffer array an ArrayIndexOutOfBoundsException exception is thrown and no copy is performed.

<u>CRRP4</u>: If dstLength is greater than the value field of the available TLV, a OUT_OF_TLV_BOUNDARIES <u>ToolkitException is thrown.</u>

6.2.8.22.3 Context errors

<u>CRRC1</u>: The method shall throw a <u>UNAVAILABLE_ELEMENT ToolkitException</u> if the Result TLV is not <u>present.</u>

6.2.8.22.2 Test suite files

Test Script: API 2 PRH CCHD BSS 1.scr

Test Applet: API 2 PRH CCHD BSS 1.java

Load Script: API 2 PRH CCHD BSS 1.ldr

Cleanup Script: API 2 PRH CCHD BSS 1.clr

Parameter File: API 2 PRH CCHD BSS 1.par

6.2.8.22.3 Test Procedure

ld	Description	API Expectation	APDU Expectation
	1- Applet1 is installed with maximum	AFT Expectation	2- OPEN CHANNEL
<u>0</u>	number of channel = 01.		
	Trumber of Chamier - of:		proactive command is
	2- Applet1 builds proactive commands OPEN		<u>fetched</u>
	CHANNEL with init() method in order to		
	open one channel.		TERMINAL RESPONSE is
	ProactiveHandler.send() method is called.		issued with Channel Id = 01
1	CopyChannelData() with NULL dstBuffer		RECEIVE DATA Proactive
_			command is fetched.
	Build and send a RECEIVE DATA command	NullPointerException is thrown	
		Train onto Excoption to the own	TERMINAL RESPONSE
			with not empty Channel
			Data TLV is issued.
	Call		Data 12 V 13 133dcd.
	ProactiveResponseHandler.copyChannelData dstBuffer = NULL		
	DstOffset = 0		
	DstLength = 1		
2	CopyChannelData() with negative dstOffset		1- RECEIVE DATA
_	Sopy Sharmor Bata() With Hogative dot Shoot		proactive command is
	1- call init() method for the RECEIVE DATA	2- an	fetched.
	proactive command.	ArrayIndexOutOfBoundsException	reterieu.
		exception is thrown.	TERMINAL RESPONSE
	<u>2- call</u>	<u>елоерион із шиомп.</u>	with 6 bytes avalaible
	ProactiveResponseHandler.copyChannelData()	3- no copy is performed.	('Hello1')
	DstBuffer.length = 6	3- no copy is performed.	(Hello I)
	DstOffset = -1		
	DstLength = 1		
	3- check dstBuffer is empty.		
3	CopyChannelData() with negative dstLength	1- an	
<u> </u>	ObyonamiciData() with negative distrength	ArrayIndexOutOfBoundsException	
	1- call	exception is thrown.	
	ProactiveResponseHandler.copyChannelData()	exception is thrown.	
	DstBuffer.length = 6	2 no convict performed	
	DstOffset = 0	2- no copy is performed.	
	DstLength = -1		
4	2- check dstBuffer is empty.	4	
4	CopyChannelData() with dstOffset+dstLength	1- an	
	greater than dstBuffer.length	<u>ArrayIndexOutOfBoundsException</u>	
	1- call	exception is thrown.	
	ProactiveResponseHandler.copyChannelData()		
	with dstOffset+dstLength greater than	2- no copy is performed.	
	dstBuffer.length.		
	DstBuffer.length = 6		
	DstOffset = 5		
	DstLength = 2		
	O should det profession to some		
	2- check dstBuffer is empty.		
<u>5</u>	CopyChannelData() with dstLength too large	a OUT_OF_TLV_BOUNDARIES	
2	CopyonannerData() with ustbength too large	ToolkitException is thrown.	
	Call	TOURILL ACEPTION IS UNOWIL	
	ProactiveResponseHandler.copyChannelData()		
	with dstLength greater than the value		
	field of the available TLV.		
	DstBuffer.length = 6		
	DstOffset = 0		
	DstLength = 10		
6	ConvChannelDate() without Channel Date TIV		1- RECEIVE DATA
<u>6</u>	CopyChannelData() without Channel Data TLV	•	proactive command is
	<u>element</u>	2 ALINIAVALLADI E ELEMENT	fetched
	1- call init() method for the RECEIVE DATA	2- a UNAVAILABLE_ELEMENT	<u>retoried</u>
	proactive command.	ToolkitException is thrown.	TEDMINIAL DESPONSE
	Call send() method.		TERMINAL RESPONSE
			without ChannelData TLV
	<u>2- call</u>		<u>element.</u>
	ProactiveResponseHandler.copyChannelData()		
	DstBuffer.length = 10		
	DstOffset = 0		

83

DstLength = 10	

	Successful copyChannelData()		1- RECEIVE DATA
7	Call init() method for the RECEIVE DATA proactive command. Call send() method.	3- the Channel Data TLV is copied into dstBuffer.	proactive command is fetched
	2- Call findTLV() with TAG of DEVICE IDENTITY.	The applet checks the returned value is dstOffset + dstLength = 6.	TERMINAL RESPONSE with one Channel data TLV element. (6 bytes available = 'Hello2')
	3- Call_ ProactiveResponseHandler.copyChannelData() DstBuffer.length = 6 DstOffset = 0 DstLength = 6 DstBuffer is the whole Buffer.		
8	Compare copied Buffer Check dstBuffer.	The applet checks that dstBuffer contains the channel data from the TERMINAL RESPONSE.	
9	Check the Channel Data TLV is selected	The returned byte is the same than the first byte of the Channel data TLV (i.e. 'H')	
10	Call the ViewHandler.getValueByte(0) method		
10	Successful copyChannelData() Call ProactiveResponseHandler.copyChannelData() DstBuffer.length = 6 DstOffset = 2 DstLength = 3	The Channel Data TLV is copied into dstBuffer. The applet checks the returned value is dstOffset + dstLength = 5.	
11	DstBuffer is a part of Buffer. Compare copied Buffer		
111	Compare copied Buller	The applet checks that bytes from 2	
	Check dstBuffer.	to 4 of dstBuffer contain the first 3 bytes of channel data TLV from the TERMINAL RESPONSE.	
12	Successful copyChannelData() 1- Initialise dstBuffer to [00, 01]	bytes of channel data TLV from the	
12	Successful copyChannelData() 1- Initialise dstBuffer to [00, 01] 2- Call ProactiveResponseHandler.copyChannelData() DstBuffer.length = 6 DstOffset = 2 DstLength = 3	bytes of channel data TLV from the TERMINAL RESPONSE. 2- The Channel Data TLV is copied	
12	Successful copyChannelData() 1- Initialise dstBuffer to [00, 01] 2- Call ProactiveResponseHandler.copyChannelData() DstBuffer.length = 6 DstOffset = 2	bytes of channel data TLV from the TERMINAL RESPONSE. 2- The Channel Data TLV is copied into dstBuffer. The returned value is dstOffset +	
	Successful copyChannelData() 1- Initialise dstBuffer to [00, 01] 2- Call ProactiveResponseHandler.copyChannelData() DstBuffer.length = 6 DstOffset = 2 DstLength = 3 DstBuffer is a part of buffer. Compare copied Buffer	bytes of channel data TLV from the TERMINAL RESPONSE. 2- The Channel Data TLV is copied into dstBuffer. The returned value is dstOffset + dstLength = 5. The applet checks that only bytes from 2 to 4 of dstBuffer have been updated with the first 3 bytes of channel data TLV from the TERMINAL RESPONSE.	1- RECEIVE DATA proactive command is fetched TERMINAL RESPONSE with two Channel data TLV element 1st TLV: 6 bytes available = 'Hello3' 2nd TLV: 6 bytes available = 'Hello4'

Check dstBuffer.	first Channel Data TLV from the
	TERMINAL RESPONSE.

6.2.8.22.4 Test Coverage

CRR number	Test case number
<u>N1</u>	<u>7, 10, 12, 14</u>
<u>N2</u>	<u>14</u>
<u>N3</u>	<u>9</u>
<u>N4</u>	<u>8, 11, 13, 15</u>
<u>P1</u>	<u>1</u>
<u>P2</u>	<u>2, 3</u>
<u>P3</u>	<u>4</u>
<u>P4</u>	<u>5</u>
<u>C1</u>	<u>6</u>

6.2.9 Class ToolkitRegistry

6.2.9.3 Method clearEvent

Test Area Reference: API_2_TKR_CEVTB-

6.2.9.3.1 Conformance requirement:

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

6.2.9.3.1.1 Normal execution

CRRN1: A call to isEventSet() method for a cleared event should return false after a call to clearEvent.

—CRRN2: The SIM Toolkit Framework shall not trigger the applet on the occurrence of the cleared event anymore.

CRRN3: if event was EVENT_CALL_CONTROL_BY_SIM and after the call, no applet is registered to it, The SIM Toolkit Framework shall allow an applet to register to this event.

CRRN4: if event was EVENT_CALL_CONTROL_BY_SIM and one applet is still registered to these event, The SIM Toolkit Framework shall not allow an applet to register to this event.

CRRN5: if event was EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM and after the call, no applet is registered to it, The SIM Toolkit Framework shall allow an applet to register to this event.

—CRRN6: if event was EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM and one applet is still registered to these event, The SIM Toolkit Framework shall not allow an applet to set this event.

6.2.9.3.1.2 Parameters error

CRRP1: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event was EVENT_MENU_SELECTION.

CRRP2: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event was EVENT_MENU_SELECTION_HELP_REQUEST.

CRRP3: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event was EVENT_TIMER_EXPIRATION.

CRRP4: Shall throw a Toolkit Exception with reason EVENT_NOT_ALLOWED if event was EVENT_STATUS_COMMAND.

6.2.9.3.1.3 Context errors

No requirements. Context errors

<u>CRRC1</u>: shall throw javacard.framework.TransactionException - if the operation would cause the commit capacity to be exceeded.

6.2.9.3.2 Test suite files

Test Script: ____API_2_TKR_CEVTB_1.scr

Test Applet: _____API_2_TKR_CEVTB_1.java

As default but applet registers to an event list which contains all defined events in GSM 03.19 TS 43.019 [7] excepted those that aren't allowed or supported by setEvent().

Load Script: API_2_TKR_CEVTB_1.ldr

Cleanup script: API_2_TKR_CEVTB_1.clr

Parameter File: Script: API_2_TKR_CEVTB_1.ldr

Cleanup script: API_2_TKR_CEVTB_1.clr

Parameter File: API_2_TKR_CEVTB_1.par

6.2.9.3.3 Test procedure

	ld	Description		API Expectation	APDU Expectation
4		Clear ALLOWED unregistered events	<u> </u>		
	For	overta ranging from 1 to 127 evaceted			
	tho	events ranging from 1 to 127 excepted se that aren't allowed			
	(EVI	ENT_MENU_SELECTION, AT MENU_SELECTION HELD REQUEST.		ption is thrown each	
	EVE	NT_TIMER_EXPIRATION,	time.		
	EVE	NT_STATUS_COMMAND),	2- Shall ret	urn false each time.	
	CIR	e applet calls:			
	1-	-clearEvent() method			
	2	isEventSet() method			
	1	Clear ALLOWED unregistered events			
	_				
		For events ranging from -1, 1 to 24 and 127* excepted those that aren't allowed		exception is thrown eacl	<u>n</u>
		(7, 8, 11, 19), the applet calls:	tillio.		
		1- clearEvent() method	2- Sha	all return false each time	<u>s</u>
		2- isEventSet() method			
2		Clear registered events			
∠		Glear registered events			
	1 + b c	For each ALLOWED and SUPPORTED events, applet calls setEvent() method.			
	the	appiet carrs seesvene() meenou.	1- No exce	ption shall be thrown.	
	2	For events ranging from 1 to 127-	2.1- No exc	eption shall be thrown.	
		let calls:			
	2 1	alcomErrort() mothed	2.2- Shall re	eturn false.	
	2.2	-clearEvent() method -isEventSet() method			
		Clear registered events	1		
	<u>2</u>	Clear registered events			
	_		ıt.		
	<u>∠</u>	1- For each ALLOWED and SUPPORTED ever (-1, 1 to 24 and 127)* excepted those to	hat		
	<u>4</u>	1- For each ALLOWED and SUPPORTED ever (-1, 1 to 24 and 127)* excepted those taren't allowed (7, 8, 11, 19), the app.	hat et	exception shall be throw	<u>/n.</u>
	₹	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those taren't allowed (7, 8, 11, 19), the applicalls setEvent() method.	et 1- No		
	<u>2</u>	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those taren't allowed (7, 8, 11, 19), the applicalls setEvent() method. 2- For each ALLOWED and SUPPORTED even	1- No	exception shall be throw	
	<u> </u>	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those taren't allowed (7, 8, 11, 19), the applicable setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those taren't allowed (7, 8, 11, 19), the applications of the set of th	1- No 1- No 1- No 1- No		
	<u> </u>	1- For each ALLOWED and SUPPORTED ever (-1, 1 to 24 and 127)* excepted those taren't allowed (7, 8, 11, 19), the applicalls setEvent() method. 2- For each ALLOWED and SUPPORTED ever (-1, 1 to 24 and 127)* excepted those targets.	1- No 1- No 1- No 1- No	exception shall be thro	
	<u>2</u>	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those to aren't allowed (7, 8, 11, 19), the applicalls setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those to aren't allowed (7, 8, 11, 19), the applicalls: 2.1- clearEvent() method	1- No 1- No 1- No 1- No	exception shall be thro	
	<u>2</u>	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those taren't allowed (7, 8, 11, 19), the applicalls setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those taren't allowed (7, 8, 11, 19), the applicalls:	1- No 1- No 1- No 1- No	exception shall be thro	
3	2	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those to aren't allowed (7, 8, 11, 19), the applicalls setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those to aren't allowed (7, 8, 11, 19), the applicalls: 2.1- clearEvent() method	1- No 1- No 1- No 1- No	exception shall be thro	
3	<u>∠</u>	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls: 2.1- clearEvent() method 2.2- isEventSet() method	1- No 1- No 1- No 1- No	exception shall be thro	
3	For EVE	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls: 2.1- clearEvent() method 2.2- isEventSet() method Clearing NOT ALLOWED events each event among: NT_MENU_SELECTION,	1- No 2.1- No 2.2- Sr	exception shall be through	
3	For EVER EVER	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicable setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicable: 2.1- clearEvent() method 2.2- isEventSet() method Clearing NOT ALLOWED events each event among: NT_MENU_SELECTION, NT_MENU_SELECTION, NT_MENU_SELECTION, NT_TIMER_EXPIRATION,	1- No 2.1- No 2.1- No 2.2- Sh 1- Each tim throw a Too	exception shall be through the shall return false. The clear Event shall likit Exception with	
3	For EVER EVER	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicable setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicable: 2.1- clearEvent() method 2.2- isEventSet() method Clearing NOT ALLOWED events each event among: NT_MENU_SELECTION, NT_MENU_SELECTION, NT_MENU_SELECTION, NT_MENU_SELECTION_HELP_REQUEST,	1- No 2.1- No 2.1- No 2.2- Sh 1- Each tim throw a Too	exception shall be through the shall return false.	
ণ	For EVER EVER	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicable setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicable: 2.1- clearEvent() method 2.2- isEventSet() method Clearing NOT ALLOWED events each event among: NT_MENU_SELECTION, NT_MENU_SELECTION, NT_MENU_SELECTION, NT_TIMER_EXPIRATION,	1- No 2.1- No 2.1- No 2.2- Sh 1- Each tim throw a Too	exception shall be through the shall return false. The clear Event shall likit Exception with	
3	For EVER EVER	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicable setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicable: 2.1- clearEvent() method 2.2- isEventSet() method Clearing NOT ALLOWED events each event among: NT_MENU_SELECTION, NT_MENU_SELECTION_HELP_REQUEST, NT_TIMERU_SELECTION, NT_TIMERU_SELECTION, NT_STATUS_COMMAND	1- No 2.1- No 2.1- No 2.2- Sh 1- Each tim throw a Too	exception shall be through the shall return false. The clear Event shall likit Exception with	
3	For EVER EVER	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicable setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicable: 2.1- clearEvent() method 2.2- isEventSet() method Clearing NOT ALLOWED events each event among: NT_MENU_SELECTION, NT_MENU_SELECTION_HELP_REQUEST, NT_TIMERU_SELECTION, NT_TIMERU_SELECTION, NT_STATUS_COMMAND	1- No 2.1- No 2.1- No 2.2- Sh 1- Each tim throw a Too	exception shall be through the shall return false. The clear Event shall likit Exception with	
3	For EVEN EVEN EVEN 1 met1	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls: 2.1- clearEvent() method 2.2- isEventSet() method Clearing NOT ALLOWED events each event among: NT_MENU_SELECTION, NT_STATUS_COMMAND The applet calls clearEvent(event) and.	1- No 2.1- No 2.1- No 2.2- Sh 1- Each tim throw a Too	exception shall be through the shall return false. The clear Event shall likit Exception with	
3	For EVEN EVEN EVEN 1 met1	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls: 2.1- clearEvent() method 2.2- isEventSet() method Clearing NOT ALLOWED events each event among: NT_MENU_SELECTION, NT_MENU_SELECTION, NT_STATUS_COMMAND The applet calls clearEvent(event) and. Clearing NOT ALLOWED events For each event among: EVENT_MENU_SELECTION,	1- Each time throw a Tool reason EVE	ne, clearEvent shall-lkit Exception with NT_NOT_ALLOWED.	wn.
7	For EVEN EVEN EVEN 1 met1	1- For each ALLOWED and SUPPORTED ever (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls setEvent() method. 2- For each ALLOWED and SUPPORTED ever (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls: 2.1- clearEvent() method 2.2- isEventSet() method Clearing NOT ALLOWED events each event among: NT_MENU_SELECTION, NT_STATUS_COMMAND The applet calls clearEvent(event) and. Clearing NOT ALLOWED events Clearing NOT ALLOWED events Clearing NOT ALLOWED events	1- Each time throw a Too reason EVE	ne, clearEvent shall-lkit Exception with NT_NOT_ALLOWED.	wn.
3	For EVEN EVEN EVEN 1 met1	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls: 2.1- clearEvent() method 2.2- isEventSet() method Clearing NOT ALLOWED events each event among: NT_MENU_SELECTION, NT_STATUS_COMMAND The applet calls clearEvent(event) and. Clearing NOT ALLOWED events Clearing NOT ALLOWED events For each event among: EVENT_MENU_SELECTION, EVENT_MENU_SELECTIO	1- Each time throw a Too reason EVE	ne, clearEvent shall-likit Exception with NT_NOT_ALLOWED.	wn.
3	For EVEN EVEN EVEN 1 met1	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls: 2.1- clearEvent() method 2.2- isEventSet() method Clearing NOT ALLOWED events each event among: httmenu_Selection_Help_Request, httmenu_Selection_Help_Request, httmenu_Selection_Help_Request, httmenu_Selection_Help_Request, httmenu_Selection_Help_Request, httmenu_Selection_Help_Request, httmenu_Selection_Help_Request, httmenu_Selection_Help_Request, https://doi.org/10.1006/journal.pub.	1- Each time throw a Too reason EVE	ne, clearEvent shall-lkit Exception with NT_NOT_ALLOWED.	wn.
्	For EVEN EVEN EVEN 1 met1	1- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls setEvent() method. 2- For each ALLOWED and SUPPORTED even (-1, 1 to 24 and 127)* excepted those is aren't allowed (7, 8, 11, 19), the applicalls: 2.1- clearEvent() method 2.2- isEventSet() method Clearing NOT ALLOWED events each event among: NT_MENU_SELECTION, NT_STATUS_COMMAND The applet calls clearEvent(event) and. Clearing NOT ALLOWED events For each event among: EVENT_MENU_SELECTION, EVENT_MENU_SELECTION, EVENT_MENU_SELECTION, EVENT_MENU_SELECTION, EVENT_MENU_SELECTION, EVENT_MENU_SELECTION, EVENT_TIMER_EXPIRATION, EVENT_TIMER_EXPIRATION, EVENT_STATUS_COMMAND	1- Each time throw a Too reason EVE	ne, clearEvent shall-likit Exception with NT_NOT_ALLOWED.	wn.

4	1	Checking applet isn't triggered by an IVELOPE(SMS-PP DOWNLOAD) command reset and initialize the card An ENVELOPE(SMS PP DOWNLOAD) is sent a TAR referencing applet.	EN\	let is not trigged by an- /ELOPE(SMS-PP- WNLOAD) command		
	4	Checking applet isn't triggered by an ENVELOPE(SMS-PP DOWNLOAD) comma 1 - reset and initialise the card 2 - An ENVELOPE(SMS-PP DOWNLOAD) is ser with a TAR referencing applet.		Applet is not trigged by an ENVELOPE(SMS-PP DOWNLOAD) command		

*Note: Although the method clearEvent is defined for a range from -128 to 127 only the allowed events are tested here, because the range from -128 to -2 is reserved for propriatary use in 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]

6.2.9.3.4 Test Coverage

CRR number	Test case number				
N1	1, 2				
<u>N1</u>	<u>1,2</u>				
N2	4				
N3	Framework				
N4	Framework				
N5	Framework				
N6	Framework				
P1	3				
P2	3				
P3	3				
P4	3				
<u>C1</u>	not testable				

6.2.9.9 Method is Event Set

Test Area Reference: API_2_TKR_IEVSB-

6.2.9.9.1 Conformance requirement:

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

public boolean isEventSet(byte event)

6.2.9.9.1.1 Normal execution

-CRRN1: shall return true if the event is set in the Toolkit Registry for the applet-

-CRRN2: shall return false if the event isn't set in the Toolkit Registry for the applet-

6.2.9.9.1.2 Parameters error

No requirements.

6.2.9.9.1.3 Context errors

No requirements.

6.2.9.9.2 Test suite files

Test Script: ____API_2_TKR_IEVSB_1.scr

Test Applet: API_2_TKR_IEVSB_1.java

Installation parameter: API 2 TKR IEVSB 1.java					
API_2_TKR_IEVSB_2.java					
Installation parameter:					
Same as default applet but with:					
- Maximum text length for a menu entry: 15					
- Maximum number of menu entries:1					
- Position / Identifier for each menu entry: '01'/'01'					
- Maximum number of timers:1					
Load Script:API_2_TKR_IEVSB_1.ldr					
Cleanup script:API_2_TKR_IEVSB_1.clr					
Parameter File:API_2_TKR_IEVSB_1.par					

6.2.9.9.3 Test procedure

	ld	Description		API Expectation	APDU Expectation
4		Events aren't set		•	
	rang	et calls isEventSet() for each events- ying from 1 to 127 excepted- IT_FORMATTED_SMS_PP_ENV and IT_MENU_SELECTION.	Sha l	I return false each time.	
	1	Install Applet1 only registered to EVENT FORMATTED SMS PP ENV and EVENT_MENU_SELECTION	<u>1</u>		
		Test that events aren't set Applet calls isEventSet() for each ever ranging from -1, 1 to 24 and 127* excep		Shall return false each time.	
2		EVENT_FORMATTED_SMS_PP_ENV (2) and EVENT_MENU_SELECTION (7). FOR EVENT_FORMATTED_SMS_PP_ENV			
		rentSet (EVENT_FORMATTED_SMS_PP_ENV)	Shal	I return true.	
	2	For EVENT FORMATTED SMS PP ENV	/	<u> </u>	
	_	isEventSet(EVENT_FORMATTED_SMS_PP_ENV)	_	Shall return true.	
3		For EVENT_MENU_SELECTION			
	isEv	rentSet (EVENT_MENU_SELECTION)	Shal	I return true	
1	3	For EVENT MENU SELECTION		•	
		isEventSet(EVENT_MENU_SELECTION)		Shall return true	
4	2- 	After clearing EVENT_FORMATTED_SMS_PP_ENV clearEvent(EVENT_FORMATTED_SMS_PP_ENV) ioEventSet(EVENT_FORMATTED_SMS_PP_ENV)		lo exception shall be thrown. Shall return false.	
	4	After clearing EVENT FORMATTED SMS PP ENV 1- clearEvent(EVENT_FORMATTED_SMS_PP_E 2- isEventSet(EVENT_FORMATTED_SMS_PP_E		No exception shall be thrown. Shall return false.	<u>.</u>
5		Setting events			
	1—for	gotErront() applot gallg:	1.1- thro	No exception shall be	
	1.1	setEvent() method	1 2	Shall return true each time.	
	1.2	isEventSet() method.	1.2	Shair feturi true each time.	
	<u>5</u>	Setting events For all allowed events defined in TS 43.019-[7] for method setEvent(): 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_PP_UPD, EVENT_UNFORMATTED_SMS_CB,		1- No exception shall be thrown.2- Shall return true each time.	<u>.</u>

		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
		EVENI_UNRECOGNIZED_ENVEROPE
		applet calls:
		1- setEvent() method
		2- isEventSet() method
6	Ϊ	For
💆		• • •
	EV	ENT_MENU_SELECTION_HELP_REQUEST
	1	-isEventSet - 1- Shall return false.
	(EVI	ENT_MENU_SELECTION_HELP_REQUEST)
	2	call changeMenuEntry with help
	sup	3- Shall return true
	3	-isEventSet
	(EVI	ENT MENU SELECTION HELP REQUEST)
<u> </u>	6	For EVENT MENU SELECTION HELP
	<u>6</u>	
		<u>REQUEST</u>
		1- Shall return false.
		1- isEventSet(EVENT_MENU_SELECTION_HELP_
		REQUEST)
		2- call changeMenuEntry() with help
		supported
		545-51
		3- isEventSet(EVENT_MENU_SELECTION_HELP_ 3- Shall return true.
		REQUEST)
		REQUESTY
7	1	For EVENT_TIMER_EXPIRATION
+		
		4- Shall return false.
	+	isEventSet(EVENT_TIMER_EXPIRATION)
	2	call allocateTimer() inFugntSet (FVENT TIMER FYDIRATION) 3- Shall return true
	3	-iseventSet (EVENT_TIMER_EXPIRATION) 3- Shall return true
	<u>7</u>	For EVENT TIMER EXPIRATION
	-	
		1- isEventSet(EVENT_TIMER_EXPIRATION) 1- Shall return false.
		2- call allocateTimer()
		3- isEventSet(EVENT_TIMER_EXPIRATION) 3- Shall return true.
		5 222, 61-65-6 (DV 2617_2 21-164_2017)
		F EVENT CTATIC COMMAND
8		For EVENT_STATUS_COMMAND
	1-	iseventset (EVENT_STATUS_COMMAND) 1- Shall return false.
	2	-call
	req	uestPollInterval(POLL_SYSTEM_DURATION) 3- Shall return true
	3	isEventSet(EVENT_STATUS_COMMAND)
	8	For EVENT_STATUS_COMMAND
	~	
		1- isEventSet(EVENT_STATUS_COMMAND)
		2- call requestPollInterval(POLL_SYSTEM_
		O Ondi Total i tradi
		3- isEventSet(EVENT_STATUS_COMMAND)
ļ l		
	<u>9</u>	Install Applet2 only registered to
	-	EVENT FORMATTED_SMS_PP_ENV
		is Event Set (EVENT MENT) SET ESTION)
!		isEventSet(EVENT_MENU_SELECTION) Shall return false.

*Note: Although the method isEventSet() 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 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]

6.2.9.9.4 Test Coverage

CRR number	Test case number
N1	2, 3, 4, 5, 6, 7, 8
N2	1, 5, 6, 7, 8
<u>N1</u>	<u>2,3,4,5,6,7,8</u>
<u>N2</u>	<u>1,5,6,7,8,9</u>

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.

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: this the method shall accept all the events defined in GSM 0319 excepted: 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.

<u>CRRC3</u>: shall throw a ToolkitException with TAR NOT DEFINED if event is FORMATTED SMS PP ENV and the applet has no TAR defined.

<u>CRRC4</u>: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_PP_UPD and the applet has no TAR defined.

<u>CRRC5</u>: shall throw a ToolkitException with TAR NOT DEFINED if event is FORMATTED SMS CB ENV and the applet has no TAR defined.

<u>CRRC6</u>: 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
Load Script:	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 24 instances.
Cleanup script:	API_2_TKR_SEVTB_1.clr
Parameter File:	API_2_TKR_SEVTB_1.par

6.2.9.12.3 Test Procedure

	ld	Description		API Expectation	APDU Expectation
4		Applet 1 is triggered by ENVELOPE(SMS_PP_FORMATTED) command.	Арі	olet 1 shall be triggered	
	1	Applet 1 is triggered by ENVELOPE(SMS PP_FORMATTED) command. Send ENVELOPE(SMS_PP_FORMATTED)	<u> </u>	Applet 1 shall be triggered	
2	1—1±VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV	ENT_EVENT_DOWNLOAD_USER_ACTIVITY, ENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILAB	1.2 1.3 thro 1.4 1.5	- No exception shall be own Shall return false No exception shall be own Shall return true No exception shall be own.	

EV	ENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION,		
EV	ENT_EVENT_DOWNLOAD_BROWSER_TERMINATION		
1.	1 clearEvent(event)		
1.	2 isEventSet(event)		
1	3 setEvent(event)		
1.	4 isEventSet(event)		
+	5 clearEvent(event)		
2	Setting ALLOWED and SUPPORTED events		
	1		
	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_FROFTHE_DOWNHOAD, 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_LOCATION_STATUS		
	EVENT_EVENT_DOWNLOAD_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_USER_ACTIVITY,		
	EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILAB		
	LE,		
	EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS,		
	EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION,		
	EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION		
	<u>'</u>		
	EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE,		
	EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS,		
	EVENT_FIRST_COMMAND_AFTER_SELECT,	1.1- No exception shall be	
	EVENT_UNRECOGNIZED_ENVELOPE	thrown.	
	1 1 (2) (2) (2) (2) (3)		
	1.1- clearEvent(event)	1.2- Shall return false.	
	1.2- isEventSet(event)	1.2 Ondir rotarr raiso.	
	1.2 ISEVENEDEC (CVENE)	1.3- No exception shall be	
	1.3- setEvent(event)	thrown.	
		THOWIL	
	1.4- isEventSet(event)	1.4 Shall return true	
		1.4- Shall return true.	
	1.5- clearEvent(event)	4.E. No expension of all be	
		1.5- No exception shall be	
_	F	thrown.	
3	Event 0	0.000	
	Call setEvent(0)	Shall throw a ToolkitException with	
	Call SetEvent(0)	EVENT_NOT_SUPPORTED	
		reason code.	
4	Setting EVENT_MENU_SELECTION		
		Shall throw a ToolkitException with	
	Call setEvent(EVENT_MENU_SELECTION)	EVENT_NOT_ALLOWED reason	
		code.	
-		†	
5	Setting		
5			
5	EVENT_MENU_SELECTION_HELP_REQUES	Shall throw a ToolkitException with	
5		Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason	
5	EVENT_MENU_SELECTION_HELP_REQUES	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason	
5	EVENT_MENU_SELECTION_HELP_REQUES T	Shall throw a ToolkitException with	
5	EVENT_MENU_SELECTION_HELP_REQUES T	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason	
6	EVENT_MENU_SELECTION_HELP_REQUES T Call setEvent(EVENT_MENU_SELECTION_HELP_REQUE	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason	
	Call setEvent(EVENT_MENU_SELECTION_HELP_REQUEST)	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason	

		leada	
		code.	
7	Setting EVENT_STATUS_COMMAND Call setEvent(EVENT_STATUS_COMMAND)	Shall throw a ToolkitException with EVENT_NOT_ALLOWED reason code.	
8	Setting EVENT_CALL_CONTROL_BY_SIM	No Exception shall be thrown	
	Call setEvent(EVENT_CALL_CONTROL_BY_SIM)		
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 triggered by an ENVELOPE(CALL_CONTROL_BY _SIM)	
11	Check applet is triggered by an ENVELOPE(MO_SHORT_MESSAGE_CONTR OL_BY_SIM) Trigger the Applet	Applet is triggered by an ENVELOPE(MO_SHORT_MESSA GE_CONTROL_BY_SIM)	
12	Applet 2 is triggered by ENVELOPE(SMS_PP_DOWNLOAD) command. Trigger the applet 2	Applet 2 is triggered 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.	
<u>15</u>	Applet 3 with no TAR defined registers to EVENT UNFORMATTED SMS CB		
	1- send ENVELOPE(CELL_BROADCAST_DATA_ DOWNLOAD)	1- Applet 3 shall be triggered	
	2- setEvent(FORMATTED_SMS_PP_ENV)	2- ToolkitException with reason code TAR_NOT_DEFINED should be thrown	
	3- setEvent(FORMATTED_SMS_PP_UPD)	3- ToolkitException with reason code TAR NOT DEFINED should be thrown	
	4- setEvent(FORMATTED_SMS_CB_ENV)	4- ToolkitException with reason code TAR_NOT_DEFINED should be thrown	
<u>16</u>	Applet 4 registers multiple to EVENT FORMATTED SMS PP ENV		
	1- send ENVELOPE(EVENT_FORMATTED_ SMS_PP_ENV) 2- setEvent(EVENT_FORMATTED_SMS_PP_ UPD)	1- Applet 4 shall be triggered2- no Exception shall be thrown	
	3- setEvent(EVENT_FORMATTED_SMS_PP_ UPD)	3- no Exception shall be thrown	
	4- send ENVELOPE(EVENT_FORMATTED_ SMS_PP_UPD)	4- Applet 4 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
<u>N2</u>	<u>1, 8,9,10, 11, 12</u>
N3	2, 4, 5, 6, 7
<u>N3</u>	<u>2,4,5,6,7</u>
<u>N4</u>	<u>16</u>
<u>N5</u>	not testable
P1	3
P2	4
P3	5
P4	6
P5	7
C1	13
C2	14
C2 C3 C4 C5 C6	<u>15</u>
<u>C4</u>	<u>15</u>
<u>C5</u>	<u>15</u>
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.

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 <u>GSM 0319 excepted:</u> 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.
- <u>CRRC3</u>: 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.
- <u>CRRC5</u>: shall throw a ToolkitException with TAR_NOT_DEFINED if event is FORMATTED_SMS_CB_ENV and the applet has no TAR defined.
- <u>CRRC6</u>: 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
Load Script:	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
Th	ne load script installs the 24 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
Applet 1 Registering all eventList buffer		
EventList = all allowed events defined in		
GSM 0319:		
EVENT_PROFILE_DOWNLOAD,		
EVENT_FORMATTED_SMS_PP_ENV,		
EVENT_FORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_PP_ENV,		
EVENT_UNFORMATTED_SMS_PP_UPD,		
EVENT_FORMATTED_SMS_CB,		
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,	1- No exception shall be thrown.	
EVENT_EVENT_DOWNLOAD_LOCATION_STATUS,	0. N	
EVENT_EVENT_DOWNLOAD_USER_ACTIVITY,	2- No exception shall be thrown.	
EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE	3- Each time shall return true.	
EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION,	C Lacri time shair retarri trae.	
EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION.	4- No exception shall be thrown.	
1 For each event in EventList		
clearEvent(event)		
· · ·		
2 setEventList(eventList)		
Offset = 0		
Length = eventList.lentgh		
3 For all events in eventList is EventSet(event)		
4 For each event in EventList		
clearEvent(event)		
<u>Applet 1 Registering all eventList buffe</u>	<u>er</u>	
EventList = all allowed events (-1, 1	to	
24 and 127 excepted 7, 8, 11, 19) defi	ned	
<u>in TS 43.019-[7]:</u>		
<pre>EVENT_PROFILE_DOWNLOAD,</pre>		
EVENT_FORMATTED_SMS_PP_ENV,		
EVENT_FORMATTED_SMS_PP_UPD,		
EVENT_FORMATTED_SMS_CB,		
EVENT_UNFORMATTED_SMS_PP_ENV,		
<pre>EVENT_UNFORMATTED_SMS_PP_UPD,</pre>		
EVENT_UNFORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_CB,		
EVENT_UNFORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_CB, EVENT_CALL_CONTROL_BY_SIM,		
EVENT_UNFORMATTED_SMS_PP_UPD, EVENT_UNFORMATTED_SMS_CB, EVENT_CALL_CONTROL_BY_SIM, EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM,	_	
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_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_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_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_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_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_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_AVAIL	ABLE	
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_AVAIL . EVENT_EVENT_DOWNLOAD_CARD_READER_STATU EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTIO	ABLE	
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_AVAIL	ABLE	
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_AVAIL	ABLE	
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_AVAIL . EVENT_EVENT_DOWNLOAD_CARD_READER_STATU EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTIO EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATI EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS,	ABLE	
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_DISCONNECTED, EVENT_EVENT_DOWNLOAD_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_USER_ACTIVITY, EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAIL . EVENT_EVENT_DOWNLOAD_LOCATION_STATUS EVENT_EVENT_DOWNLOAD_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_LOCARD_READER_STATU EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTIO EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS,	ABLE	
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_AVAIL . EVENT_EVENT_DOWNLOAD_CARD_READER_STATU EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTIO EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATI EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS,	ABLE S, N, ON,	
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_DISCONNECTED, EVENT_EVENT_DOWNLOAD_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_USER_ACTIVITY, EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAIL . EVENT_EVENT_DOWNLOAD_LOCATION_STATUS EVENT_EVENT_DOWNLOAD_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_LOCARD_READER_STATU EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTIO EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS,	ABLE	
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_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_LOCATION_STATUS, EVENT_EVENT_DOWNLOAD_USER_ACTIVITY, EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAIL	ABLE SS, N, ON, ON, 1- No exception shall be thrown.	
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_AVAIL EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTIO EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTIO EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATI EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS, EVENT_EVENT_DOWNLOAD_ATTER_SELECT, EVENT_UNRECOGNIZED_ENVELOPE 1 - For each event in EventList	ABLE S, N, ON,	
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_AVAIL EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTIO EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTIO EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATI EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS, EVENT_EVENT_DOWNLOAD_ATTER_SELECT, EVENT_UNRECOGNIZED_ENVELOPE 1 - For each event in EventList	ABLE SS, NN, ON, 1- No exception shall be thrown. 2- No exception shall be thrown.	
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_AVAIL / EVENT_EVENT_DOWNLOAD_CARD_READER_STATU EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTIO EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATI EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS, EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS, EVENT_FIRST_COMMAND_AFTER_SELECT, EVENT_UNRECOGNIZED_ENVELOPE 1- For each event in EventList clearEvent(event)	ABLE SS, N, ON, ON, 1- No exception shall be thrown.	

			4- No exception shall be thrown.
		3- For all events in eventList isEventSet(event)	4- No exception shall be thown.
		<pre>4- For each event in EventList clearEvent(event)</pre>	
2		Registering part of eventList buffer	
E-		atList = all allowed events defined in- 0319 (see test case 1).	
1		For each event in EventList arEvent(event)	1- No exception shall be thrown.
		<pre>setEventList(eventList, offset, yth)</pre>	2- No exception shall be thrown.
0:	eng	<pre>pet > 0 yth = eventList.lentgh offset For all events in eventList:</pre>	3- Each time shall return true for events ranging from offset to-offset+length else shall return false.
i.		rentSet(event)	4- No exception shall be thrown.
4- c :		For each event in EventList: arEvent(event)	
	2	Registering part of eventList buffer	
		EventList = all allowed events defined TS 43.019-[7] (see test case 1). 1- For each event in EventList	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
	5	Out of bounds and big offset Offset = 255	Shall throw a java.lang.ArrayIndexOutOfBounds
	6	Length = 1 Offset < 0	Exception
	O	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

	Offset = 0 Length = 255	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
Rese	Check applet is triggered by an ENVELOPE(CALL_CONTROL_BY_SIM) et and initialize the card gger the applet	let is trigged by an /ELOPE(CALL_CONTROL_BY W)
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_MESSA GE_CONTROL_BY_SIM)
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.
	EVENT_MO_SHORT_MESSAGE_CONTROL_BY	

	t 3 with no TAR defined registers to /ENT_UNFORMATTED_SMS_CB		
1- send	ENVELOPE(EVENT_UNFORMATTED_SMS_CB)	1- Applet3 shall be triggered	
, EVE EVENT	entList(EVENT_FORMATTED_SMS_PP_ENV NT_UNFORMATTED_SMS_PP_ENV, _UNFORMATTED_SMS_PP_ENV)	2- ToolkitException with reason code TAR_NOT_DEFINED should be thrown	
ENV,	entList(EVENT_UNFORMATTED_SMS_PP_ EVENT_FORMATTED_SMS_PP_UPD, _UNFORMATTED_SMS_PP_ENV)	3- ToolkitException with reason code TAR_NOT_DEFINED should be thrown	
SETEV NV, E EVENT	entList(EVENT_UNFORMATTED_SMS_PP_E VENT_UNFORMATTED_SMS_PP_ENV, _FORMATTED_SMS_CB_ENV)	4- ToolkitException with reason code TAR_NOT_DEFINED should be thrown	
<u>5-</u> isEve	ntSet(EVENT_UNFORMATTED_SMS_PP_ENV	5- method should return FALSE	
6- <u>isEve</u> <u>)</u>	ntSet(EVENT_UNFORMATTED_SMS_PP_UPD	6- method should return FALSE 7- method should return FALSE	
7- isEve	ntSet(EVENT_FORMATTED_SMS_PP_ENV)	8- method should return FALSE	
	ntSet(EVENT_FORMATTED_SMS_PP_UPD)	9- method should return FALSE	
22 1- setEv	ntSet(EVENT_FORMATTED_SMS_CB_ENV) entList(EVENT_UNFORMATTED_SMS_PP_E VENT_UNFORMATTED_SMS_PP_ENV)	1- no exception should be thrown	
<u>2-</u> <u>isEve</u>	ntSet(EVENT_UNFORMATTED_SMS_PP_ENV	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		
<u>N4</u>			
<u>N5</u>	2 <u>1</u> 2 <u>2</u>		
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		
<u>C3</u>	<u>21</u>		
<u>C4</u>	<u>21</u> <u>21</u>		
C3 C4 C5 C6	<u>21</u>		
C6	not testable		

6.3 SIM Toolkit Framework

6.3.1 Minimum Handler Availability

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

6.3.1.1 ProactiveHandler

Test Area Reference: FWK_MHA_PAHD-

6.3.1.1.1 Conformance Requirement

6.3.1.1.1.1 Normal Execution

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

EVENT_FORMATTED_SMS_PP_ENV

EVENT_UNFORMATTED_SMS_PP_ENV

EVENT_FORMATTED_SMS_PP_UPD

EVENT_UNFORMATTED_SMS_PP_UPD

EVENT_FORMATTED_SMS_CB

EVENT_UNFORMATTED_SMS_CB

EVENT_MENU_SELECTION

EVENT_MENU_SELECTION_HELP_REQUEST

EVENT_TIMER_EXPIRATION

EVENT_EVENT_DOWNLOAD_MT_CALL

EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE

EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION

EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION_

EVENT_UNRECOGNIZED_ENVELOPE

EVENT_STATUS_COMMAND

EVENT_CALL_CONTROL

EVENT_SMS_MO_CONTROL

EVENT_PROFILE_DOWNLOAD

6.3.1.1.2 Parameters error

No requirements.

6.3.1.1.3 Context errors

No requirements. EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

EVENT EVENT DOWNLOAD CHANNEL STATUS

6.3.1.1.1.2 Context Errors

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

103

EVENT_FIRST_COMMAND_AFTER_SELECT

6.3.1.1.2 Test Suite Files

Test Script: ____FWK_MHA_PAHD_1.scr

Test Applet: FWK_MHA_PAHD_1.java

FWK_MHA_PAHD_2.java

Load Script: FWK_MHA_PAHD_1.ldr

Cleanup Script: FWK_MHA_PAHD_1.clr

Parameter File: FWK_MHA_PAHD_1.java

FWK_MHA_PAHD_2.java

Load Script: FWK MHA PAHD 1.ldr

Cleanup Script: FWK_MHA_PAHD_1.clr

Parameter File: FWK_MHA_PAHD_1.par

6.3.1.1.3 Test Procedure

	ld	Description	API /Framework Expectation	on APDU Expectation
4	App	olets registration to all events and Proactive Handler availability with EVENT_PROFILE_DOWNLOAD		
	def: Usin EVEN ():	letl is registered to all events ined in [7]. ing the methods initMenuEntry () for— NT_MENU_SELECTION, requestPollInterval— for EVENT_STATUS_COMMAND, allocateTimer for EVENT_TIMER_EXPIRATION and— EventList () for the rest of the— its.		
	Appidef:	let2 is registered to all events ined in [7], except NT_CALL_CONTROL_BY_SIM and NT_MO_SHORT_MESSAGE_CONTROL_BY_SIM.	1- Applet1 is triggered	
	() :	ng the methods initMenuEntry () for NT_MENU_SELECTION, requestPollInterval for EVENT_STATUS_COMMAND, allocateTimer for EVENT_TIMER_EXPIRATION and EventList () for the rest of the	2-No exception is thrown.	
	eve	nts.	3- Applet2 is triggered	
	pri	priority of applet1 is higher than- prity of applet2 erminal Profile command is sent to SIM-	4- No exception is thrown	

	ld	Description		API /Framework Expectation	n	APDU Expectation	
11 1	WI CI.	out the facility of SET_EVENT_LIST,					
	POLI	_INTERVAL, SET UP IDLE MODE TEXT and					
	SET	UP MENU.					
	2 Ar	plet1 gets the Proactive Handler					
	Appl	etl is deregistered to					
	EVEN	T_PROFILE_DOWNLOAD					
	3 Ar	plet2 gets the Proactive Handler					
	Appl	et2 is deregistered to					
	EVEN	T_PROFILE_DOWNLOAD	۸۰۰	et1 finalizes			
			~pp	let i imanzes			

ld	Description	API /Framework Expectation	APDU Expectation
<u>1</u>	Applets registration to all events and Proact	<u>tive</u>	
	Handler availability with EVENT FIRST COMMAND AFTER SELEC	ст	
	Applet1 is registered to all events defined in TS 43.019 [7].		
	Using the methods initMenuEntry() for		
	<pre>EVENT_MENU_SELECTION, requestPollInterval() for</pre>		
	<pre>EVENT_STATUS_COMMAND, allocateTimer() f</pre>		
	EVENT_TIMER_EXPIRATION and setEventList for the rest of the events.	()	
	2		
	Applet2 is registered to all events defined in TS 43.019 [7],	1- Applet1 is triggered by	
	<pre>EVENT_CALL_CONTROL_BY_SIM and EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM.</pre>	EVENT_FIRST_COMMAND_AFTE	
	Using the methods initMenuEntry() for	R_SELECT	
	<pre>EVENT_MENU_SELECTION, requestPollInterval() for</pre>		
	<pre>EVENT_STATUS_COMMAND, allocateTimer() f</pre>		
	EVENT_TIMER_EXPIRATION and setEventList for the rest of the events.	Z 71 TOORIT EXCEPTION	
		HANDLER_NOT_AVAILABLE is thrown.	
	The priority of applet1 is higher than priority of applet2		
	1- Select MF	Applet1 finalizes	
	2- Applet1 gets the Proactive Handler.		
	Applet1 is deregistered from EVENT_FIRST_COMMAND_AFTER_SELECT.		
	3- Applet2 gets the Proactive Handler		
	Applet2 is deregistered to		
	EVENT_FIRST_COMMAND_AFTER_SELECT.		
		Analyto is triangued by	
		Applet2 is triggered by EVENT_FIRST_COMMAND_AFTE	
		R_SELECT	_
		3- A Toolkit Exception	
		HANDLER_NOT_AVAILABLE is	
		thrown. Applet2 finalizes	
2 EV	Proactive Handler availability with- /ENT_MENU_SELECTION_HELP_REQUEST		
	form SIM initialization with all the- Hilities supported	1- Applet1 is triggered	
		, Applet is triggered	
	nvelope menu selection with help	2. No expention is thrown	
req	ruest is sent to the SIM	2- No exception is thrown	
2 A	pplet1 gets the Proactive Handler		
3 E	nvelope menu selection with help	Applet1 finalizes	
	puest is sent to the SIM		
	į.	3- Applet2 is triggered	
4 A	pplet2 gets the Proactive Handler		
		4- No exception is thrown	
<u>2</u>	Proactive Handler availability with EVENT PROFILE DOWNLOAD	1- Applet1 is triggered by EVENT_PROFILE_DOWNLOAD	
	1- Terminal Profile command is sent to SIM without the facility of	<u>tne</u>	
	SET_EVENT_LIST, POLL_INTERVAL, SET UP ID	2- No exception is thrown.	
	MODE TEXT and SET UP MENU.		

Ī	ld	Description		API /Framework Expectation	APDU Expectation
		2- Applet1 gets the Proactive Handler		Applet1 finalizes.	
		Applet1 is deregistered to EVENT_PROFILE_DOWNLOAD		Applet2 is triggered by EVENT_PROFILE_DOWNLOAD	
		3- Applet2 gets the Proactive Handler Applet2 is deregistered to EVENT_PROFILE_DOWNLOAD		3- No exception is thrown	
3		Proactive Handler availability with			
	1 Er SIM	EVENT_MENU_SELECTION avelope menu selection is sent to the	1- A	pplet1 is triggered	
	2 A _I	opleth gets the Proactive Handler	2- N	lo exception is thrown.	
	3 Er SIM	avelope menu selection is sent to the	Арр	let1 finalizes	
	4 A _f	oplet2 gets the Proactive Handler	3- A	pplet2 is triggered	
			4- N	o exception is thrown.	
	3	Proactive Handler availability with		o exception is thrown.	
		EVENT MENU SELECTION HELP REQUI			
		Perform SIM initialization with all the facilities supported	<u>e</u>	1- Applet1 is triggered	
		1- Envelope menu selection with help request is sent to the SIM		2- No exception is thrown	
		2- Applet1 gets the Proactive Handler		Applet1 finalizes	
4		Proactive Handler availability with			
		EVENT_FORMATTED_SMS_PP_ENV	1A	pplet1 is triggered	
	1 Er	nvelope dataDownLoad formatted is sent-		ppiotrio inggorod	
	to the SIM		2- N	o exception is thrown.	
	2 71	pplet1 gets the Proactive Handler	App	let1 finalizes	
		nvelope dataDownLoad formatted is sent- the SIM		pplet2 is triggered	
	4-A <u>r</u>	oplet2 gets the Proactive Handler	4_ N	o exception is thrown.	
	4	Proactive Handler availability with	1	o oxeoption to thio wn.	
		EVENT MENU SELECTION 1- Envelope menu selection is sent to to	the	1- Applet1 is triggered	
		SIM 2- Applet1 gets the Proactive Handler		2- No exception is thrown.	
		2 Appreci yets the Proactive name		Applet1 finalizes	
5		Proactive Handler availability with			

	ld	Description		API /Framework Expectation	n	APDU Expectation	
	EVENT_UNFORMATTED_SMS_PP_ENV		1- A	1- Applet1 is triggered		Do Exposition	
1 Envelope dataDownLoad unformatted is sent to the SIM		The state of the s					
2 Applet1 gets the Proactive Handler		2- No exception is thrown.					
			Applet1 finalizes				
			3- A	pplet2 is triggered			
	3 Applet2 gets the Proactive Handler			4- No exception is thrown.			
	<u>5</u>	Proactive Handler availability with EVENT FORMATTED SMS PP ENV		1- Applet1 is triggered			
		1- Envelope dataDownLoad formatted is s to the SIM	sent				
		2- Applet1 gets the Proactive Handler		2- No exception is thrown.			
				Applet1 finalizes			
	<u>6</u>	Proactive Handler availability with EVENT UNFORMATTED SMS PP ENV 1- Envelope dataDownLoad unformatted is sent to the SIM		1- Applet1 is triggered			
		2- Applet1 gets the Proactive Handler		2- No exception is thrown.			
				Applet1 finalizes Applet2 is triggered			
		3- Applet2 gets the Proactive Handler		3 No exception is thrown.			
6	1	Proactive Handler availability with EVENT_FORMATTED_CELL BROADCAST					
		invelope cell broadcast formatted is to the SIM	1- Applet1 is triggered				
	3 Envelope cell broadcast formatted is sent to the SIM 4 Applet2 gets the Proactive Handler 7 Proactive Handler availability with EVENT_UNFORMATTED_CELL BROADCAST 1 Envelope cell broadcast unformatted is sent to the SIM 2 Applet1 gets the Proactive Handler		2-N (o-exception is thrown			
				let1 finalizes oplet2 is triggered			
			4 -N (o-exception is thrown			
7			1- Applet1 is triggered				
				lo exception is thrown let1 finalizes			
	2 4	relat2 cate the Deceative Handler		pplet2 is triggered-			
	3 Applet2 gets the Proactive Handler		4- N	lo exception is thrown			

	ld	Description		API /Framework Expectation	APDU Expectation
	<u>7</u>	Proactive Handler availability with		·	·
		EVENT FORMATTED CELL BROADCAS	<u>ST</u>		
		1- Envelope cell broadcast formatted is	3	1- Applet1 is triggered	
		sent to the SIM			
		2- Applet1 gets the Proactive Handler			
				2-No exception is thrown	
				Applet1 finalizes	
8		Proactive Handler availability with EVENT_TIMER_EXPIRATION			
		EVENT_TIMEN_EXTRATION			
	1 T:	imer Id =1	1- A	pplet1 is triggered	
	Enve	elope Timer Expiration is sent to the	2- N	o exception is thrown.	
	2 A	pplet1 gets the Proactive Handler	Арр	let1 finalizes	
		imer id=2	2 ^	nnlot? in triggered	
	SIM	elope Timer Expiration is sent to the	3- /\	pplet2 is triggered	
	4 A:	oplet2 gets the Proactive Handler			
	8	Proactive Handler availability with	4- N	o exception is thrown	
		EVENT UNFORMATTED CELL BROADCA	AST		
		1- Envelope cell broadcast unformatted	is	1- Applet1 is triggered	
		sent to the SIM			
		2- Applet1 gets the Proactive Handler		2- No exception is thrown	
				Applet1 finalizes Applet2 is triggered	
		3- Applet2 gets the Proactive Handler			
		Proactive Handler availability with	1	3 No exception is thrown	
9		EVENT_CALL_CONTROL_BY_SIM			
	1 00	evolone call central by CIM is sent to	1- A	pplet1 is triggered	
	the	SIM			
			2- N	o exception is thrown.	
	2 A	Proactive Handler availability with			T
	<u>9</u>	Proactive Handler availability with EVENT TIMER EXPIRATION			
				4. Applete to trippe	
		1- Timer Id =1		1- Applet1 is triggered	
		Envelope Timer Expiration is sent to the SIM	ne_		
		2- Applet1 gets the Proactive Handler			
		2 Appreci gets the Froactive handler		2- No exception is thrown.	
				Applet1 finalizes	
	<u>10</u>	Proactive Handler availability with EVENT CALL CONTROL BY SIM			
				1- Applet1 is triggered	
		1- Envelope call control by SIM is sent the SIM	t to		
		_		2- No exception is thrown.	
		2- Applet1 gets the Proactive Handler			
10	E	Proactive Handler availability with- VENT_MO_SHORT_MESSAGE_CONTROL			
11	=		1- A	pplet1 is triggered	

ld	Description		API /Framework Expectation	APDU Expectation
	nvelope mo short message control by SIM			
15-6	Jent to the Sim			
2 45	plot 1 gets the Propotive Handler	2- N	lo exception is thrown	
11	plet1 gets the Proactive Handler Proactive Handler availability with			
	EVENT_MO_SHORT_MESSAGE_CONTRO	<u>JL</u>	A A COLOMA CONTROL OF THE	
	1- Envelope mo short message control by	Y	1- Applet1 is triggered	
	SIM is sent to the SIM			
			2- No exception is thrown	
11	2- Applet1 gets the Proactive Handler Proactive Handler availability with		2- NO exception is thrown	
	EVENT_EVENT_DOWNLOAD_MT_CALL			
1 170		1- A	pplet1 is triggered	
to t	the SIM			
2 A _E	ppletl gets the Proactive Handler	2- N	lo exception is thrown.	
		App	let1 finalizes	
3 Ar	oplet2 gets the Proactive Handler	3- A	pplet2 is triggered	
12	Proactive Handler availability with	4 - N	l o exception is thrown	
12	EVENT EVENT DOWNLOAD MT CALL	_		
	1- Envelope event download mt call is a	zent	1- Applet1 is triggered	
	to the SIM	Jene		
	2- Applet1 gets the Proactive Handler		2- No exception is thrown.	
			Applet1 finalizes	
	3- Applet2 gets the Proactive Handler		Applet2 is triggered	
			3-No exception is thrown	
12	Proactive Handler availability with		O 140 CACCPHOLLIS HILOWII	
EVE	NT_EVENT_DOWNLOAD_CALL_CONNECT			
	ED-			
1 Er	nvelope event download call connected			
18 	sent to the SIM	1- A	pplet1 is triggered	
2 Ar	oplet1 gets the Proactive Handler			
2.15		2- N	lo exception is thrown.	
		Арр	let1 finalizes	
3 Ar	oplet2 gets the Proactive Handler	2 ^	polot2 in triaggrad	
		3- A	pplet2 is triggered	
		4 - N	lo exception is thrown	

	ld	Description		API /Framework Expectation	APDU Expectation
ļ	<u>13</u>	Proactive Handler availability with			
		EVENT EVENT DOWNLOAD CALL CONN	ECT		
		<u>ED</u>			
		1- Envelope event download call connect	-ed		
		is sent to the SIM		- Applet1 is triggered	
			-	- Applet is triggered	
		2 Applet1 gets the Dweestire Handler			
		2- Applet1 gets the Proactive Handler	2	- No exception is thrown.	
				applet1 finalizes	
		3- Applet2 gets the Proactive Handler	A	Applet2 is triggered	
			2	No expention is thrown	
13	I	Proactive Handler availability with	<u> </u>	- No exception is thrown	1
10	EVI	ENT EVENT DOWNLOAD CALL DISCONN			
		ECTED-			
			1- App	olet1 is triggered	
	1 E1	nvelope event download call			
	disc	connected is sent to the SIM			
			2- No	exception is thrown.	
	2 A	pplet1 gets the Proactive Handler			
			Apple	t1 finalizes	
			трріс	t i manzes	
	3 A	eplet2 gets the Proactive Handler	3- Apr	olet2 is triggered	
			- 11	33	
ļ L		,	4 - No	exception is thrown.	
	<u>14</u>	Proactive Handler availability with			
		EVENT_EVENT_DOWNLOAD_CALL_DISCO	<u>NNC</u>		
		<u>ECTED</u>	1	- Applet1 is triggered	
		1- Envelope event download call	-	- Applett is triggered	
		disconnected is sent to the SIM			
		2- Applet1 gets the Proactive Handler	2	- No exception is thrown.	
			_	unplot4 finalizes	
		3- Applet2 gets the Proactive Handler		Applet1 finalizes Applet2 is triggered	
				hphere is mygered	
			3	- No exception is thrown.	
14		Applets triggering with			
		EVENT_EVENT_LOCATION_STATUS			
			1- App	olet1 is triggered	
	1 - E 1	nvelope event download location status			
	13 1	School Color Diff			
	2 A	pplet1 gets the Proactive Handler	2- No	exception is thrown.	
			_ 140	S.Coption to thrown.	
			Apple	t1 finalizes	
	3 A	pplet2 gets the Proactive Handler	3- App	olet2 is triggered	
11	Ì		4 - No	exception is thrown	

Γ	ld	Description	1	API /Framework Expectation	APDU Expectation
	<u>15</u>	Applets triggering with			1,000
		EVENT EVENT LOCATION STATUS			
		1- Envelope event download location sta	atus	1- Applet1 is triggered	
		is sent to the SIM			
		2- Applet1 gets the Proactive Handler			
		TT January	2	2- No exception is thrown.	
				Applet1 finalizes	
				Applet2 is triggered	
		3- Applet2 gets the Proactive Handler			
15		Proactive Handler availability with		3- No exception is thrown	
	EVE	ENT EVENT DOWNLOAD USER ACTIVITY			
		rvelope event download user activity is - to SIM	4 4		
	BCIT	C CO DIM	1- Ap	plet1 is triggered	
	0 3	mileti make the Duranti e wording			
	2 Ar	pplet1 gets the Proactive Handler	2- No	exception is thrown	
			A == 1 :	Ad finalines	
			Apple	et1 finalizes	
	3 Ar	pplet2 gets the Proactive Handler	3- Ap	plet2 is triggered	
				,	
			4 81	and the standard second	
	16	Proactive Handler availability with	4 - N 0	exception is thrown	
	10	EVENT EVENT DOWNLOAD USER ACTIV	VITY		
		<pre>1- Envelope event download user activit is sent to the SIM</pre>		A Anni and in this manned	
			-	1- Applet1 is triggered	
		2- Applet1 gets the Proactive Handler			
		2 Appleed gees ene floadelive manater	2	2- No exception is thrown	
				Applot1 finalizes	
				Applet1 finalizes Applet2 is triggered	
		3- Applet2 gets the Proactive Handler	-		
				S. N. C.	
16		Proactive Handler availability with	<u> </u>	3- No exception is thrown	<u> </u>
	EVI	Proactive Handler availability with ENT_EVENT_DOWNLOAD_IDLE_SCREEN_			
		AVAILABLE			
	1 -		1- Ар	plet1 is triggered	
		nvelope event download idle screen- ilable is sent to the SIM			
					
<u> </u>	<u>2 A</u> z	pplet1 gets the Proactive Handler	2- No	exception is thrown.	
	- AF	spicer good one fromeerve manarer		M. Contract	
			Apple	et1 finalizes	
			3- An	plet2 is triggered	
	3 <i>I</i>	Applet2 gets the Proactive Handler	J 7.10	F.o. 10 (11990) 00	
			4- No	exception is thrown	

Ī	ld	Description		API /Framework Expectatio	n	APDU Expectation
	<u>17</u>	Proactive Handler availability with EVENT EVENT DOWNLOAD IDLE SCREI AVAILABLE 1- Envelope event download idle screen available is sent to the SIM		1- Applet1 is triggered		
		2- Applet1 gets the Proactive Handler		2- No exception is thrown.		
		3- Applet2 gets the Proactive Handler		Applet1 finalizes Applet2 is triggered 3- No exception is thrown		
17	EVI	Proactive Handler availability with- ENT_EVENT_DOWNLOAD_CARD_READER _STATUS- nvelope_event_download_card_reader_	1- A	pplet1 is triggered		
		opleth gets the Proactive Handler	2- N	lo exception is thrown.		
	3 А ў	pplet2 gets the Proactive Handler	• • •	let1 finalizes pplet2 is triggered		
			4-N	lo exception is thrown		
18	1 Er	Proactive Handler availability with ENT_EVENT_DOWNLOAD_LANGUAGE_SE LECTION- avelope event download language ection is sent to the SIM	1- A	pplet1 is triggered		
	2 Aj	pplet1 gets the Proactive Handler	2-N (exception is thrown.		
				let1 finalizes let2 is triggered		
	3 A ₁	pplet2 gets the Proactive Handler	3-N	o exception is thrown		

Ī	ld	Description		API /Framework Expectation	APDU Expectation	_
	18	Proactive Handler availability with		·		_
		EVENT EVENT DOWNLOAD CARD REAL	<u>DER</u>			
		STATUS		4. Appletd is triangue.		
		1- Envelope event download card reader		1- Applet1 is triggered		
		status is sent to the SIM	_			
		2 Applet1 gots the Presenting Handley				
		2- Applet1 gets the Proactive Handler		2- No exception is thrown.		
				A colored Group		
				Applet 1 finalizes		
		3- Applet2 gets the Proactive Handler		Applet2 is triggered		
				3- No exception is thrown		
j †	19	Proactive Handler availability with				
	_	EVENT EVENT DOWNLOAD LANGUAGE	SE			
		<u>LECTION</u>		4. Appletd is triangue.		
		1- Envelope event download language		1- Applet1 is triggered		
		selection is sent to the SIM				
		2- Applet1 gets the Dreastine Handler		2-No exception is thrown.		
		2- Applet1 gets the Proactive Handler				
				Applet1 finalizes		
				Applet2 is triggered		
		3- Applet2 gets the Proactive Handler				
				3-No exception is thrown		
19		Proactive Handler availability with			<u> </u>	
	EVE	ENT_EVENT_DOWNLOAD_BROWSER_TER				
		MINATION .	4 ^	pplot1 is triggered		
	1 Ex	nvelope event download browser	1- /\	pplet1 is triggered		
	tern	nination is sent to the SIM				
	2 3	which gots the Duranting World or	2-N	o exception is thrown.		
	z Ař	plet1 gets the Proactive Handler				
			App	let1 finalizes		
			Δnn	let2 is triggered		
			4 dh	iciz i s inggerea		
	3 A _E	plet2 gets the Proactive Handler				
ļ L			3-N	exception is thrown		
	<u>20</u>	Proactive Handler availability with EVENT EVENT DOWNLOAD BROWSER	TED			
		MINATION	<u>ı ek</u>			
		initia i oli		1- Applet1 is triggered		
		1- Envelope event download browser				
		termination is sent to the SIM				
		2- Applet1 gets the Proactive Handler		2-No exception is thrown.		
				Applet1 finalizes		
				APPIOCI III GII 200		
				Applet2 is triggered		
		3- Applet2 gets the Proactive Handler				
		5 Apprec2 yets the Proactive mandler		3-No exception is thrown		
20		Proactive Handler availability with		S-NO exception is thrown		
=0		EVENT STATUS COMMAND				
			1- A	pplet1 is triggered		
	1 St	catus command is sent to the SIM				
			2 1	o execution is through		
	2 Ar	pletl gets the Proactive Handler	Z- \	o exception is thrown.		
			App	let1 finalizes		
	3 I	Applet2 gets the Proactive Handler	3- A	pplet2 is triggered		
			4_ N	o exception is thrown.		
I L	J		1	o caocption is thrown.	Į	

	ld	Description		API /Framework Expectation	APDU Expectation	
	<u>21</u>	Proactive Handler availability with EVENT STATUS COMMAND 1- Status command is sent to the SIM		1- Applet1 is triggered		
		2- Applet1 gets the Proactive Handler		2- No exception is thrown.		
				Applet1 finalizes		
		3- Applet2 gets the Proactive Handler		Applet2 is triggered		
				3- No exception is thrown.		
21		Proactive Handler availability with- UNRECOGNIZED_ENVELOPE-	1- Δ	pplet1 is triggered		
		n unrecognized Envelope (BER TLV Tagecognized) is sent to the SIM-		pplott is triggorou		
	2 A	pplet1 gets the Proactive Handler	2- N	lo exception is thrown.		
			Арр	let1 finalizes		
	3-Ap	plet2 gets the Proactive Handler	3- A	pplet2 is triggered		
			4-A	lo exception is thrown		

ld	Description	API /Framework Expectation	APDU Expectation
22	Proactive Handler availability with	P	1- OPEN CHANNEL
	EVENT EVENT DOWNLOAD DATA AVAILAB	2 Applot1 is triggered	proactive Command is
	<u>LE</u>	2-Applet1 is triggered	<u>fetched</u>
	1- Applet1 builds a proactive command OPEN CHANNEL proactiveHandler.send() method is	1	TERMINAL RESPONSE is
	called.	3-No exception is thrown.	issued with Channel Id = 01
		Applet1 finalizes	
	2- An Envelope Event Download Data Available is sent to the SIM, with		
	channelId=01.		
	3- Applet1 gets the Proactive Handler		
23	Proactive Handler availability with		
	EVENT EVENT DOWNLOAD CHANNEL STAT		
	<u>US</u>	1- Applet1 is triggered	
	1- An Envelope Event Download Channel		
	Status is sent to the SIM, with ChannelId=01	2- No exception is thrown.	
		Applet1 finalizes	
	2- Applet1 gets the Proactive Handler	Applet i manzes	
<u>24</u>	Proactive Handler availability with		
	UNRECOGNIZED_ENVELOPE	1- Applet1 is triggered	
	1- An unrecognized Envelope (BER TLV Tag	1 Applett is triggered	
	unrecognized) is sent to the SIM	2. No eveention is thrown	
	2- Applet1 gets the Proactive Handler	2- No exception is thrown.	
		Applet1 finalizes Applet2 is triggered	
	3-Applet2 gets the Proactive Handler	7 ppiot2 to triggorou	
25	Proactive Handler availability with	3- No exception is thrown	
<u>25</u>	EVENT FORMATTED SMS PP UPD		
	1- Update Record EFsms instruction formatted is sent to the SIM	1- Applet1 is triggered	
	2- Applet1 gets the Proactive Handler	2- No exception is thrown.	
		Applet1 finalizes	
<u>26</u>	Proactive Handler availability with		
	EVENT UNFORMATTED SMS PP UPD		
	1- Update Record EFsms instruction		
	unformatted is sent to the SIM	1- Applet1 is triggered	
	2- Applet1 gets the Proactive Handler	2- No exception is thrown.	
		Applet4 finelizes	
	3- Applet2 gets the Proactive Handler	Applet1 finalizes	
	_	3- Applet2 is triggered	
		4- No exception is thrown.	
		T- 140 CVCCh(IOH 19 (1110MH)	

6.3.1.1.4 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				

6.3.1.1.4 Test Coverage

CRR Number	Test Case Number
CRRN1	2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
	<u>12, 13, 14, 15, 16, 17, 18, </u>
	<u>19, 20, 21, 22, 23, 24, 25, </u>
	<u>26</u>
CRRC1	<u>1</u>

6.3.1.2 ProactiveResponseHandler

Test Area Reference: FWK_MHA_PRHD-

6.3.1.2.1 Conformance Requirement

6.3.1.2.1.1 Normal Execution

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

EVENT_FORMATTED_SMS_PP_ENV

EVENT_UNFORMATTED_SMS_PP_ENV

EVENT_FORMATTED_SMS_PP_UPD

EVENT_UNFORMATTED_SMS_PP_UPD

EVENT_FORMATTED_SMS_CB

EVENT_UNFORMATTED_SMS_CB

EVENT_MENU_SELECTION

EVENT_MENU_SELECTION_HELP_REQUEST

EVENT_TIMER_EXPIRATION

EVENT_EVENT_DOWNLOAD_MT_CALL

EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

 ${\tt EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE}$

EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION

EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION_

EVENT UNRECOGNIZED ENVELOPE

EVENT_STATUS_COMMAND

EVENT_CALL_CONTROL

EVENT_SMS_MO_CONTROL

EVENT_PROFILE_DOWNLOAD

6.3.1.2.1.2 Parameters error

No requirements.

6.3.1.2.1.3 Context errors

No requirements. EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

EVENT EVENT DOWNLOAD CHANNEL STATUS

6.3.1.2.1.2 Context Errors

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

EVENT_FIRST_COMMAND_AFTER_SELECT

6.3.1.2.2 Test Suite Files

Test Script: FWK_MHA_PRHD_1.scr

Test Applet: FWK_MHA_PRHD_1.java

FWK_MHA_PRHD_2.java

Load Script: FWK_MHA_PRHD_1.ldr

Cleanup Script: FWK_MHA_PRHD_1.clr

Parameter File:

Test Script: FWK_MHA_PRHD_1.scr

Test Applet: FWK_MHA_PRHD_1.java

FWK_MHA_PRHD_2.java

Load Script: FWK_MHA_PRHD_1.ldr

Cleanup Script: FWK_MHA_PRHD_1.clr

Parameter File: FWK_MHA_PRHD_1.par

6.3.1.2.3 Test Procedure

	Id Description	API/Framework Expectation	on APDU Expectation
4	Applets registration to all events and Proactive Response Handler availability with EVENT_PROFILE_DOWNLOAD		
1	1 Applet1 is registered to all events defined in [7], applet2 is registered to all events defined in [7] except. EVENT_CALL_CONTROL_BY_SIM and EVENT_MO_SMS_CONTROL_BY_SIM. 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.		
÷ ÷	1 Terminal Profile command is sent to the SIM without the facility of SET_EVENT_LIST and POLL_INTERVAL, ,SET UP IDLE MODE TEXTand SET UP MENU.	1-Applet1 is triggered No exception is thrown	
:	Applet1 builds a proactive command DISPLAY TEXT. 2 ProactiveHandler.send() method is called	3- No exception is thrown	2- The proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
	3— ProactiveResponseHandler.getTheHandler() method is called Applet1 is deregistered to EVENT_PROFILE_DOWNLOAD	Applet1 finalizes	
	Applet2 builds a proactive command DISPLAY TEXT. 4 ProactiveHandler.send() method is called	4- Applet2 is triggered	5- The proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
	5— ProactiveResponseHandler.getTheHandler() method is called Applet1 is deregistered to EVENT_PROFILE_DOWNLOAD	6-No exception is thrown	
	Applets registration to all events and Proace Response Handler availability with EVENT PROFILE DOWNLOAD Applet1 is registered to all events defined in TS 43.019 [7] except EVENT_FIRST_COMMAND_AFTER_SELECT, Applet2 is registered to all events defined in TS 43.109—[7] except EVENT_FIRST_COMMAND_AFTER_SELECT, EVENT_CALL_CONTROL_BY_SIM and EVENT_MO_SMS_CONTROL_BY_SIM. Using the methods initMenuEntry() for EVENT_MENU_SELECTION, requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() f EVENT_TIMER_EXPIRATION and setEventList for the rest of the events.	For	

ld	Description	API/Framework Expectation	APDU Expectation
	1-Terminal Profile command is sent to the SIM without the facility of SET_EVENT_LIST, POLL_INTERVAL, SET UP IDLE MODE TEXT and SET UP MENU. 2- Applet1 builds a proactive command DISPLAY TEXT. 3- ProactiveHandler.send() method is called	1-Applet1 is triggered by EVENT_PROFILE_DOWNLOAD No exception is thrown	3- The proactive command
	4- ProactiveResponseHandler.getTheHandler() method is called Applet1 is deregistered to EVENT_PROFILE_DOWNLOAD	4- No exception is thrown	DISPLAY TEXT is fetched TERMINAL RESPONSE
	5- Applet2 builds a proactive command DISPLAY TEXT.	Applet1 finalizes	
	6- ProactiveHandler.send() method is called	Applet2 is triggered by EVENT_PROFILE_DOWNLOAD	
	7ProactiveResponseHandler.getTheHandler() method is called Applet1 is deregistered to EVENT_PROFILE_DOWNLOAD	7- No exception is thrown	6- The proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE

	_ [Id Description	API/Framework Expectation	on APDU Expectation
\prod	2	Proactive Response Handler availability with		
		EVENT_MENU_SELECTION_HELP_REQUEST		
1 -		2 Proactive Response Handler availability w	vith	
		EVENT_MENU_SELECTION_HELP_REQUE	ST	
		Perform SIM initialization with all the		
		facilities supported	=	
		1-Envelope menu selection with help request is sent to the SIM	1- Applet1 is triggered	
		Applet1 builds a proactive command DISETEXT	PLAY	
		<pre>2- ProactiveHandler.send() method is called</pre>		2- A proactive command
		cailed		DISPLAY TEXT is fetched
				TERMINAL RESPONSE
		3-		TERMINAL REGIONAL
		ProactiveResponseHandler.getTheHandler		
_	2	method is called	3- No exception is thrown	
	3	Proactive Response Handler availability with EVENT_MENU_SELECTION		
			1- Applet1 is triggered	
		SIM		
		Applet1 builds a proactive command DISPLAY		
		TEXT		
		2 ProactiveHandler.send() method is		2- A proactive command-
		called		DISPLAY TEXT is fetched
				TERMINAL RESPONSE
				I LAWIN AL TREOF ONOL
		3 ProactiveResponseHandler.getTheHandler() method is called	3- No exception is thrown	
			A - 1 - 14 G - 15	
			Applet1 finalizes	

ld	Description	API/Framework Expect	ation	APDU Expectation
4 Env SIM	elope menu selection is sent to the	I− Applet2 is triggered		
Apple TEXT	t2 builds a proactive command DISPLAY			
5 Pr	oactiveHandler.send() method is			proactive command- PLAY TEXT is fetched
6 Pro	activeResponseHandler.getTheHandler()		TER	MINAL RESPONSE
me Eno		S- No exception is thrown		
3	Proactive Response Handler availability wite EVENT_MENU_SELECTION	th_	•	
	-Envelope menu selection is sent to the	1- Applet1 is triggered		
	pplet1 builds a proactive command DISPI EXT	YAL		
l —	- ProactiveHandler.send() method is alled			2- A proactive comman DISPLAY TEXT is fetch
				TERMINAL RESPONS
	-ProactiveResponseHandler.getTheHandler ethod is called	3- No exception is thrown		

	Id Description	API/Framework Expectatio	n APDU Expectation
1	Proactive Response Handler availability with	AFI/I Tamework Expectatio	AF DO Expectation
+	EVENT FORMATTED SMS PP ENV		
	EVERTET ORMATTED_OMO_TT_ERV		
	1 Envelope dataDownLoad formatted is sent-	1- Applet1 is triggered	
	to the SIM	, Applet is tilggered	
	Applet builds a proactive command DISPLAY-TEXT		
	1581		
	2 ProactiveHandler.send() method is called		2- A proactive command
			DISPLAY TEXT is fetched
			TERMINAL RESPONSE
	3 ProactiveResponseHandler.getTheHandler() method is called	3- No exception is thrown	
	method is called	·	
		Applet1 finalizes	
	4 Envelope dataDownLoad formatted is sent		
	to the SIM	4- Applet2 is triggered	
	Puril at O. In. (114) and a superior of PTOPINY		
	Applet2 builds a proactive command DISPLAY TEXT		
	THAT		
	5 ProactiveHandler.send() method is called		5- A proactive command
			DISPLAY TEXT is fetched
			TERMINAL RESPONSE
	6 ProactiveResponseHandler.getTheHandler()		
		6- No exception is thrown	
	4 Proactive Response Handler availability w	<u>rith</u>	
	EVENT FORMATTED SMS PP ENV		
	1-Envelope dataDownLoad formatted is se	1- Applet1 is triggered	
	to the SIM		
	Applet builds a proactive command DISP	r.av	
	TEXT		
			2- A proactive command
	2-ProactiveHandler.send() method is cal	lled	DISPLAY TEXT is fetched
			TERMINAL RESPONSE
		a North Control of the Control	TERMINAL RESPONSE
	3-ProactiveResponseHandler.getTheHandle	3- No exception is thrown	
	method is called		
5	Proactive Response Handler availability with		
	EVENT UNFORMATTED SMS PP ENV		
	1 Envelope dataDownLoad unformatted is	1- Applet1 is triggered	
	sent to the SIM		
	Applet1 builds a proactive command DISPLAY		
	TEXT		
	2 ProactiveHandler.send() method is		2- A proactive command
	called		DISPLAY TEXT is fetched
			TERMINAL RESPONSE
	3 -		
	ProactiveResponseHandler.getTheHandler()		
	method is called	3- No exception is thrown	
		Applet1 finalizes	
	Applet2 builds a proactive command DISPLAY		
	TEXT		
		4- Applet2 is triggered	
	4 ProactiveHandler.send() method is		5- A proactive command

ld	Description		API/Framework Expectation	on פום	APDU Expectation
5—	activeResponseHandler.getTheHandler()				MINAL RESPONSE
inc c.	nou is curred	e N	o exception is thrown		
5	Proactive Response Handler availability w		o exception is trilown		
-	EVENT_UNFORMATTED_SMS_PP_ENV				
	1-Envelope dataDownLoad unformatted is	_	1- Applet1 is triggered		
	sent to the SIM				
	Applet1 builds a proactive command DISETEXT	PLAY			
	2- ProactiveHandler.send() method is				2- A proactive command DISPLAY TEXT is fetche
	called				
					TERMINAL RESPONSE
	3- ProactiveResponseHandler.getTheHandler(method is called	()	3- No exception is thrown		
			Applet1 finalizes		
			Applet2 is triggered		
	Applet 2 builds a properties gommand DIGI	OT 70.37			
	Applet2 builds a proactive command DISP $\overline{\text{TEXT}}$	PLAY			
	4- ProactiveHandler.send() method is				4- A proactive command
	called				DISPLAY TEXT is fetche
	5- ProactiveResponseHandler.getTheHandler(<u>)</u>	5- No exception is thrown		TERMINAL RESPONSE
	method is called				
Pr	reactive Response Handler availability with EVENT_UNFORMATTED_SMS_CB				
1 E	nvelope call broadcast unformatted is to the SIM	1- A	pplet1 is triggered		
App TEX	let1 builds a proactive command DISPLAY				
2 P:	roactiveHandler.send() method is called				proactive command PLAY TEXT is fetched
3 P	roactiveResponseHandler.getTheHandler()	3- N	o exception is thrown	TER	MINAL RESPONSE
met:	hod is called.	0 11	o oxooption to trirown	721	WINT IE REGI ONGE
A	let2 builds a proactive command DISPLAY	App	et1 finalizes		
TEX					
	ProactiveHandler.send() method is	4 - A	pplet2 is triggered	5 <u>-</u> A	proactive command
cal	led			_	PLAY TEXT is fetched
				TER	MINAL RESPONSE
5 Pro	activeResponseHandler.getTheHandler()				
	hod is called				
			and the same	1	
6	Proactive Response Handler availability w		o exception is thrown		

ld	Description	API/Framework Expectation	n APDU Expectation
	1-Envelope cell broadcast formatted is sent to the SIM	1- Applet1 is triggered	
	Applet1 builds a proactive command DISP:	LAY	
	2-ProactiveHandler.send() method is cal		2- A proactive command DISPLAY TEXT is fetched
	3-ProactiveResponseHandler.getTheHandlemethod is called.	3- No exception is thrown	TERMINAL RESPONSE
7 1	roactive Response Handler availability with EVENT_UNFORMATTED_SMS_CB		
1 E	Envelope call broadcast unformatted is to the SIM	1- Applet1 is triggered	
App TEX	eletl builds a proactive command DISPLAY-		
2- P	roactiveHandler.send() method is called		2- A proactive command- DISPLAY TEXT is fetched
	:hod is called.		TERMINAL RESPONSE
		Applet1 finalizes 4- Applet2 is triggered	
App TEX	plet2 builds a proactive command DISPLAY		
1 1	ProactiveHandler.send() method is led		5- A proactive command- DISPLAY TEXT is fetched
5	_		TERMINAL RESPONSE
Pro met	eactiveResponseHandler.getTheHandler()— chod is called	6-No exception is thrown	

	ld	Description	API/Framework Expectation	on APDU Expectation
	<u>7</u>	Proactive Response Handler availability w		
		EVENT UNFORMATTED SMS CB		
		1-Envelope call broadcast unformatted i	s 1- Applet1 is triggered	
		sent to the SIM		
		Applet1 builds a proactive command DISPI	LAY	
		TEXT		2- A proactive command
		2-ProactiveHandler.send() method is cal	led	DISPLAY TEXT is fetched
		3-ProactiveResponseHandler.getTheHandle method is called.	3- No exception is thrown	TERMINAL RESPONSE
		meeriod is carred.	Applet1 finalizes	
			Applet2 is triggered	
		Applet2 builds a proactive command DISP	LAY	
		TEXT		4- A proactive command
		4- ProactiveHandler.send() method is		DISPLAY TEXT is fetched
		<u>called</u>		
				TERMINAL RESPONSE
		5-	5- No exception is thrown	
		ProactiveResponseHandler.getTheHandler(method is called) o No exception is thrown	
8	Pre	eactive Response Handler availability with EVENT_TIMER_EXPIRATION		
		EVERT_TIMEN_EXT INATION		
		er id=1		
	I ET	nvelope Timer Expiration is sent to the	1- Applet1 is triggered	
	App]	let builds a proactive command DISPLAY		
	TEXT			
	2-Pr	oactiveHandler.send() method is called		2- A proactive command
				DISPLAY TEXT is fetched
				TERMINAL RESPONSE
				TERMINAL RESPONSE
	3 Pa	roactiveResponseHandler.getTheHandler()		
	meth	nod is called	3- No exception is thrown	
		,	3- NO exception is triown	
	m t		Applet1 finalizes	
		er id=2 elope Timer Expiration is sent to the		
	SIM	_		
	App]	let builds a proactive command DISPLAY	4- Applet2 is triggered	
	158	r I		
	4-Pr	oactiveHandler.send() method is called		
			5- No exception is thrown	
	5 Pa	roactiveResponseHandler.getTheHandler()		6. A propositive account and
	me t l	lou 19 Callea		6- A proactive command DISPLAY TEXT is fetched
				TERMINAL RESPONSE
ļ L		I Burnett Burn		
	<u>8</u>	Proactive Response Handler availability win	<u>ith</u>	
I I	<u> </u>	FACIAL TIMICIT TYLINATION	1	

I	ld	Description		API/Framework Expectation	n	APDU Expectation
	1	Fimer id=1 1-Envelope Timer Expiration is sent to SIM	the	1- Applet1 is triggered		
	Ī	Applet builds a proactive command DISPLATEXT 2-ProactiveHandler.send() method is call				2- A proactive command DISPLAY TEXT is fetched
		3-ProactiveResponseHandler.getTheHandler() method is called		3- No exception is thrown		TERMINAL RESPONSE
1		active Response Handler availability with EVENT_CALL_CONTROL_BY_SIM velope call control by sim is sent to-	1- ∧	pplet1 is triggered		
T !	EXT	et builds a proactive command DISPLAY				
2 -	-Pro	activeHandler.send() method is called			DIS	PLAY TEXT is fetched
3 .		pactiveResponseHandler.getTheHandler() od is called	3- N	o exception is thrown	I I E K	MINAL RESPONSE

Г	ld	Description	API/Framework Expectatio	n APDU Expectation
	9	Proactive Response Handler availability wi		P
		EVENT CALL CONTROL BY SIM	4 . A I	
		1-Envelope call control by sim is sent	1- Applet1 is triggered	
		the SIM		
		Applet builds a proactive command DISPLA	AY	
		TEXT		
		2-ProactiveHandler.send() method is call	od	2- A proactive command
		2 Floattivenanter. send() method is tall	led	DISPLAY TEXT is fetched
			3- No exception is thrown	TERMINAL RESPONSE
		3-ProactiveResponseHandler.getTheHandle	r()	TERMINAL RESPONSE
1		method is called		
,				
10		pactive Response Handler availability with O_SHORT_MESSAGE_CONTROL_BY_SIM		
	- IVI	U_SHUKT_WESSAGE_CONTRUL_BT_SHVI		
	1 E1		1- Applet1 is triggered	
	is 	sent to the SIM		
	An	let builds a proactive command DISPLAY		
	APP.			
	2 P :	roactiveHandler.send() method is called		2- A proactive command DISPLAY TEXT is fetched
				TERMINAL RESPONSE
	2 Dro	pactiveResponseHandler.getTheHandler() method is		
	calle		3- No exception is thrown	
	<u>10</u>	Proactive Response Handler availability wit	<u>h</u>	
		MO SHORT MESSAGE CONTROL BY SI	<u>M</u>	
		1-Envelope mo short message control by	1- Applet1 is triggered	
		is sent to the SIM		
		Applet builds a proactive command DISPLA	AY	
		TEXT		
		2-ProactiveHandler.send() method is call	led	2- A proactive command
				DISPLAY TEXT is fetched
		3-ProactiveResponseHandler.getTheHandle: method is called	3- No exception is thrown	TERMINAL RESPONSE
11		oactive Response Handler availability with	1	<u> </u>
		EVENT_EVENT_DOWNLOAD_MT_CALL		
	1 - 120	nvelope event download mt call is sent	1- Applet1 is triggered	
		the SIM	i Applet le tilggeled	
	App:	letl builds a proactive command DISPLAY		
	TEX	P		
	2 P	roactiveHandler.send() method is called		
				2- A proactive command
				DISPLAY TEXT is fetched
	3 P	roactiveResponseHandler.getTheHandler()		TERMINAL RESPONSE
	metl			
		Ę	3- No exception is thrown	
			Applet1 finalizes	
		7	Application and a second a second and a second a second and a second a second and a second and a second and a	

ld	Description	API/Framework Expecta	ation	APDU Expectation
Applet2 bui	lds a proactive command DISPLAY	4- Applet2 is triggered		
4 Proactiv	reHandler.send() method is-		DISI	proactive command PLAY TEXT is fetched
5— ProactiveRe	sponseHandler.getTheHandler()		TER	MINAL RESPONSE
		6- No exception is thrown		

ld	Description	API/Framework Expectation	APDU Expectation
<u>11</u>	Proactive Response Handler availability w	rith .	
	EVENT EVENT DOWNLOAD MT CALL		
	1-Envelope event download mt call is se	1- Applet1 is triggered	
	to the SIM		
	Applet1 builds a proactive command DISI	DIAV	
	TEXT		
	2-ProactiveHandler.send() method is cal	<u>lled</u>	2- A proactive command DISPLAY TEXT is fetched
	3-ProactiveResponseHandler.getTheHandle	2 No execution is through	DISPLAY TEXT IS fetched
	method is called.	5- No exception is thrown	TERMINAL RESPONSE
		Applet1 finalizes	TERMINAL RESIGNOE
		Applet2 is triggered	
	Applet2 builds a proactive command DISETEXT	PLAY	
	<u>IEXI</u>		
	4- ProactiveHandler.send() method is		4- A proactive command
	<u>called</u>		DISPLAY TEXT is fetched
	5-		TERMINAL RESPONSE
	ProactiveResponseHandler.getTheHandler	() No over-the in the	
	method is called	5- No exception is thrown	
	pactive Response Handler availability with		T
EVE	NT_EVENT_DOWNLOAD_CALL_CONNECT		
	ED		
1 50	nvelope event download call connected	1- Applet1 is triggered	
is s	sent to the SIM	The state of the s	
Appl	et hill de a procettive germand DTCDTAV		
Appl TEXT	let1 builds a proactive command DISPLAY- F		
TEXT	<u> </u>		
TEXT			- A proactive command
TEXT	<u> </u>		- A preactive command- DISPLAY TEXT is fetched
TEXT	<u> </u>	£	DISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr	roactiveHandler.send() method is called	£	
TEXT 2 Pr 3 Pr	roactiveHandler.send() method is called	I	DISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr	roactiveHandler.send() method is called	£	DISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr	roactiveHandler.send() method is called	I	DISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr	roactiveHandler.send() method is called	3-No exception is thrown	DISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr	roactiveHandler.send() method is called	I	DISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr	roactiveHandler.send() method is called	3-No exception is thrown	DISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr meth	reactiveHandler.send() method is called reactiveResponseHandler.getTheHandler() ned is called reactive called	3- No exception is thrown Applet1 finalizes	DISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr meth	reactiveHandler.send() method is called reactiveResponseHandler.getTheHandler() ned is called reactive called	3-No exception is thrown	DISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr meth	reactiveHandler.send() method is called reactiveResponseHandler.getTheHandler() ned is called reactive called	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered	PISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr meth	reactiveHandler.send() method is called reactiveResponseHandler.getTheHandler() ned is called let builds a preactive command DISPLAY responseHandler.send() method is	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered	DISPLAY TEXT is fetched
2 Pr 3 Pr meth	reactiveHandler.send() method is called reactiveResponseHandler.getTheHandler() ned is called let builds a preactive command DISPLAY responseHandler.send() method is	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered	ERMINAL RESPONSE - A proactive command
2 Pr 3 Pr meth	reactiveHandler.send() method is called reactiveResponseHandler.getTheHandler() ned is called let builds a preactive command DISPLAY responseHandler.send() method is	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered	ERMINAL RESPONSE - A proactive command
2 Pr 3 Pr meth	reactiveHandler.send() method is called reactiveResponseHandler.getTheHandler() ned is called let builds a preactive command DISPLAY responseHandler.send() method is	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered	FERMINAL RESPONSE A proactive command- DISPLAY TEXT is fetched
2 Pr 3 Pr meth Appl TEXT 4 I	reactiveHandler.send() method is called reactiveResponseHandler.getTheHandler() ned is called let builds a preactive command DISPLAY receiveHandler.send() method is led	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered	FERMINAL RESPONSE A proactive command- DISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr meth TEXT 4 F	reactiveHandler.send() method is called reactiveResponseHandler.getTheHandler() med is called let builds a preactive command DISPLAY reconstitute and the command DISPLAY reconstitute and the command bush and the command bush and the command bush are command bush as a command bush as a command bush are command bush as a command bu	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered	FERMINAL RESPONSE A proactive command- DISPLAY TEXT is fetched
2 Pr 3 Pr meth Appl TEXT 4 I	reactiveHandler.send() method is called reactiveResponseHandler.getTheHandler() ned is called let builds a preactive command DISPLAY receiveHandler.send() method is led	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered	FERMINAL RESPONSE A proactive command- DISPLAY TEXT is fetched
2 Pr 3 Pr meth Appl TEXT 4 I call	reactiveHandler.send() method is called reactiveResponseHandler.getTheHandler() ned is called reactive command DISPLAY representative Handler.send() method is called rectiveResponseHandler.getTheHandler() ned is called	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered 5- No exception is thrown	FERMINAL RESPONSE A proactive command- DISPLAY TEXT is fetched
2 Pr 3 Pr meth TEXT 4 F call	reactiveHandler.send() method is called reactiveResponseHandler.getTheHandler() ned is called reactive command DISPLAY representative Handler.send() method is called rectiveResponseHandler.getTheHandler() ned is called reactive Response Handler availability were	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered 5- No exception is thrown ith	FERMINAL RESPONSE A proactive command- DISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr 4 Pr call 5 Proceedings	reactiveHandler.send() method is called reactiveResponseHandler.getTheHandler() ned is called reactive command DISPLAY representative Handler.send() method is called rectiveResponseHandler.getTheHandler() ned is called	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered 5- No exception is thrown ith	FERMINAL RESPONSE A proactive command- DISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr 4 Pr call 5 Proceedings	ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveHandler.getTheHandler ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() Proactive Response Handler availability was proactive Response Res	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered 5- No exception is thrown ith	FERMINAL RESPONSE A proactive command- DISPLAY TEXT is fetched
2 Pr 3 Pr meth TEXT 4 F call	ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() Proactive Response Handler availability was selected.	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered 6- No exception is thrown itth ECT	FERMINAL RESPONSE A proactive command- DISPLAY TEXT is fetched
TEXT 2 Pr 3 Pr 4 Pr call 5 Proceedings	ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() Proactive Response Handler availability was been set to be set to	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered 6- No exception is thrown ith ECT	FERMINAL RESPONSE A proactive command- DISPLAY TEXT is fetched
2 Pr 3 Pr meth TEXT 4 F call	ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() Proactive Response Handler availability was selected.	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered 6- No exception is thrown ith ECT	FERMINAL RESPONSE A proactive command- DISPLAY TEXT is fetched
2 Pr 3 Pr meth TEXT 4 F call	ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() Proactive Response Handler availability was selected.	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered 6- No exception is thrown ith ECT 1- Applet1 is triggered	ERMINAL RESPONSE A proactive command- DISPLAY TEXT is fetched ERMINAL RESPONSE
2 Pr 3 Pr meth TEXT 4 F call	ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() ProactiveHandler.send() method is ProactiveResponseHandler.getTheHandler() ProactiveResponseHandler.getTheHandler() Proactive Response Handler availability we EVENT EVENT DOWNLOAD CALL CONNED 1-Envelope event download call connected is sent to the SIM	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered 6- No exception is thrown ith ECT 1- Applet1 is triggered	2- A proactive command ERMINAL RESPONSE ERMINAL RESPONSE 2- A proactive command ERMINAL RESPONSE
2 Pr 3 Pr meth TEXT 4 F call	ProactiveResponseHandler.getTheHandler() RecactiveResponseHandler.getTheHandler() RecactiveResponseHandler.getTh	3- No exception is thrown Applet1 finalizes 4- Applet2 is triggered 6- No exception is thrown ith ECT ad	ERMINAL RESPONSE A proactive command- DISPLAY TEXT is fetched ERMINAL RESPONSE

	ld	Description	API/Framework Expectation	
		3-ProactiveResponseHandler.getTheHandlermethod is called	3- No exception is thrown	TERMINAL RESPONSE
		Applet builds a proactive command DISPLATE	Applet1 finalizes Applet2 is triggered	
		4- ProactiveHandler.send() method is called		4- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE
13		5 ProactiveResponseHandler.getTheHandler() method is called	5- No exception is thrown	TERMINAL RESIGNOE
13		Pactive Response Handler availability with ENT_EVENT_DOWNLOAD_CALL_DISCONN ECTED		
	1 Er disc	evelope event download call connected is sent to the SIM	I- Applet1 is triggered	
	A pp] FX3T	etl builds a proactive command DISPLAY		
	2 Pr	coactiveHandler.send() method is called		2- A proactive command- DISPLAY TEXT is fetched
	2 Da	-oastiveResponseHandler.getTheHandler()	=	FERMINAL RESPONSE
		rod is called	3- No exception is thrown Applet1 finalizes	
	App]	let2 builds a proactive command DISPLAY	1- Applet2 is triggered	
	4 I call	ProactiveHandler.send() method is		
	5 —		Ę	5- A proactive command- DISPLAY TEXT is fetched
		activeResponseHandler.getTheHandler()— nod is called €	S- No exception is thrown	FERMINAL RESPONSE

ld	Description	API/Framework Expectation	n APDU Expectation
<u>13</u>	Proactive Response Handler availability with	<u>th</u>	
	EVENT EVENT DOWNLOAD CALL DISCO	<u>NN</u>	
	ECTED		
	1-Envelope event download call disconnected is sent to the SIM	1- Applet1 is triggered	
	disconnected is sent to the sim		
	Applet1 builds a proactive command DISPI	LAY	
	TEXT		
	2-ProactiveHandler.send() method is call	<u>led</u>	2- A proactive command
			DISPLAY TEXT is fetched
		3- No exception is thrown	TERMINAL RESPONSE
	3-ProactiveResponseHandler.getTheHandlermethod is called	<u>r()</u>	
	meeriod 15 curren	Applet1 finalizes	
		Applet2 is triggered	
	Applet2 builds a proactive command DISPI		
	TEXT		
	4- ProactiveHandler.send() method is		
	called		4- A proactive command
			DISPLAY TEXT is fetched
			TEDMINIAL DECORONICE
	5- ProactiveResponseHandler.getTheHandler()	5- No exception is thrown	TERMINAL RESPONSE
	method is called oactive Response Handler availability with	O 140 CACCPHOTTIS THOWN	
Appl TEXT	let1 builds a proactive command DISPLAY F roactiveHandler.send() method is called	1	2-A proactive command- DISPLAY TEXT is fotched
		3- No exception is thrown Applet1 finalizes	TERMINAL RESPONSE
	 	Applet i ilitalizes	
	4	1- Applet2 is triggered	
Appl	let2 builds a proactive command DISPLAY		
1 fi A	-		
4 I	ProactiveHandler.send() method is		5- A proactive command
call	led		5- A proactive command DISPLAY TEXT is fetched
_			TERMINAL RESPONSE
5 Pros	activeResponseHandler.getTheHandler()	6- No exception is thrown	
meth	nod is called		
14	Proactive Response Handler availability wite EVENT_EVENT_DOWNLOAD_LOCATION_S TUS		
	<u></u>		
	1-Envelope event download location statu	1- Applet1 is triggered	

	ld	Description	API/Framework Expectatio	n APDU Expectation
		is sent to the SIM		
		Applet1 builds a proactive command DISPI	LAY	
		TEXT		2-A proactive command
		2-ProactiveHandler.send() method is call	Led	DISPLAY TEXT is fetched
				TERMINAL RESPONSE
		3-ProactiveResponseHandler.getTheHandlermethod is called	3- No exception is thrown	
		meenod 15 Garren	Applet1 finalizes	
			Applet2 is triggered	
		Applet2 builds a proactive command DISPI	LAY	
		TEXT		
		4- ProactiveHandler.send() method is		4- A proactive command
		<u>called</u>		DISPLAY TEXT is fetched
		5_		TERMINAL RESPONSE
		ProactiveResponseHandler.getTheHandler()	5- No exception is thrown	
15	Dr	method is called pactive Response Handler availability with		
	EVE	ENT_EVENT_DOWNLOAD_USER_ACTIVITY		
	1		L. A. Clark to the control	
	sent	nvelope event download user activity is to the SIM	I - Applet1 is triggered	
1	App]	etl builds a proactive command DISPLAY		
1	rex:	<u> </u>		
<u> </u>	2 Pa	coactiveHandler.send() method is called		
				2-A proactive command DISPLAY TEXT is fetched
				BIOLEXT TEXT IS retoried
		roactiveResponseHandler.getTheHandler()	N. Ntion in the	TERMINAL RESPONSE
ŧ	neth	is called	3- No exception is thrown	
		<i>+</i>	Applet1 finalizes	
			1- Applet2 is triggered	
	App] FEXT	et2 builds a proactive command DISPLAY	55	
				5- A proactive command
	1 1 call	ProactiveHandler.send() method is		DISPLAY TEXT is fetched
				TERMINAL RESPONSE
	5 <u> </u>	activeResponseHandler.getTheHandler()		
		nod is called		
		le	6- No exception is thrown	
· L			oncophorno unown	

Description APDU Expectation APDU Expectation APDU Expectation	-	Description	API/Framework Expectation	APDU Expectation
EVENT EVENT DOWNLOAD USER ACTIVITY 1. Envelope event download user activity is sent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2-ProactiveReagonseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4. ProactiveReagonseHandler.getTheHandler() method is called Applet2 is inggered Applet2 is inggered Applet3 in inggered 4. A proactive command DISPLAY TEXT 4. ProactiveReagonseHandler.getTheHandler() method is called FroactiveReagonseHandler.getTheHandler() method is called Applet4 is triggered 4. Applet4 is triggered Applet4 finalizes 4. Applet4 finalizes 4. Applet4 finalizes 4. Applet4 finalizes 4. Applet4 finalizes 5. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 6. No exception is thrown Applet4 finalizes 4. Applet4 is triggered 5. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 6. No exception is thrown 5. No exception is thrown 6. No exception is thrown 6. Applet4 finalizes 4. Applet4 finalizes 4. Applet4 finalizes 4. Applet4 finalizes 5. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 6. No exception is thrown 6. No exception is thrown 6. No exception is thrown 6. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 6. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 6. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 6. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 6. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 6. A proactive command-DISPLAY TEXT is fetched TE			1	711 DO EXPEDIATION
Envelope event download user activity is sent to the SIM	<u></u>	EVENT EVENT DOWNLOAD USER ACTIVIT	<u></u>	
pent to the SIM Applet1 builds a proactive command DISPLAY TEXT 2. Proactive(andler.send() method is called 3. Proactive(seponse(landler.getThe(landler())) method is called Applet2 builds a proactive command DISPLAY TEXT 4. Proactive(landler.nend() method is colled TERMINAL RESPONSE Proactive(landler.nend() method is colled TERMINAL RESPONSE 5. Proactive(landler.nend() method is colled TERMINAL RESPONSE Proactive(landler.nend() method is colled TERMINAL RESPONSE 5. No exception is thrown Proactive (landler.nendler.getThe(landler()) method is called TERMINAL RESPONSE 1. Applet1 is triggered 2. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 3. No exception is thrown Applet2 builds a proactive command DISPLAY TEXT 3. Proactive(landler.nend() method is called TERMINAL RESPONSE 4. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 2. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 4. Applet1 finalizes 4. Applet1 finalizes 4. Applet2 is triggered 5. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 5. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 5. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 5. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 5. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 6. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 5. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 6. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 5. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 6. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 6. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 6. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 6. A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 6. A proactive command DISPLAY TEXT is fetched TEXT is fetched TEXT is fetched TERMINAL RESPONSE			_	
sent to the SIM Applett builds a proactive command DISPLAY 22A Proactive 2A Pro		1-Envelope event download user activity i	s 1- Applet1 is triggered	
2-A proactiveResponseRandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4 - ProactiveResponseRandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4 - ProactiveResponseRandler.getTheHandler() method is called FroactiveResponseRandler.getTheHandler() method is called Applet1 buildo a proactive command DISPLAY TEXT TEXT 3 - No exception is thrown FroactiveResponseRandler.getTheHandler() applet1 buildo a proactive command DISPLAY TEXT Applet2 buildo a proactive command DISPLAY TEXT Applet3 buildo a proactive command DISPLAY TEXT Applet4 buildo a proactive command DISPLAY TEXT Applet4 buildo a proactive command DISPLAY TEXT FroactiveResponseRandler.getTheHandler() Buildo ResponseRandler.getTheHandler() Guildo Respon		sent to the SIM		
2-A proactiveResponseRandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4 - ProactiveResponseRandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4 - ProactiveResponseRandler.getTheHandler() method is called FroactiveResponseRandler.getTheHandler() method is called Applet1 buildo a proactive command DISPLAY TEXT TEXT 3 - No exception is thrown FroactiveResponseRandler.getTheHandler() applet1 buildo a proactive command DISPLAY TEXT Applet2 buildo a proactive command DISPLAY TEXT Applet3 buildo a proactive command DISPLAY TEXT Applet4 buildo a proactive command DISPLAY TEXT Applet4 buildo a proactive command DISPLAY TEXT FroactiveResponseRandler.getTheHandler() Buildo ResponseRandler.getTheHandler() Guildo Respon		Applot1 builds a projective servered DICDIA		
2 ProactiveReaponseHandler.send() method is called 3-ProactiveReaponseHandler.getTheHandler() ethod is called Applet2 builds a proactive command DISPLAY 4- A proactiveReaponseHandler.send() method is called 5- ProactiveReaponseHandler.send() method is called 5- ProactiveReaponseHandler.getTheHandler() ethod is called FroactiveReaponseHandler.getTheHandler() ethod is called 5- No exception is thrown FroactiveReaponseHandler.getTheHandler() ethod is called 5- No exception is thrown FroactiveReaponseHandler.getTheHandler() ethod is called 1- Applet1 is triggered 2- A proactive command DISPLAY 8- Proactive ReaponseHandler.getTheHandler() ethod is called 1- Applet1 is triggered 2- A proactive command DISPLAY 8- ProactiveReaponseHandler.getTheHandler() ethod is called 2- A proactive command DISPLAY 8- ProactiveReaponseHandler.getTheHandler() ethod is called 1- Applet1 is triggered 4- Applet2 is triggered 5- A proactive command DISPLAY 5- A proactive command DISPLAY 8- ProactiveReaponseHandler.getTheHandler() ethod is called 1- Applet1 is triggered 5- A proactive command DISPLAY 5- A			<u> </u>	
DISPLAY TEXT is fetched TERMINAL RESPONSE Applet2 builds a proactive command DISPLAY TEXT 4. ProactiveResponseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4. ProactiveResponseHandler.getTheHandler() method is called FroactiveResponseHandler.getTheHandler() method is called Applet1 builds a proactive command DISPLAY TERMINAL RESPONSE 1. Applet1 builds a proactive command DISPLAY Applet2 builds a proactive command DISPLAY TERMINAL RESPONSE 2. A proactive command DISPLAY TERMINAL RESPONSE 3. No exception is thrown Applet1 finalizes 4. A proactive command DISPLAY TERMINAL RESPONSE 5. A proactive command DISPLAY TERMINAL RESPONSE 5. A proactive command DISPLAY TERMINAL RESPONSE 6. No exception is thrown 5. A proactive command DISPLAY TERMINAL RESPONSE 5. A proactive command DISPLAY TERMINAL RESPONSE 6. No exception is thrown 6. No exception is thrown 6. No exception is thrown 7. A proactive command DISPLAY TERMINAL RESPONSE 6. No exception is thrown 6. No exception is thrown 7. A proactive command DISPLAY TERMINAL RESPONSE 6. No exception is thrown 6. No exception is thrown 7. A proactive command DISPLAY TERMINAL RESPONSE 6. No exception is thrown 7. A proactive command DISPLAY TERMINAL RESPONSE 8. No exception is thrown 8. A proactive command DISPLAY 1. Applet1 is triggered 1. Applet1 is triggered 1. Applet2 is triggered 1. Applet3 is triggered 1. Applet4 is triggered				
3-ProactiveResponseHandler.getTheHandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4- ProactiveResponseHandler.setfleEandler() method is called ProactiveResponseHandler.getTheHandler() method is called Applet3 builds a proactive command DISPLAY maxw 2- A proactive ResponseHandler.getTheHandler() method is called Applet4 builds a proactive command DISPLAY maxw 4- ProactiveResponseHandler.getTheHandler() method is called Applet5 builds a proactive command DISPLAY method is called Applet6 builds a proactive command DISPLAY method is called Applet6 builds a proactive command DISPLAY method is called FroactiveResponseHandler.getTheHandler() method is called Applet6 builds a proactive command DISPLAY EXECUTED DISPLAY TEXT is fetched TERMINAL RESPONSE 5- A proactive command DISPLAY Applet6 builds a proactive command DISPLAY BroactiveResponseHandler.getTheHandler() method is called 1- Applet6 is triggered		2-ProactiveHandler.send() method is calle	ed ed	
3. No exception is thrown Applet Section				DISPLAY TEXT is fetche
3. Proactive command 1. Applet1 finalizes				TERMINIAL RECOGNICE
Applet2 builds a proactive command DISPLAY 4. ProactiveNeaponseNandler.getTheNandler() method is called ProactiveReaponseNandler.getTheNandler() method is called ProactiveReaponseNandler.getTheNandler() method is called ProactiveReaponseNandler.getTheNandler() method is called ProactiveReaponseNandler.getTheNandler() method is called ProactiveNorther builds a proactive command DISPLAY TEXT 3. ProactiveNandler.getTheNandler() method is called Applet1 builds a proactive command DISPLAY TEXT 3. ProactiveNandler.getTheNandler() method is called Applet1 finalizes 4. A proactiveNandler.getTheNandler() getTheNandler() Applet1 finalizes 4. Applet2 is triggered Applet2 is triggered Applet3 builds a proactive command DISPLAY TEXT 5. A proactiveNandler.getTheNandler() method is called FroactiveNandler.getTheNandler() S. No exception is thrown Applet1 finalizes 4. Applet2 is triggered Applet2 is triggered FroactiveNandler.getTheNandler() S. No exception is thrown 1. Applet1 finalizes 4. A proactive command DISPLAY TEXT 5. A proactive command DISPLAY TEXT 5. A proactive command DISPLAY TEXT 5. No exception is thrown 1. Applet1 is triggered 1. Applet1 is triggered 1. Applet1 is triggered 1. Applet1 is triggered		3-ProactiveResponseHandler.getTheHandler() 3- No exception is through	TERMINAL RESPONSE
Applet2 builds a proactive command DISPLAY TEXT 4. ProactiveNandler.send() method is called FroactiveResponseHandler.getTheNandler() method is called FroactiveResponseHandler.getTheNandler() method is called FroactiveResponseHandler.getTheNandler() method is called FroactiveResponseHandler.getTheNandler() FroactiveResponseHandler.getTheNandler() FroactiveResponseHandler.getTheNandler() FroactiveResponseHandler.getTheNandler() Applet1 is triggered Applet2 is triggered Applet3 in Institute thrown Applet4 finalizes 4. Applet4 finalizes 5. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 5. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 6. No exception is thrown 6. No exception is thrown 16. ProactiveResponseHandler.getTheNandler() TERMINAL RESPONSE 17. Applet4 is triggered 18. ProactiveResponseHandler.getTheNandler() TERMINAL RESPONSE 19. Applet4 is triggered 10. Applet4 is triggered 10. Applet4 is triggered 11. Applet4 is triggered 11. Applet4 is triggered 12. Applet4 is triggered 13. Applet4 is triggered 14. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 18. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 19. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 19. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 19. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 19. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 10. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 10. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 19. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 19. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 19. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 10. A proactive command-DISPLAY TEXT is fetched TERMINAL RESPONSE 19. A proactive comma			- 140 evcebriori is miromii	
Applet2 builds a proactive command DISPLAY TEXT 4 - ProactiveResponseHandler.send() method is called 5- ProactiveResponseHandler.getTheNandler() sethod is called FroactiveResponseHandler.getTheNandler() sethod is called 5- No exception is thrown FroactiveResponseHandler.getTheNandler() sethod is called 4- A proactive Response 5- No exception is thrown 5- No exception is thrown 4- A proactive ResponseHandler.getTheNandler() sethod is called 4- A proactive ResponseHandler.getTheNandler() sethod is called 4- A proactive ResponseHandler.getTheNandler() sethod is getTheNandler.getTheNandler() sethod is getTheNandler() sethod is			Applet1 finalizes	
Applet2 builds a proactive command DISPLAY ### ProactiveReaponseHandler.send() method is called				
## ProactiveResponseHandler.send() method is called ## ProactiveResponseHandler.getTheHandler() method is called ## ProactiveResponseHandler.getTheHandler() ## ProactiveResponseHandler.getTheHandler		Applet2 builds a proactive command DISPLA		
DISPLAY TEXT is fetched TERMINAL RESPONSE				
DISPLAY TEXT is fetched TERMINAL RESPONSE				4- A proactive command
FroactiveResponseHandler.getTheHandler() method is called ProactiveResponseHandler availability with EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_ AVAILABLE i_Envelope_cvent_download_idle_screen_ available_is_sent_to_the_SIM Applet1_builds_a_proactive_command_DISPLAY_ TEXT 2_ProactiveResponseHandler.getTheHandler() method is_called Applet2_builds_a_proactive_command_DISPLAY_ TEXT 4_ProactiveResponseHandler.getTheHandler() method is_called Applet2_builds_a_proactive_command_DISPLAY_ TEXT 4_ProactiveIndex command_DISPLAY_ TEXT 4_ProactiveIndex command_DISPLAY_ TEXT 5_A proactive command_DISPLAY_ TEXT 5_A proactive command_DISPLAY_ TEXT 5_A proactive command_DISPLAY_ TEXT 6_No exception is thrown 1_EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_ AVAILABLE 1_Envelope_event_download_idle_screen_ available_is_sent_to_the_SIM_ Applet1_builds_a_proactive_command_DISPLAY_ 1_Applet1 is triggered 1_Applet1 is triggered				DISPLAY TEXT is fetche
S- ProactiveResponseHandler.getTheHandler()		Called		
Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_IBLE_SCREEN_ AVAILABLE 1-Envelope_event_download_idle_excent available_is_ent_to_the_EIM Applet1 builds_a proactive_command_DISPLAY ##### 3 ProactiveResponseIIandler.getTheliandler() ####################################				TERMINAL RESPONSE
Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_IBLE_SCREEN_ AVAILABLE 1-Envelope—event_download_idle_oereen_ available_is_send() method_is_ealled 2-A proactive command_DISPLAY ###################################				
Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_IBLE_SCREEN_AVAILABLE 1-Envelope—event_download_idle_oereen_available_is_sent_to_the_SIM 2-A proactive Response Handler _getTheilandler() 3-No exception is thrown 2-A proactive command_DISPLAY 2-A proactive command_DISPLAY 2-A proactive command_DISPLAY 3-ProactiveResponseIlandler.getTheilandler() 3-No exception is thrown Applet1 finalizes 4-Applet2 builds_a proactive_command_DISPLAY 3-ProactiveResponseIlandler.getTheilandler() 3-No exception is thrown Applet1 finalizes 4-Applet2 builds_a proactive_command_DISPLAY 3-No exception is thrown Applet3 builds_a proactive_command_DISPLAY 3-No exception is thrown 4-ProactiveResponseIlandler.getTheilandler() 4-ProactiveRespo		<u>5-</u>		
Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_ AVAILABLE 1			E. No evecation is the second	
LENVELOPE - EVENT_DOWNLOAD_IDLE_SCREEN_ AVAILABLE I Envelope - event download idle sereen available is sent to the SIM Applet1 builds a proactive command_DISPLAY ###################################	D-		o- INO exception is thrown	-
AVAILABLE 1 Envelope event download idle screen available is aent to the SIM Applet! builds a proactive command DISPLAY TEXT 2 ProactiveResponsellandler.getThellandler() acalled Applet2 builds a proactive command DISPLAY TEXT Applet3 builds a proactive command DISPLAY TEXT Applet4 finalizes 4 Applet2 is triggered Applet2 is triggered 5 A proactive command DISPLAY TEXT TEXT 5 A proactive command DISPLAY TEXT is fotched TERMINAL RESPONSE 5 ProactiveResponsellandler.getThellandler() method is called 6 No exception is thrown 6 No exception is thrown 16 ProactiveResponsellandler.getThellandler() method is called 1 Applet1 is triggered 1 Applet1 is triggered 1 Applet1 is triggered	EV	ENT EVENT DOWNLOAD IDLE SCREEN		
DISPLAY TEXT is fetched TERMINAL RESPONSE 3 ProactiveResponsellandler.getThellandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4 ProactiveResponsellandler.getThellandler() method is called 5 A proactive command DISPLAY TEXT 5 A proactive command DISPLAY TEXT 5 A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 5 ProactiveResponsellandler.getThellandler() method is called 6 No exception is thrown 16 Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1 Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY				
DISPLAY TEXT is fetched TERMINAL RESPONSE 3 ProactiveResponsellandler.getThellandler() method is called Applet2 builds a proactive command DISPLAY TEXT 4 ProactiveResponsellandler.getThellandler() method is called 5 A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 5 ProactiveResponsellandler.getThellandler() method is called 6 No exception is thrown 6 No exception is thrown 16 Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1 Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY	2 P	reactiveHandler gend() method is called	2	- A proactive command
3 - No exception is thrown Applet2 builds a proactive command DISPLAY TEXT 4 - ProactiveResponsellandler.send() method is called 5 - A proactive command DISPLAY TEXT is fotched TERMINAL RESPONSE 5 - ProactiveResponsellandler.getThellandler() method is called 6 - No exception is thrown 16 Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1 - Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY	2	rougervenumater. Bena() meemoa ib carrea		
3 No exception is thrown Applet2 builds a proactive command DISPLAY TEXT 4 ProactiveResponsellandler.send() method is called 5 A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 5 No exception is thrown 16 Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1 - Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY				
Applet2 builds a proactive command DISPLAY TEXT 4 ProactiveResponseHandler.send() method is called 5- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 5- ProactiveResponseHandler.getTheHandler() method is called 6- No exception is thrown 16 Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY			-	EDMINIAL DECEDANCE
Applet2 builds a proactive command DISPLAY TEXT 4 ProactiveResponseHandler.send() method is called 5- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 5- ProactiveResponseHandler.getTheHandler() method is called 6- No exception is thrown 16 Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY				ERMINAL RESPONSE
Applet2 builds a proactive command DISPLAY TEXT 4 ProactiveHandler.send() method is called 5	3 Pa	oacervenesponsenanarer geernenanarer ()	No exception is thrown	ERMINAL RESPONSE
Applet2 builds a proactive command DISPLAY TEXT 4 ProactiveHandler.send() method is called 5		loa 19 callea	No exception is thrown	ERMINAL RESPONSE
Applet2 builds a proactive command DISPLAY TEXT 4 ProactiveHandler.send() method is called 5- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 5- ProactiveResponseHandler.getTheHandler() method is called 6- No exception is thrown 16 Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY		Hod IS CALLED	No exception is thrown	ERMINAL RESPONSE
Applet2 builds a proactive command DISPLAY TEXT 4 ProactiveHandler.send() method is called 5- A proactive command DISPLAY TEXT is fetched TERMINAL RESPONSE 5- ProactiveResponseHandler.getTheHandler() method is called 6- No exception is thrown 16 Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY			·	ERMINAL RESPONSE
4 ProactiveHandler.send() method is— called TERMINAL RESPONSE 5— ProactiveResponseHandler.getTheHandler() method is—called 16 Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY		A e	pplet1 finalizes	EKMINAL KESPONSE
4 ProactiveHandler.send() method is— called TERMINAL RESPONSE 5— ProactiveResponseHandler.getTheHandler() method is—called 16 Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY		A e	pplet1 finalizes	ERMINAL RESPONSE
DISPLAY TEXT is fetched tended TERMINAL RESPONSE TERMINAL RESPONSE No exception is thrown Terminal response Industrial re	App :	AF 4- let2 builds a proactive command DISPLAY	pplet1 finalizes	ERMINAL RESPONSE
TERMINAL RESPONSE Terminal Response From the Front is thrown 6 No exception is thrown	App:	AF 4- let2 builds a proactive command DISPLAY	oplet1 finalizes Applet2 is triggered	
ProactiveResponseHandler.getTheHandler() method is called 16 Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY	App: TEX:	Ap 4-let2 builds a proactive command DISPLAY	pplet1-finalizes Applet2 is triggered	i∼ A proactive command
Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY	4—1	Applet2 builds a proactive command DISPLAY ProactiveHandler.send() method is	oplet1 finalizes Applet2 is triggered €	i- A preactive command DISPLAY TEXT is fetched
Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY	4—1	Applet2 builds a proactive command DISPLAY ProactiveHandler.send() method is	oplet1 finalizes Applet2 is triggered €	i- A preactive command DISPLAY TEXT is fetched
Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY	4—1	Applet2 builds a proactive command DISPLAY ProactiveHandler.send() method is	oplet1 finalizes Applet2 is triggered €	i- A preactive command DISPLAY TEXT is fetched
1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY	4—1 cal :	Applet2 builds a proactive command DISPLAY ProactiveHandler.send() method is led	oplet1 finalizes Applet2 is triggered €	i- A preactive command DISPLAY TEXT is fetched
AVAILABLE 1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY	4 1 cal:	Applet2 builds a proactive command DISPLAY ProactiveHandler.send() method is led activeResponseHandler.getTheHandler() activeResponseHandler.getTheHandler()	Poplet1-finalizes Applet2 is triggered	i- A proactive command DISPLAY TEXT is fetched
1-Envelope event download idle screen available is sent to the SIM Applet1 builds a proactive command DISPLAY	4—1 cal 5— Prod metl	Applet2 builds a proactive command DISPLAY ProactiveHandler.send() method is led activeResponseHandler.getTheHandler() hod is called Proactive Response Handler availability with	Applet2 is triggered Respectively.	i- A proactive command DISPLAY TEXT is fetched
available is sent to the SIM Applet1 builds a proactive command DISPLAY	4—1 cal: 5— Proc	Applet2 builds a proactive command DISPLAY FroactiveHandler.send() method is led activeResponseHandler.getTheHandler() activeResponseHandler.getTheHandler() activeResponseHandler.getTheHandler() bood is called Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN	Applet2 is triggered Respectively.	i- A proactive command DISPLAY TEXT is fetched
available is sent to the SIM Applet1 builds a proactive command DISPLAY	4—1 cal : 5— Prod	Applet2 builds a proactive command DISPLAY FroactiveHandler.send() method is led activeResponseHandler.getTheHandler() activeResponseHandler.getTheHandler() activeResponseHandler.getTheHandler() bood is called Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN	Applet2 is triggered Respectively.	i- A proactive command DISPLAY TEXT is fetched
Applet1 builds a proactive command DISPLAY	4—1 cal	Applet2 builds a proactive command DISPLAY ProactiveHandler.send() method is led activeResponseHandler.getTheHandler() for a called Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE	Applet2 is triggered No exception is thrown	i- A proactive command DISPLAY TEXT is fetched
	4—1 cal	Applet2 builds a proactive command DISPLAY ProactiveHandler.send() method is led activeResponseHandler.getTheHandler() activeResponseHandler.getTheHandler() bood is called Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1-Envelope event download idle screen	Applet2 is triggered No exception is thrown	i- A proactive command DISPLAY TEXT is fetched
TEXT	4—1 cal 5— Prod metl	Applet2 builds a proactive command DISPLAY ProactiveHandler.send() method is led activeResponseHandler.getTheHandler() activeResponseHandler.getTheHandler() bood is called Proactive Response Handler availability with EVENT EVENT DOWNLOAD IDLE SCREEN AVAILABLE 1-Envelope event download idle screen	Applet2 is triggered No exception is thrown	i- A proactive command DISPLAY TEXT is fetched
	4—1 cal 5— Prod metl	Applet1 builds a proactive command DISPLAY ProactiveResponseHandler.getTheHandler() activeResponseHandler.getTheHandler() activeResponseHandler.g	Applet1 finalizes Applet2 is triggered No exception is thrown 1	i- A proactive command DISPLAY TEXT is fetched

ld	Description	API/Framework Expectation	APDU Expectation
	2-ProactiveHandler.send() method is called		2- A proactive command DISPLAY TEXT is fetched
	3-ProactiveResponseHandler.getTheHandler() method is called	3- No exception is thrown	TERMINAL RESPONSE
	Applet2 builds a proactive command DISPLAY TEXT	Applet1 finalizes Applet2 is triggered	
	4- ProactiveHandler.send() method is called		4- A proactive command DISPLAY TEXT is fetched
	5- ProactiveResponseHandler.getTheHandler()		TERMINAL RESPONSE
	method is called	5- No exception is thrown	

	Γ	Id Description	API/Framework Expectation	n APDU Expectation
	17	Proactive Response Handler availability with		
		EVENT_EVENT_DOWNLOAD_LANGUAGE_		
		17 Proactive Response Handler availability wit	<u> </u>	
		EVENT EVENT DOWNLOAD CARD READE _STATUS	<u>IR</u>	
		1-Envelope event download card reader status is sent to the SIM	1- Applet1 is triggered	
		Applet1 builds a proactive command DISPLATEXT	<u>Y</u>	
			- 4	2-A proactive command
		2-ProactiveHandler.send() method is call	<u>=u</u>	DISPLAY TEXT is fetched
				TERMINAL DESPONSE
		3-ProactiveResponseHandler.getTheHandler	() 3- No exception is thrown	TERMINAL RESPONSE
		method is called		
			Applet1 finalizes	
			Applet2 is triggered	
		2001-00-0-112		
		Applet2 builds a proactive command DISPL TEXT	AY	
		4- ProactiveHandler.send() method is called		4- A proactive command
				DISPLAY TEXT is fetched
		5		TERMINAL RESPONSE
		ProactiveResponseHandler.getTheHandler() method is called		
I	L		5- No exception is thrown	

	Id Description	API/Framework Expectation	APDU Expectation
18	Proactive Response Handler availability with EVENT_EVENT_DOWNLOAD_BROWSER_		
Į	EVENT_EVENT_DOWNLOAD_BROWSER_		
	18 Proactive Response Handler availability with	h	
	EVENT EVENT DOWNLOAD LANGUAGE SELECTION	-	
	1-Envelope event download language selection is sent to the SIM	1- Applet1 is triggered	
	Applet1 builds a proactive command DISPL	AY	
	TEXT		
	2-ProactiveHandler.send() method is call	ed	2-A proactive command DISPLAY TEXT is fetched
	3-ProactiveResponseHandler.getTheHandler		TERMINAL RESPONSE
	method is called	3-No exception is thrown	
		Applet1 finalizes	
	Applet2 builds a proactive command DISPI	Applet2 is triggered	
	TEXT		
	4- ProactiveHandler.send() method is called		
			4-A proactive command DISPLAY TEXT is fetched
	5	E No expention is thrown	
	ProactiveResponseHandler.getTheHandler() method is called	5-No exception is thrown	TERMINAL RESPONSE

[Id Description	API/Framework Expectation	n APDU Expectation
19	Proactive Response Handler availability with EVENT_STATUS_COMMAND		
!	EVENI_STATUS_COMMAND		
,	19 Proactive Response Handler availability w	ith	
	EVENT_EVENT_DOWNLOAD_BROWSER	<u>-</u>	
	TERMINATION		
	1-Envelope event download Browser termination is sent to the SIM	1- Applet1 is triggered	
	Applet1 builds a proactive command DISPI	- AV	
	TEXT	JAI _	
	2-ProactiveHandler.send() method is cal	led	2-A proactive command
			DISPLAY TEXT is fetched
	3-ProactiveResponseHandler.getTheHandle	3-No exception is thrown	TERMINAL RESPONSE
	method is called	5 NO CACOPHOLI IS HILOWII	
		Applet1 finalizes Applet2 is triggered	
	Applet2 builds a proactive command DISP		
	TEXT		
	4- ProactiveHandler.send() method is called		4-A proactive command
	5-		DISPLAY TEXT is fetched
	ProactiveResponseHandler.getTheHandler(method is called	5-No exception is thrown	TERMINAL RESPONSE
20	Proactive Response Handler availability with UNRECOGNIZED ENVELOPE		,
	_		
	1 An unrecognized Envelope is sent to the SIM	1- Applet1 is triggered	
	Applet1 builds a proactive command DISPLAY		
	APPICTI DULLUS A PROACTIVE COMMAND DISPLAY- TEXT		

ld	Description	API/Framework Expectation		APDU Expectation
2 Pr	roactiveHandler.send() method is called			proactive command- PLAY TEXT is fetched
			TER	WINAL RESPONSE
3 Pr	roactiveResponseHandler.getTheHandler()			
metł	ned is called	3- No exception is thrown		
		Applet1 finalizes		
App]		4- Applet2 is triggered		proactive command- LAY TEXT is fetched
4 - I	ProactiveHandler.send() method is		TERI	WINAL RESPONSE
call	Led			
_				
Proc	activeResponseHandler.getTheHandler()	6- No exception is thrown		
20	Proactive Response Handler availability w EVENT_STATUS_COMMAND			
		4. A		
	1-Status command is sent to the SIM	1- Applet1 is triggered		
	Applet1 builds a proactive command DIST	PLAY		
	2-ProactiveHandler.send() method is cal	lled		2- A proactive command
				DISPLAY TEXT is fetche
	3-ProactiveResponseHandler.getTheHandle	er() 3- No exception is thrown		TERMINAL RESPONSE
	meeriod 15 dand	Applet1 finalizes		
	Applet2 builds a proactive command DIS	Applet2 is triggered		
	TEXT			
	4- ProactiveHandler.send() method is called			4- A proactive command DISPLAY TEXT is fetche
				TERMINAL RESPONSE
	5- ProactiveResponseHandler.getTheHandler	5- No exception is thrown		
<u>21</u>	method is called Proactive Handler availability with			
	EVENT EVENT DOWNLOAD DATA AVAIL	LAB		
	1- Applet1 builds a proactive command (OPEN 1- Applet1 is registered to		1- OPEN CHANNEL
	<pre>CHANNEL. proactiveHandler.send() method is calle</pre>	ed TA_AVAILABLE and		proactive command is fetched
		EVENT EVENT DOWNLOAD ANNEL_STATUS		TERMINAL RESPONSE
	2- An Envelope Event Download Data	2- Applet1 is triggered		ssued with Channel Id =
	In add able to make the the GIM date			
	Available is sent to the SIM, with ChannelId=01.			
	ChannelId=01. 3-Applet1 builds a proactive command			
	ChannelId=01.			4- A proactive command

ld	Description	API/Framework Expectation	APDU Expectation
	5- ProactiveResponseHandler.getTheHandler() method is called	5- No exception is thrown	TERMINAL RESPONSE
22	Proactive Handler availability with EVENT EVENT DOWNLOAD CHANNEL STAT US	1- Applet1 is triggered	
	1-An Envelope Event Download Channel Status is sent to the SIM with ChannelId=01.		
	Applet1 builds a proactive command DISPLAY TEXT		
	2- ProactiveHandler.send() method is called		2- A proactive command DISPLAY TEXT is fetched
	ProactiveResponseHandler.getTheHandler() method is called	3- No exception is thrown	TERMINAL RESPONSE

ld	Description	API/Framework Expectation	APDU Expectation
<u>23</u>	Proactive Response Handler availability with		
	UNRECOGNIZED ENVELOPE		
	1-An unrecognized Envelope is sent to the SIM	1- Applet1 is triggered	
	Applet1 builds a proactive command DISPLAY TEXT		
	2-ProactiveHandler.send() method is called		2- A proactive command
	3-ProactiveResponseHandler.getTheHandler() method is called	3- No exception is thrown Applet1 finalizes Applet2 is triggered	DISPLAY TEXT is fetched TERMINAL RESPONSE
	Applet2 builds a proactive command DISPLAY <u>TEXT</u>		
	4- ProactiveHandler.send() method is called		4- A proactive command DISPLAY TEXT is fetched
	5 ProactiveResponseHandler.getTheHandler() method is called	5- No exception is thrown	TERMINAL RESPONSE
<u>24</u>	Proactive Response Handler availability with EVENT FORMATTED SMS PP UPD		
	1- Update Record EFsms instruction formatted is sent to the SIM	1- Applet1 is triggered	
	Applet builds a proactive command DISPLAY TEXT		2- A proactive command
	2-ProactiveHandler.send() method is called		DISPLAY TEXT is fetched
	3-ProactiveResponseHandler.getTheHandler() method is called	3- No exception is thrown	TERMINAL RESPONSE
<u>25</u>	Proactive Response Handler availability with EVENT_UNFORMATTED_SMS_PP_UPD		
	1- Update Record EFsms instruction unformatted is sent to the SIM	1- Applet1 is triggered	
	Applet1 builds a proactive command DISPLAY TEXT		
	2- ProactiveHandler.send() method is called		2- A proactive command DISPLAY TEXT is fetched
	3- ProactiveResponseHandler.getTheHandler() method is called	3- No exception is thrown	TERMINAL RESPONSE
	Applet2 builds a proactive command DISPLAY	Applet1 finalizes Applet2 is triggered	
	TEXT		
	4- ProactiveHandler.send() method is called		4- A proactive command DISPLAY TEXT is fetched
	<u>5-</u>		TERMINAL RESPONSE

ld	Description	API/Framework Expectation	APDU Expectation
		5- No exception is thrown	
	method is called		

6.3.1.2.4 Test Coverage

	CRR I	Number	Test Case Number	
CRRN	 1	1, 2, 3, 4, 5, 6,	7, 8, 9, 10, 11, 12, 13, 14, 15, 16,	17, 18, 19, 20
	CR	RRN1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25	
	CR	RC1	Not testable	

6.3.1.3 EnvelopeHandler

Test Area Reference: FWK_MHA_ENHD-

6.3.1.3.1 Conformance Requirement

6.3.1.3.1.1 Normal Execution

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

EVENT_FORMATTED_SMS_PP_ENV

EVENT_UNFORMATTED_SMS_PP_ENV

EVENT_FORMATTED_SMS_PP_UPD

EVENT_UNFORMATTED_SMS_PP_UPD

EVENT_FORMATTED_SMS_CB

EVENT_UNFORMATTED_SMS_CB

EVENT_MENU_SELECTION

EVENT_MENU_SELECTION_HELP_REQUEST

EVENT_TIMER_EXPIRATION

EVENT_EVENT_DOWNLOAD_MT_CALL

EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

 ${\tt EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE}$

EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION

EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION

EVENT_UNRECOGNIZED_ENVELOPE

EVENT_CALL_CONTROL

EVENT_SMS_MO_CONTROL

6.3.1.3.1.2 Parameters error

No requirements. EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS

6.3.1.3.1.2 Context Errors

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

EVENT_STATUS_COMMAND

EVENT_PROFILE_DOWNLOAD EVENT_STATUS_COMMAND

EVENT_PROFILE_DOWNLOAD

EVENT_FIRST_COMMAND_AFTER_SELECT

6.3.1.3.2 Test Suite Files

Test Script: FWK_MHA_ENHD_1.scr

Fest Applet: FWK_MHA_ENHD_1.java

FWK_MHA_ENHD_2.java

Load Script: FWK_MHA_ENHD_1.ldr

Cleanup Script: FWK_MHA_ENHD_1.clr

Parameter File: FWK_MHA_ENHD_1.par

6.3.1.3.3 Test Procedure

Test Script: FWK_MHA_ENHD_1.scr

Test Applet: FWK_MHA_ENHD_1.java

FWK_MHA_ENHD_2.java

Load Script: FWK_MHA_ENHD_1.ldr

Cleanup Script: FWK_MHA_ENHD_1.clr

Parameter File: FWK_MHA_ENHD_1.par

6.3.1.3.3 Test Procedure

		d Description	API/Framework Expectatio	n APDU Expectation	
4	Δ,	Applet1 and Applet2 registration and Envelope Handler availability with EVENT_PROFILE_DOWNLOAD			
	1	- Applet1 is registered to all events			
	d.	efined [7].			
	U.	sing the methods initMenuEntry() for VENT MENU SELECTION.			
	r (equestPollInterval() for			
	E.	VENT_STATUS_COMMAND, allocateTimer() for VENT_TIMER_EXPIRATION and setEventList()			
	£	or the rest of the events.			
	Aj	pplet2 is registered to all events- efined [7] except-			
	E.	VENT_CALL_CONTROL_BY_SIM and			
	U	VENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM. sing the methods initMenuEntry() for			
	₩.	VENT_MENU_SELECTION, equestPollInterval() for			
	판		1- No exception is thrown		
	£	VENT_TIMER_EXPIRATION and setEventList or the rest of the events.	The exception is the wife		
	7	Terminal Profile command is sent to SIM- ithout the facility of SET_EVENT_LIST- SETUP_IDLE_MODE_TEXT , POLL_INTERVAL and ETUP MENU	2- Applet1 is triggered		
		s called by Appletl ppletl is deregistered to	3- A Toolkit exception- HANDLER_NOT_AVAILABLE is- thrown		
			4- Applet2 is triggered		
	A	EnvelopeHandler.getTheHandler() method- s-called by Applet2 pplet2 is deregistered to- VENT_PROFILE_DOWNLOAD	5- A Toolkit exception- HANDLER_NOT_AVAILABLE is thrown		

	ld	Description	API/Framework Expectation	n APDU Expectation
2		Envelope Handler availability with		
	EV	ENT_MENU_SELECTION_HELP_REQUEST		

[ld	Description	API/Framework Expectation	APDU Expectation
	1	Applet1 and Applet2 registration and Envelor Handler availability with EVENT FIRST COMMAND AFTER SELECT		
		1.Applet1 is registered to all events defined TS 43.019 [7]. The registration is done using the methor initMenuEntry() for EVENT_MENU_SELECTION requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer() for EVENT_TIMER_EXPIRATION and setEventList for the rest of the events.	n,	
		Applet2 is registered to all events defined TS 43.019 [7] except EVENT_PROFILE_DOWNLOAD, EVENT_CALL_CONTROL_BY_SIM and EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM. The registration is done using the methor initMenuEntry() for EVENT_MENU_SELECTION requestPollInterval() for EVENT_STATUS_COMMAND, allocateTimer for EVENT_TIMER_EXPIRATION and setEventList for the rest of the events.	2- Applet1 is triggered by EVENT_FIRST_COMMAND_AFTE R SELECT	
		2- Select MF. 3-EnvelopeHandler.getTheHandler() method is called by Applet1 Applet1 is deregistered from EVENT_FIRST_COMMAND_AFTER_SELECT. 4-EnvelopeHandler.getTheHandler() method is called by Applet2 Applet2 is deregistered to	thrown Applet1 finalizes Applet2 in triggered	
-		EVENT_FIRST_COMMAND_AFTER_SELECT.	4- A Toolkit exception HANDLER NOT AVAILABLE is thrown	
	2	Handler availability with EVENT_PROFILE_DOWNLOAD 1- Terminal Profile command is sent to selected without the facility of SET_EVENT_LIST, SETUP_IDLE_MODE_TEXT, POLL_INTERVAL and SETUP MENU	1- Applet1 is triggered by EVENT PROFILE DOWNLOAD 2- A Toolkit exception	
		2- EnvelopeHandler.getTheHandler() methors called by Applet1 Applet1 is deregistered to EVENT_PROFILE_DOWNLOAD	HANDLER NOT AVAILABLE is	
		3-EnvelopeHandler.getTheHandler() method is called by Applet2 Applet2 is deregistered to EVENT_PROFILE_DOWNLOAD	Applet2 is triggered by EVENT_PROFILE_DOWNLOAD 3- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
3	1 Ex	Envelope Handler availability with EVENT_MENU_SELECTION nvelope menu selection is sent to the	1- Applet1 is triggered	
	2 Er	nvelopeHandler.getTheHandler() method galled by Applet1	2- No exception is thrown. Applet1 finalizes	
	3 Er SIM	nvelope menu selection is sent to the	3- Applet2 is triggered	

	ld	Description		API/Framework Expectation	n	APDU Expectation	
		nvelopeHandler.getTheHandler() method- called by Applet2	4 - N	o exception is thrown.			
	15 (carrea by Apprecia					
	3	Envelope Handler availability with					
		EVENT MENU SELECTION HELP REQUE	<u>EST</u>				
		Perform SIM initialization with all the	2				
		facilities supported	<u> </u>				
		Envelope menu selection with help reque	est	1- Applet1 is triggered			
		is sent to the SIM		1- Applett is triggered			
				O. No consention to the con-			
		1-EnvelopeHandler.getTheHandler() method	bc	2- No exception is thrown.			
		is called by Applet1					
4		Envelope Handler availability with					
		EVENT_FORMATTED_SMS_PP_ENV					
	1 A	EVENT_FORMATTED_SMS_PP_ENV envelope is	1- A	pplet1 is triggered			
	sent	to the SIM		Fb			
			2 N	o exception is thrown.			
	2 Er	nvelopeHandler.getTheHandler() method- called by Applet1	2-14	о ехсернон із інгомн.			
	10 0	Salica Sy Appleed	Appl	let1 finalizes			
	3 A	EVENT FORMATTED SMS PP ENV envelope is	3_ A	pplet2 is triggered			
	sent	to the SIM	0 / 1	ppiote to triggorou			
	4 Er	nvelopeHandler.getTheHandler() method	4- N	o exception is thrown.			
	4	Envelope Handler availability with					
	_	EVENT MENU SELECTION					
		1-Envelope menu selection is sent to the	he	1- Applet1 is triggered			
		SIM		1- Applett is triggered			
		2-EnvelopeHandler.getTheHandler() method	od	2- No exception is thrown.			
		is called by Applet1					

	lo		API/Framework Expectation	APDU Expectation
5		Envelope Handler availability with		
I		EVENT_UNFORMATTED_SMS_PP_ENV		
	5	Envelope Handler availability with		
		EVENT_FORMATTED_SMS_PP_ENV		
		1-A EVENT_FORMATTED_SMS_PP_ENV envelope sent to the SIM	1- Applet1 is triggered	
		Selft to the SIM		
		2-EnvelopeHandler.getTheHandler() method	2- No exception is thrown.	
		is called by Applet1		
	6	Envelope Handler availability with EVENT UNFORMATTED SMS PP ENV		
		1-An unformatted sms pp envelope is ser to the SIM	1- Applet1 is triggered	
		2-EnvelopeHandler.getTheHandler() method	od O No overstien is the same	
		is called by Applet1	2- No exception is thrown.	
			Applet1 finalizes	
			3- Applet2 is triggered	
		3-EnvelopeHandler.getTheHandler() metho	od	
		is called by Applet2		
_		Freedom Hondler evellebility with	4- No exception is thrown.	
6		Envelope Handler availability with EVENT_FORMATTED_CB		
	1	To allow well board out formatted to		
		Envelope cell broadcast formatted is ont to the SIM	1- Applet1 is triggered	
		EnvelopeHandler.getTheHandler() method	2-No exception is thrown	
	18	- called by Applet1	·	
			Applet1 finalizes	
	2	Envolope gell breedgest formatted /		
	3	<u>Envelope cell broadcast formatted is not to the SIM</u>	3- Applet2 is triggered	
		EnvelopeHandler.getTheHandler() method	4-No exception is thrown	
I <u>L</u>	is	called by Applet2	4-140 exception is tritown	

	ld	Description		API/Framework Expectation	APDU Expectation
	7	Envelope Handler availability with EVENT FORMATTED CB			
		1-Envelope cell broadcast formatted is sent to the SIM	_	1- Applet1 is triggered	
		2-EnvelopeHandler.getTheHandler() method is called by Applet1	od_	2-No exception is thrown	
7		Envelope Handler availability with EVENT_UNFORMATTED_CB			
	1 E	nvelope cell broadcast unformatted is t to the SIM	1- A	pplet1 is triggered	
	2 En	nvelopeHandler.getTheHandler() method called by Applet1	2- N	o exception is thrown	
	3 E	nvelopeHandler.getTheHandler() method- called by Applet2		let1-finalizes pplet2 is triggered	
			4- N	o exception is thrown	
	8	Envelope Handler availability with EVENT UNFORMATTED CB			
		1-Envelope cell broadcast unformatted sent to the SIM	is_	1- Applet1 is triggered	
		2-EnvelopeHandler.getTheHandler() methods called by Applet1	od_	2- No exception is thrown	
		3-EnvelopeHandler.getTheHandler() methods called by Applet2	od_	Applet1 finalizes 3- Applet2 is triggered	
				4- No exception is thrown	

	ld	Description	API/Framework Expectation	APDU Expectation
8		Envelope Handler availability with EVENT_TIMER_EXPIRATION		
9		Envelope Handler availability with		
		EVENT_CALL_CONTROL_BY_SIM		
		nvelope call control by sim is sent to— SIM	1- Applet1 is triggered	
	2 - 120	nvoloneWandlor gotThoWandlor() mothod		
	is (called by Applet1	2- No exception is thrown.	
	9	Envelope Handler availability with EVENT TIMER EXPIRATION		
		Timer id=1		
		1-Envelope Timer Expiration is sent to SIM	1- Applet1 is triggered	
		2-EnvelopeHandler.getTheHandler() method	2- No exception is thrown.	
	<u>10</u>	is called by Applet1 Envelope Handler availability with		
		EVENT CALL CONTROL BY SIM 1-Envelope call control by sim is sent	to 4 Applet4 is triggered	
		the SIM	1- Applet1 is triggered	
		2-EnvelopeHandler.getTheHandler() method	2- No exception is thrown.	
10		is called by Applet1 Envelope Handler availability with	2- No exception is thrown.	
10	EV	ENT_MO_SHORT_MESSAGE_CONTROL_B Y_SIM		
	1 E	1_0mm		
	is (sent to the SIM	1- Applet1 is triggered	
	2 E1	nvelopeHandler.getTheHandler() method		
	18 (called by Applet1.	2- No exception is throw	

	ld	Description		API/Framework Expectation	APDU Expectation
	<u>11</u>	Envelope Handler availability with EVENT MO SHORT MESSAGE CONTROL Y SIM	<u> B</u>		
		1-Envelope mo short message control by is sent to the SIM	sim	1- Applet1 is triggered	
		2-EnvelopeHandler.getTheHandler() methodis called by Applet1.	od_	2- No exception is throw	
11		Envelope Handler availability with- EVENT_EVENT_DOWNLOAD_MT_CALL			
	1 Er to t	nvelope event download mt call is sent-	1- A	pplet1 is triggered	
	2 Er	nvelopeHandler.getTheHandler() method- called by Applet1	2- N	o exception is thrown.	
	3 Er	nvelopeHandler.getTheHandler() method-		et1 finalizes pplet2 is triggered	
			4 - N	o exception is thrown.	
	<u>12</u>	Envelope Handler availability with EVENT EVENT DOWNLOAD MT CALL			
		1-Envelope event download mt call is se to the SIM	ent_	1- Applet1 is triggered	
		2-EnvelopeHandler.getTheHandler() methodis called by Applet1		2- No exception is thrown. Applet1 finalizes	
		3-EnvelopeHandler.getTheHandler() methodis called by Applet2		3- Applet2 is triggered	
				4- No exception is thrown.	

. —	Id Description	API/Framework Expectation APDU Expectation
12	Envelope Handler availability with EVENT_EVENT_DOWNLOAD_CALL_CONNECT	
	ED	
ı	13 Envelope Handler availability with	
	EVENT_EVENT_DOWNLOAD_CALL_CONN ED	<u>IECT</u>
	1-Envelope event download call connecte	
	is sent to the SIM	1- Applet1 is triggered
	<pre>2-EnvelopeHandler.getTheHandler() methor is called by Applet1</pre>	2- No exception is thrown.
	3-EnvelopeHandler.getTheHandler() metho	Applet1 finalizes
	is called by Applet2	3- Applet2 is triggered
13	Envelope Handler availability with	4- No exception is thrown.
	EVENT_EVENT_DOWNLOAD_CALL_DISCONE	
	CTTED	
	1 Envelope event download call disconnected is sent to the SIM	1- Applet1 is triggered.
	2 EnvelopeHandler.getTheHandler() methodis called by Applet1	2- No exception is thrown.
		Applet1 finalizes
	2 EnvelopeHandler getTheHandler() method	''
	is called by Applet2	3- Applet2 is triggered
	14 Envelope Handler availability with	4- No exception is thrown.
	EVENT EVENT DOWNLOAD CALL DISCO	<u>ONE</u>
	CTTED	
	1-Envelope event download call disconnected is sent to the SIM	1- Applet1 is triggered.
	a25001110000a 25 50115 50 5115 521.	
	2-EnvelopeHandler.getTheHandler() method	2- No exception is thrown.
	is called by Applet1	
	3-EnvelopeHandler.getTheHandler() metho	Applet1 finalizes
	is called by Applet2	3- Applet2 is triggered
<u> </u>		4- No exception is thrown.
14	Envelope Handler availability with EVENT_EVENT_DOWNLOAD_LOCATION_STA	
	TUS	
	1 Envelope event download location status	
	is sent to the SIM	1- Applet1 is triggered

152

[ld	Description	API/Framework Expectation APDU Expectation
	2 Er	nvelopeHandler.getTheHandler() method- called by Applet1	2- No exception is thrown.
			Applet1 finalizes
	2 17	nvelopeHandler.getTheHandler() method	3- Applet2 is triggered
		called by Applet2	4- No exception is thrown.
	<u>15</u>	Envelope Handler availability with	
		EVENT EVENT DOWNLOAD LOCATION :	<u>STA</u>
		1-Envelope event download location statis sent to the SIM	1- Applet1 is triggered
		2-EnvelopeHandler.getTheHandler() method is called by Applet1	2- No exception is thrown.
			Applet1 finalizes
		3-EnvelopeHandler.getTheHandler() method is called by Applet2	3- Applet2 is triggered
			4- No exception is thrown.
15	EVE	Envelope Handler availability with ENT EVENT DOWNLOAD USER ACTIVITY	
		nvelope event download user activity is	1- Applet1 is triggered
	sent	to the SIM	1- Applet is triggered
	2 Er	nvelopeHandler.getTheHandler() methodealled by Applet1	2- No exception is thrown
			Applet1 finalizes
			3- Applet2 is triggered
	3 Er	welopeHandler.getTheHandler() method called by Applet2	4- No exception is thrown
	<u>16</u>	Envelope Handler availability with	
		EVENT EVENT DOWNLOAD USER ACTIV	
		1-Envelope event download user activity sent to the $\underline{\text{SIM}}$	1- Applet1 is triggered
		2-EnvelopeHandler.getTheHandler() method is called by Applet1	2- No exception is thrown
			Applet1 finalizes
		3-EnvelopeHandler.getTheHandler() method is called by Applet2	3- Applet2 is triggered
		* **	4- No exception is thrown

	ld	Description		API/Framework Expectation	n	APDU Expectation
16		Envelope Handler availability with		-		-
	₽¥	ENT_EVENT_DOWNLOAD_IDLE_SCREEN_ AVAILABLE				
1		, , , , , , , , , , , , , , , , , , ,				
, L				T		
	<u>17</u>	Envelope Handler availability with EVENT EVENT DOWNLOAD IDLE SCREI	=N			
		AVAILABLE				
		1-Envelope event download idle screen				
		available is sent to the SIM		1- Applet1 is triggered		
		2-EnvelopeHandler.getTheHandler() methors called by Applet1	od_			
		The Carried By Apprecia		2- No exception is thrown.		
				Applet1 finalizes		
		3-EnvelopeHandler.getTheHandler() methors called by Applet2	od_			
		The Carried By Apprecia		3- Applet2 is triggered		
17		Envelope Handler availability with		4- No exception is thrown.		
	€V	ENT_EVENT_DOWNLOAD_CARD_READER				
		_STATUS				
	1-E	nvelope event download card reader	1- A	pplet1 is triggered		
	sca					
		nvelopeHandler.getTheHandler() method called by Applet1	2- N	o exception is thrown.		
			Ann	let1 finalizes		
		nvelopeHandler.getTheHandler() method	3_ A	pplet2 is triggered		
	is	called by Applet2	,	FF. S.E. IO MISSONOG		
			4- N	o exception is thrown.		
	<u>18</u>	Envelope Handler availability with		Cocopilorris tillown.		
		EVENT EVENT DOWNLOAD CARD REAL STATUS	<u>DER</u>			
		1-Envelope event download card reader status is sent to the SIM		1- Applet1 is triggered		
		2-EnvelopeHandler.getTheHandler() method	nđ			
		is called by Applet1	<u> </u>	2- No exception is thrown.		
				Applet1 finalizes		
		3-EnvelopeHandler.getTheHandler() metho	nđ			
		is called by Applet2	<u> </u>	3- Applet2 is triggered		
				A. Nie same die ete d		
18		Envelope Handler availability with		4- No exception is thrown.		
'`	ΙĘ	VENT EVENT DOWNLOAD LANGUAGE				

Г	ld	Description		API/Framework Expectation	n T	APDU Expectation
		SELECTION		- S g. Came on Exposition	<u>- 1</u>	in 20 Exposition
		evelope event download language exetion is sent to the SIM	1- A	pplet1 is triggered		
		welopeHandler.getTheHandler() method- called by Applet1	2-N (exception is thrown.		
	<u>2</u>	welopeHandler.getTheHandler() method	Арр	let1 finalizes		
	is c	valled by Applet2	3- A	pplet2 is triggered		
			4- N	o exception is thrown.		
	<u>19</u>	Envelope Handler availability with EVENT_EVENT_DOWNLOAD_LANGUAG SELECTION		o exception is unewn.		
		1-Envelope event download language selection is sent to the SIM		1- Applet1 is triggered		
		2-EnvelopeHandler.getTheHandler() method is called by Applet1	od_	2-No exception is thrown.		
		Applet1 finalizes.		Applet1 finalizes. Applet2 is triggered		
		3-EnvelopeHandler.getTheHandler() method is called by Applet2	od_	3-No exception is thrown.		
19	9 Envelope Handler availability with EVENT_EVENT_DOWNLOAD_BROWSER_ TERMINATION					
		welope event download browser- mination is sent to the SIM	1- A	pplet1 is triggered		
		evelopeHandler.getTheHandler() methodealled by Applet1	2-N (o exception is thrown.		
	3 En	welopeHandler.getTheHandler() method	Арр	let1 finalizes		
	is c	called by Applet2	3 - A	pplet2 is triggered		
			4- N	o exception is thrown.		
	<u>20</u>	Envelope Handler availability with EVENT EVENT DOWNLOAD BROWSER TERMINATION	₹_			
		1-Envelope event download browser termination is sent to the SIM		1- Applet1 is triggered		
		2-EnvelopeHandler.getTheHandler() method is called by Applet1	od_	2-No exception is thrown.		
				Applet1 finalizes. Applet2 is triggered		
		3-EnvelopeHandler.getTheHandler() method is called by Applet2	od_	3-No exception is thrown.		
20		Envelope Handler availaibility with EVENT_STATUS_COMMAND				
		atus command is sent to the SIM	1- A	pplet1 is triggered		
	2 En	ealled by Applet1	2- A	Toolkit exception		

Ī	ld	Description		API/Framework Expectatio	n	APDU Expectation
			HAN throv	IDLER_NOT_AVAILABLE is	1	
		welopeHandler.getTheHandler() method-		et1 finalizes		
	is c	called by Applet2	4 - A	Polkit exception		
21	E	nvelope Handler availability with EVENT_	throv	IDLER_NOT_AVAILABLE is wa		
	l An SIM	UNRECOGNIZED_ENVELOPE unrecognized Envelope is sent to the	1- ∧	pplet1 is triggered		
		welopeHandler.getTheHandler() method- ealled by Applet1	2- N	o exception is thrown.		
			Appl	et1-finalizes		
		ealled by Applet2	3- ∧	pplet2 is triggered		
			4 - N	o exception is thrown.		

ld	Description	API/Framework Expectation	APDU Expectation
<u>21</u>	Envelope Handler availability with		•
	EVENT STATUS COMMAND		
	1-Status command is sent to the SIM	1- Applet1 is triggered	
	2-EnvelopeHandler.getTheHandler() method	2- A Toolkit exception	
	is called by Applet1	HANDLER NOT AVAILABLE is	
		<u>thrown</u>	
		Applet1 finalizes.	
		3- Applet2 is triggered	
	3-EnvelopeHandler.getTheHandler() method is called by Applet2	4- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
22	Envelope Handler availability with		
	EVENT EVENT DOWNLOAD DATA AVAILAB		
	<u>LE</u>		
	1- Applet1 builds a proactive command OPEN	1 Applet 16 registered to	1- OPEN CHANNEL
	CHANNEL. proactiveHandler.send() method is called	EVENT EVENT DOWNLOAD DA TA_AVAILABLE and	proactive command is fetched
		EVENT_EVENT_DOWNLOAD_CH	<u>ictoricu</u>
		ANNEL STATUS	TERMINAL RESPONSE is
	2-Envelope event download data available	2- Applet1 is triggered	issued with Channel Id = 01
	is sent to the SIM with ChannelId=01.		
	3-EnvelopeHandler.getTheHandler() method is called by Applet1	3-No exception is thrown.	
<u>23</u>	Envelope Handler availability with EVENT_EVENT_DOWNLOAD_CHANNEL_STAT US		
	1-Envelope event download channel status is sent to the SIM with ChannelId=01.	1- Applet1 is triggered	
	2-EnvelopeHandler.getTheHandler() method is called by Applet1	2-No exception is thrown.	
<u>24</u>	Envelope Handler availability with EVENT UNRECOGNIZED ENVELOPE		
	1-An unrecognized Envelope is sent to the SIM	1- Applet1 is triggered	
	2-EnvelopeHandler.getTheHandler() method	2- No exception is thrown.	
	is called by Applet1	2- 140 GAUGPHOIT IS HILUWII.	
		Applet1 finalizes Applet2 is triggered	
	3-EnvelopeHandler.getTheHandler() method is called by Applet2	3- No exception is thrown.	
25			
20	Envelope Handler availability with EVENT FORMATTED SMS PP UPD		
	1- A formatted Update Record EFsms instruction is sent to the SIM	1- Applet1 is triggered	
	2-EnvelopeHandler.getTheHandler() method	2- No exception is thrown.	
	is called by Applet1	-	
26	Envelope Handler availability with		
20		Î.	

Description	API/Framework Expectation	APDU Expectation
EVENT_UNFORMATTED_SMS_PP_UPD		
1-An unformatted Update Record EFsms instruction is sent to the SIM	1- Applet1 is triggered	
2-EnvelopeHandler.getTheHandler() method is called by Applet1	2- No exception is thrown.	
The second secon	Applet1 finalizes Applet2 is triggered	
3-EnvelopeHandler.getTheHandler() method is called by Applet2	3- No exception is thrown.	

6.3.1.3.4 Test Coverage

	CRR Number	Test Case Number
CRRN1	2, 3, 4, 5, (5, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19
CRRC1		1, 20

CRRN1	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 22, 23, 24, 25, 26
	<u>ZZ, ZJ, ZT, ZJ, ZU</u>
CRRC1	<u>1, 2, 21</u>

6.3.1.4 EnvelopeResponseHandler

Test Area Reference: FWK_MHA_ERHD.FWK_MHA_ERHD

6.3.1.4.1 Conformance Requirement

6.3.1.4.1.1 Normal Execution

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

____EVENT_FORMATTED_SMS_PP_ENV
___EVENT_UNFORMATTED_SMS_PP_ENV
__EVENT_CALL_CONTROL
__EVENT_SMS_MO_CONTROL
_EVENT_UNRECOGNIZED_ENVELOPE

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

CRRN3: After a call to the send method the handler is not longer available

6.3.1.4.1.2 Parameters error

No requirements.

6.3.1.4.1.3 Context Errors

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

EVENT FORMATTED SMS CB

EVENT_UNFORMATTED_SMS_CB

EVENT_MENU_SELECTION

EVENT_MENU_SELECTION_HELP_REQUEST

EVENT_TIMER_EXPIRATION

EVENT_EVENT_DOWNLOAD_MT_CALL

EVENT_EVENT_DOWNLOAD_CALL_CONNECTED

EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED

EVENT_EVENT_DOWNLOAD_LOCATION_STATUS

EVENT_EVENT_DOWNLOAD_USER_ACTIVITY

EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE

EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS

EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION

EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION

EVENT_STATUS_COMMAND

EVENT_PROFILE_DOWNLOAD

EVENT_FIRST_COMMAND_AFTER_SELECT

EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS

EVENT_FORMATTED_SMS_PP_UPD

EVENT_UNFORMATTED_SMS_PP_UPD

6.3.1.4.2 Test Suite Files

Test Script:	FWK_MHA_ ERHD_1.ser
Test Applet:	FWK_MHA_ ERHD_1.java
	FWK_MHA_ERHD_2.java
Load Script:	FWK_MHA_ ERHD_1.ldr
Cleanup Script:	FWK_MHA_ ERHD_1.clr
Parameter File: FWK	_MHA_ ERHD_1.par
Test Script:	FWK_MHA_ERHD_1.scr
Test Applet:	FWK_MHA_ERHD_1.java
	FWK_MHA_ERHD_2.java
Load Script:	FWK_MHA_ERHD_1.ldr
Cleanup Script:	FWK_MHA_ERHD_1.clr

Parameter File: FWK MHA ERHD 1.par

6.3.1.4.3 Test Procedure

ld	Description	API/Framework Expectation	APDU Expectatio
rei	Toolkit Applet1 and Toolkit Applet2- gistration and Envelope Response Handler- availability with EVENT_PROFILE_DOWNLOAD		
	Applet1 Toolkit 1 is registered to all-		
Usi	ents defined in [7]. Ing the methods initMenuEntry() for ENT_MENU_SELECTION,		
req	NI_MENO_SERECTION, questPollInterval() for- NT_STATUS_COMMAND, allocateTimer() for-		
EVE	NT_TIMER_EXPIRATION and setEventList() the rest of the events.		
	Plet2 Toolkit 2 is registered to	1- No exception is thrown	
	NT_UNFORMATTED_SMS_PP_ENV and NT_UNRECOGNIZED_ENVELOPE.	·	
	Cerminal Profile command is sent to SIM- Chout the facility of SET_EVENT_LIST	2- Applet1 is triggered	
, SE	TUP_IDLE_MODE_TEXT, SETUP_MENU and		
	e letl is triggered		
	reloneResponseHandler getTheHandler()	3- A Toolkit exception	
		HANDLER_NOT_AVAILABLE is thrown	
	NT_PROFILE_DOWNLOAD		
1	Toolkit Applet1 and Toolkit Applet2 registration and Envelope Response Hand	ler	
	availability with EVENT_FIRST_COMMAND_AFTER_SELECT		
	1- Applet1 is registered to all events		
	defined in TS 43.019 [7]. Using the methods initMenuEntry() for	-	
	EVENT_MENU_SELECTION, requestPollInterval() for		
	EVENT_STATUS_COMMAND, allocateTimer() f EVENT_TIMER_EXPIRATION and setEventList		
	Applet2 is registered to		
	EVENT_UNFORMATTED_SMS_PP_ENV and EVENT_UNRECOGNIZED_ENVELOPE.	1- No exception is thrown	
	4- Select MF.	O. Appletd in triproperd by	
	3-EnvelopeResponseHandler.getTheHandlermethod is called by Applet1	2- Applet1 is triggered by EVENT_FIRST_COMMAND_AF ER_SELECT	Τ
	Applet1 is deregistered to EVENT_FIRST_COMMAND_AFTER_SELECT.		
		3- A Toolkit exception HANDLER_NOT_AVAILABLE is	
		Let	İ
En	velope Response Handler availability with	<u>thrown</u>	
	nvelope Response Handler availability with ENT_MENU_SELECTION_HELP_REQUEST	tnrown	

ld	Description	API/Framework Expectation APDU Expectation
1 E	Envelope menu selection with help- quest is sent to the SIM	1- Applet1 is triggered.
	EnvelopeResponseHandler.getTheHandler()- thod is called by Applet1	2 A Toolkit exception HANDLER_NOT_AVAILABLE is thrown
2	Handler availability with EVENT PROFILE DOWNLOAD	
	1- Terminal Profile command is sent to the SIM without the facility of SET_EVENT_LIST, SETUP_IDLE_MODE_TEXT, SETUP_MENU and POLL_INTERVAL.	1- Applet1 Is Triggered By EVENT PROFILE DOWNLOAD
	2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1 Applet1 is deregistered to EVENT_PROFILE_DOWNLOAD	2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown
3	Envelope Response Handler availability winder EVENT_MENU_SELECTION_HELP_REQUE Perform SIM initialization with all the facilities supported	<u>ST</u>
	1-Envelope menu selection with help request is sent to the SIM	1- Applet1 is triggered.
	2-EnvelopeResponseHandler.getTheHandlermethod is called by Applet1	2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown
1 A	Nelope Response Handler availability with EVENT_MENU_SELECTION Nenvelope menu selection is sent to SIM	1- Applet1 is triggered
The	2 Appletl is triggered	
2 E met	EnvelopeResponseHandler.getTheHandler()- Thed is called by Applet1	2- A Toolkit exception HANDLER_NOT_AVAILABLE is- thrown

Id Description API/Fran	nework Expectation APDU Expectation
4 Envelope Response Handler availability with	
EVENT MENU SELECTION	
1-A envelope menu selection is sent to 1- Applet1 i	a triaggered
1-A envelope menu selection is sent to the SIM	<u>s triggered</u>
2 Flored and Demonstrated law matter diam ()	
2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	
2- A Toolkit	
	NOT AVAILABLE is
<u>thrown</u>	
4 Envelope Response Handler availability with EVENT_UNFORMATTED_CB	
EVENT_ONI ONIMATTED_OD	
1 Envelope cell broadcast unformatted is 1- Applet1 is tri	ggered.
sent to the SIM	
2 EnvelopeResponseHandler.getTheHandler()	
method is called by Applet1	SEPTION OT AVAILABLE is
thrown	71_7.V/116/10CC 10
l l l l l l l l l l l l l l l l l l l	
5 Envelope Response Handler availability with	
EVENT_TIMER_EXPIRATION	
1 Envelope Timer Expiration is sent to	
1 Envelope Timer Expiration is sent to the SIM 1- Applet1 is tri	ggered.
2 EnvelopeResponseHandler.getTheHandler() 2- A Toolkit exc	ception
	OT_AVAILABLE is
thrown thrown	
5 Envelope Response Handler availability with	
EVENT_FORMATTED_CB	
1-Envelope cell broadcast formatted is 1- The appl	et1 is triggered.
sent to the SIM	
2-A Toolkit	exception
2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1 HANDLER	NOT AVAILABLE is
thrown	THO I THE TOTAL TO
6 Envelope Response Handler availability with	
EVENT_EVENT_DOWNLOAD_MT_CALL	
1 Envelope event download mt call is sent 1- Applet1 is tri	ggered
to the SIM	ggerou.
2 EnvelopeResponseHandler.getTheHandler() 2-A Toolkit exc	
method is called by Applet1 HANDLER_NO	T_AVAILABLE is
thrown thrown	
6 Envelope Response Handler availability with	
EVENT_UNFORMATTED_CB	
1-Envelope cell broadcast unformatted is sent to the SIM	s triggered.
Sent to the Sim	
2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1 2- A Toolkit	exception
	NOT AVAILABLE is
thrown	

	ld	Description	API/Framework Expectation	APDU Expectation
7	E	velope Response Handler availability with	•	•
		ENT EVENT DOWNLOAD CALL CONNEC		
		TED		
	1 F	Envelope event download call connected	1- Applet1 is triggered.	
	is	sent to the SIM		
	2 I	hvelopeResponseHandler.getTheHandler()		
	met		2- A Toolkit exception	
			HANDLER_NOT_AVAILABLE is	
			thrown	
<u> </u>		Facelone Bossess Handler availability of	41-	
	<u>7</u>	Envelope Response Handler availability wi	<u>tn</u>	
		EVENT TIMER EXPIRATION		
		1-Envelope Timer Expiration is sent to	1- Applet1 is triggered.	
		the SIM	1- Applet is triggered.	
		2-EnvelopeResponseHandler.getTheHandler	2- A Toolkit exception	
		method is called by Applet1	HANDLER_NOT_AVAILABLE is	
			thrown	
8	E	velope Response Handler availability with		·
		VENT_EVENT_DOWNLOAD_CALL_DISCON		
		NECTED		
	1 F	Invelope event download call	1- Applet1 is triggered.	
	ui.	beoinected is bent to the sim		
			O. A. T II '	
	2 I		2- A Toolkit exception	
	met		HANDLER_NOT_AVAILABLE is thrown	
			HHOWH	
<u> </u>	8	Envelope Response Handler availability wi	th	
	<u>u</u>	EVENT_EVENT_DOWNLOAD_MT_CALL	<u>ur</u>	
		1-Envelope event download mt call is se	nt 1- Applet1 is triggered.	
		to the SIM	33	
		2-EnvelopeResponseHandler.getTheHandler	2 -A Toolkit exception	
		method is called by Applet1	HANDLER NOT AVAILABLE is	
			<u>thrown</u>	
لـِـا ا	_			
9		nvelope Response Handler availability with		
	₽V	ENT_EVENT_DOWNLOAD_LOCATION_STA TUS		
		+03		
	1 E	invelope event download location status	1- Applet1 is triggered	
	is	sent to the SIM	T Applet to triggered.	
	Î			
	2. z	applet A obtains the Envelope Response		
		idler	2- A Toolkit exception	
	ĺ		HANDLER_NOT_AVAILABLE is	
	Î		thrown	
ļ <u>L</u>	1			
	<u>9</u>	Envelope Response Handler availability wi		
		EVENT EVENT DOWNLOAD CALL CONN	<u>EC</u>	
		<u>TED</u>		
		1 Berelon	a d Amminut in this court	
		1-Envelope event download call connecte is sent to the SIM	a 1- Applet1 is triggered.	
		TO SELLE CO CHE SIM		
	_			

	Description	API/Framework Expectation	APDU Expectation
	2-EnvelopeResponseHandler.getTheHandlermethod is called by Applet1	2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	
	 nvelope Response Handler availability with /ENT_EVENT_DOWNLOAD_USER_ACTIVIT ¥	1- Applet1 is triggered.	
	Envelope event download user activity sent to the SIM	2- A Toolkit exception	
2 I met	EnvelopeResponseHandler.getTheHandler() chod is called by Applet1 Envelope Response Handler availability w	HANDLER_NOT_AVAILABLE is thrown	
10	EIVEROPE RESPONSE HANGIE AVAILABILITY W EVENT EVENT DOWNLOAD CALL DISCO NECTED 1-Envelope event download call disconnected is sent to the SIM	1- Applet1 is triggered.	
	2-EnvelopeResponseHandler.getTheHandlermethod is called by Applet1	2- A Toolkit exception HANDLER NOT AVAILABLE is thrown	
1 F	Ent_EVENT_DOWNLOAD_IDLE_SCREEN_ AVAILABLE Envelope event download idle screen- vilable is sent to the SIM EnvelopeResponseHandler.getTheHandler() End is called by Applet1	1- Applet1 is triggered.	
		2- A Toolkit exception- HANDLER_NOT_AVAILABLE is-	
11	Envelope Response Handler availability w EVENT_EVENT_DOWNLOAD_LOCATION_S	HANDLER_NOT_AVAILABLE is thrown	
11	Envelope Response Handler availability w	HANDLER_NOT_AVAILABLE is- thrown ith STA	
11	Envelope Response Handler availability w EVENT_EVENT_DOWNLOAD_LOCATION_S TUS 1-Envelope event download location state	HANDLER_NOT_AVAILABLE is thrown ith STA 1- Applet1 is triggered.	
2 E	Envelope Response Handler availability w EVENT_EVENT_DOWNLOAD_LOCATION_S TUS 1-Envelope event download location statis sent to the SIM 2-Applet1 obtains the Envelope Response	HANDLER_NOT_AVAILABLE is thrown ith STA 1- Applet1 is triggered. 2- A Toolkit exception HANDLER NOT AVAILABLE is thrown	
2 Ex	Envelope Response Handler availability w EVENT_EVENT_DOWNLOAD_LOCATION_S TUS 1-Envelope event download location state is sent to the SIM 2-Applet1 obtains the Envelope Response Handler Evelope Response Handler availability with ENT_EVENT_DOWNLOAD_CARD_READER	HANDLER_NOT_AVAILABLE is thrown ith STA 1- Applet1 is triggered. 2- A Toolkit exception HANDLER NOT AVAILABLE is thrown	

		ld	Description	API/Framework Expectation	APDU Expectation
ļ		<u>12</u>	Envelope Response Handler availability with EVENT EVENT DOWNLOAD USER ACTIVITY		
			1-Envelope event download user activity is sent to the SIM	1- Applet1 is triggered.	
			2-EnvelopeResponseHandler.getTheHandler(method is called by Applet1	2- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown	_
	13		EVENT_STATUS_COMMAND Status command is sent to the SIM	- Applet1 is triggered	
			H	- A Toolkit exception ANDLER_NOT_AVAILABLE is- rown	

	ld	Description		API/Framework Expectation	n	APDU Expectation	
	<u>13</u>	Envelope Response Handler availability w EVENT EVENT DOWNLOAD IDLE SCREE AVAILABLE 1-Envelope event download idle screen available is sent to the SIM		1- Applet1 is triggered.			
		2-EnvelopeResponseHandler.getTheHandlermethod is called by Applet1	<u>r()</u>	2- A Toolkit exception_ HANDLER_NOT_AVAILABLE i thrown	<u>s</u>		
14	Er	velope Response Handler availability with EVENT_FORMATTED_SMS_PP_ENV					
		formatted sms pp envelope is sent to— SIM	4	Applet1 is triggered			
	met 3 A	nvelopeResponseHandler.getTheHandler() hod is called by Applet1 pplet1 builds an additional	2- l	No exception is thrown.			
	cal	ormation for response packet and it- ls the post method plet1 calls all methods of the			3 ser	The response packet is- nt-	
	Env inh	Envelope Response Handler (including the inherited method)		4- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method			
		<u>EVENT_FORMATTED_SMS_PP_ENV envelope</u> sent to the SIM	- 5- /	Applet1 is triggered			
	met	nvelopeResponseHandler.getTheHandler() hod is called by Applet1	6- l	No Exception is thrown	7	The preactive command	
	it	pplet1 builds a proactive command and calls the send() method pplet1 calls all methods of the clope Response Handler (including the crited method)	HA	Toolkit exception NDLER_NOT_AVAILABLE is- own for each method	is s	eent	
	14	Envelope Response Handler availability w EVENT EVENT DOWNLOAD CARD READ _STATUS					
		1-Envelope event download card reader status is sent to the SIM		1- Applet1 is triggered			
		2-EnvelopeResponseHandler.getTheHandlermethod is called by Applet1	r()	2- A Toolkit exception HANDLER_NOT_AVAILABLE i thrown	<u>s</u>		

	ld	Description	API/Framework Expectation	APDU Expectation
15	Er	nvelope Response Handler availability with EVENT_UNFORMATTED_SMS_PP_ENV		
		EVENT_UNFURMATTED_SMS_PP_ENV		
	15	Envelope Response Handler availability with		
		EVENT_EVENT_DOWNLOAD_LANGUAGE_ SELECTION		
		1-Envelope event download language selection is sent to the SIM	4. Appletd is triuman.	
		Before to the SIM	1- Applet1 is triggered	
		2-EnvelopeResponseHandler.getTheHandler(
		method is called by Applet1		
			2-A Toolkit exception HANDLER_NOT_AVAILABLE is	
			thrown	
L				

		ld	Description		API/Framework Expectatio	n	APDU Expectation
ſ	16		velope Response Handler availability with			Ī	
			EVENT_CALL_CONTROL_BY_SIM				
		<u> 16</u>	Envelope Response Handler availability w	ith			
			EVENT_EVENT_DOWNLOAD_BROWSER	_			
			TERMINATION				
			1-Envelope event download browser				
			termination is sent to the SIM		1- Applet1 is triggered		
			2-EnvelopeResponseHandler.getTheHandler	c()			
			method is called by Applet1	. ,			
					2-A Toolkit exception HANDLER NOT AVAILABLE i	_	
					thrown	<u>s</u>	
					unown		
ſ	17	En	velope Response Handler availability with		1		<u> </u>
		EV	ENT_MO_SHORT_MESSAGE_CONTROL_B				
		-	Y_SIM		American I		
			nvelope mo short message control by is sent to the SIM	1-/	Applet1 is triggered		
		2 E	nvelopeResponseHandler.getTheHandler()	2-1	No exception is thrown.		
		met	hod is called by Applet1				
			pplet1 builds the envelope response				The envelope response is
			it calls the postAsBERTLV() method			ser	nt-
		4 A	pplet1 calls all methods of the				
		Env	relope Response Handler (including the	4	A Toolkit exception		
		inh	erited method)	HA	NDLER_NOT_AVAILABLE is		
		The	Applet1 finalizes		own for each method		
				5-/	Applet1 is triggered		
		sim	is sent to the SIM				
		6-E	nvelopeResponseHandler.getTheHandler()				
			hod is called by Applet1	சுப	No exception is thrown		
		<u>7</u> 7	pplet1 builds a proactive command and	Ĭ '	TO SACOPAGE TO CHOWIT		
			calls the send method			7-	The proactive command
L						•	

 ld	Description		API/Framework Expectation	APDU Expectation
En√	pplet1 calls all methods of the elope Response Handler (including the erited method)	HΑ		is fetched and the Terminal- Response is issued
<u>17</u>	Envelope Response Handler availability wind EVENT STATUS COMMAND 1-Status command is sent to the SIM	<u>th</u>	1- Applet1 is triggered	
	2-EnvelopeResponseHandler.getTheHandlermethod is called by Applet1	r()	2- A Toolkit exception HANDLER NOT AVAILABLE is thrown	<u>.</u>

П	d Description	API/Framework Expectation	APDU Expectation
18	d Description Envelope Response Handler availability with EVENT_UNRECOGNIZED_ENVELOPE	7.11 7.1 Tallio Work Expectation	711 DO EXPOSICION
	EVENT_UNRECOGNIZED_ENVELOPE		
	_		

ld	Description	API/Framework Expectatio	n APDU Expectation
18	Envelope Response Handler availability w		1- The OPEN CHANNEL
	EVENT EVENT DOWNLOAD DATA AVAIL		command is fetched.
	<u>LE</u>		
	_	2- Applet1 is triggered	TERMINAL RESPONSE IS
	1- Applet1 initialises a proactive	2/2	SENT TO THE SIM with
	command OPEN CHANNEL and calls the send method.		channelId=01
	meeroa.	O. A. T 1136	
		3- A Toolkit exception	
	2- Envelope event download data avalaik		<u>IS</u>
	is sent to the SIM with channelId=01	<u>thrown</u>	
	3-EnvelopeResponseHandler.getTheHandlermethod is called by Applet1	<u>r()</u>	
	method is carred by Appreti		
9	The envelope response is sent when a		
	proactive session is ongoing		
	A formatted SMS PP envelope is sent to	1- Applet1 is triggered.	
th	e-SIM.		
	Proactive command DISPLAY TEXT is built		2-91-XX
and	d it calls the send() method.		2 31 AA
3;	A call control by sim envelope is sent-		
	the SIM.	3- Applet1 is triggered	
		, ipplett is inggered	
4-1	EnvelopeResponseHandler.getTheHandler()		
	thod is called by Applet1		
ļ.,	Applet1 builds the envelope response	4- No exception is thrown	
	Applet: Dullas the envelope response d it calls the postAsBERTLV		
GII	a to datib one postilissinis,		5-The envelope response is
			sent
			9F YY
			GET RESPONSE
			Data
			91 XX
			Fetch DISPLAY TEXT
			Terminal Response
			DISPLAY TEXT
TE:	Due to an inconsistency in GSM 03.19 [7] sp	ecification it is not possible to cover	
	applet try to post data in multitriggering.	·	
<u>19</u>			
	EVENT EVENT DOWNLOAD CHANNEL S	STA	
	<u>TUS</u>	4. Applied to difference 1	
		1- Applet1 is triggered	
	1- Envelope event download channel stat	tus	
	is sent to the SIM with channelId=01	2- A Toolkit exception	
		HANDLER_NOT_AVAILABLE i	is
		thrown	_
	2-EnvelopeResponseHandler.getTheHandler	<u>r()</u>	
	method is called by Applet1		
1			
20	Envelope Pasponso Handler availability with	h l	
<u>20</u>		<u>th</u>	
<u>20</u>	Envelope Response Handler availability wit EVENT FORMATTED SMS PP UPD	t <u>h</u>	
<u>20</u>	EVENT FORMATTED SMS PP UPD 1- Update Record EFsms instruction	t <u>h</u>	
<u>20</u>	EVENT FORMATTED SMS PP UPD	1- The applet1 is triggered.	
<u>20</u>	EVENT FORMATTED SMS PP UPD 1- Update Record EFsms instruction		
<u>20</u>	EVENT FORMATTED SMS PP UPD 1- Update Record EFsms instruction		

171

Description	API/Framework Expectation	APDU Expectation
2-EnvelopeResponseHandler.getTheHandler()		
method is called by Applet1	2-A Toolkit exception	
	·	
	unown	
	-EnvelopeResponseHandler.getTheHandler() ethod is called by Applet1	-EnvelopeResponseHandler.getTheHandler()

ld	Description	API/Framework Expectation	APDU Expectation
<u>21</u>	Envelope Response Handler availability with EVENT UNFORMATTED SMS PP UPD		
	EVENT UNFORMATTED SMS PP UPD		
	1- Update Record EFsms instruction unformatted is sent to the SIM	1- Applet1 is triggered.	
	anioimaceca is sene to the sim		
	2-EnvelopeResponseHandler.getTheHandler()		
	method is called by Applet1	2- A Toolkit exception HANDLER NOT AVAILABLE is	
		thrown	
<u>22</u>	Event Formatted SMS PP ENV		
	EVERT FORMATTED SING IT ERV		
	1-A formatted sms pp envelope is sent to	4. Appletd is tripped	
	the SIM	1- Applet1 is triggered	
	2. EnvelopeDegrengeHerdler		
	2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	2- No exception is thrown.	
	3-Applet1 builds an additional		3- The response packet is
	information for response packet and it		sent
	calls the post method		
	4-Applet1 calls all methods of the Envelope Response Handler (including the	4- A Toolkit exception	
	inherited method)	HANDLER NOT AVAILABLE is	
		thrown for each method	
	5-A EVENT_FORMATTED_SMS_PP_ENV envelope	Applet1 finalizes	
	is sent to the SIM	5- Applet1 is triggered	
	6-EnvelopeResponseHandler.getTheHandler()		
	method is called by Applet1	6- No Exception is thrown	7- The proactive command
	7-Applet1 builds a proactive command and it calls the send() method		is sent
	8-Applet1 calls all methods of the Envelope Response Handler (including the	8- Toolkit exception	
	inherited method)	HANDLER NOT AVAILABLE is	
23	Envelope Response Handler availability with	thrown for each method	
	EVENT_UNFORMATTED_SMS_PP_ENV		
	1-An unformatted sms pp envelope is sent	1- Applet1 is triggered	
	to the SIM	delicer of middle on	
	2-EnvelopeResponseHandler.getTheHandler()	2- No exception is thrown.	
	method is called by Applet1	2 140 CACCPROTT IS THEOWILE	
	3-Applet1 builds the envelope response		
	and it calls the post() method		3- The envelope response
	4- Applet1 calls all methods of the	4- A Toolkit exception	<u>is sent</u>
	Envelope Response Handler (including the inherited method)	HANDLER NOT AVAILABLE is	
		thrown for each method Applet1 finalizes	
		<u>Uphieri iiriaii7e2</u>	
		5- Applet2 is triggered.	
	5-EnvelopeResponseHandler.getTheHandler()		
	method is called	A Toolkit exception	
		HANDLER NOT AVAILABLE is thrown.	
		u II OWII.	

ld	Description	API/Framework Expectation	APDU Expectation
	6-An unformatted sms pp envelope is sent to the SIM	Applet2 finalizes 6- Applet1 is triggered.	
	7-EnvelopeResponseHandler.getTheHandler() method is called. 8-Applet1 builds a proactive command and it calls the send() method	7- No exception is thrown.	
	9-Applet1 calls all methods of the Envelope Response Handler (including the inherited method)	O. A Tabilita avanting	9- The proactive command is fetched and the Terminal response is issued.
		9- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method.	
		Applet1 finalizes 10- Applet2 is triggered.	
	10- EnvelopeResponseHandler.getTheHandler() method is called by Applet2		
		A Toolkit exception HANDLER NOT AVAILABLE is thrown.	

ld	Description	API/Framework Expectation	APDU Expectation
<u>24</u>	Envelope Response Handler availability with	·	•
	EVENT CALL CONTROL BY SIM		
	1-Envelope call control by sim is sent to the SIM	1- Applet1 is triggered	
	2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1		
	3-Applet1 builds the envelope response and it calls the postAsBERTLV() method	2- No exception is thrown.	3- The envelope response is sent
	4-Applet1 calls all methods of the Envelope Response Handler (including the inherited method)	4- Toolkit exception HANDLER_NOT_AVAILABLE is	<u>is seni</u>
	5-Envelope call control by sim is sent to the SIM	thrown for each method Applet1 finalizes	
	6-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	5- Applet1 is triggered	
	7-Applet1 builds a proactive command and it calls the send() method	6- No Exception is thrown	
	8-Applet1 calls all methods of the Envelope Response Handler (including the inherited method)		7- The proactive command is fetched and the Terminal response is issued
		8- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method	
<u>25</u>	ENT_MO_SHORT_MESSAGE_CONTROL_B		
	Y SIM 1-Envelope mo short message control by sim is sent to the SIM	1- Applet1 is triggered	
	2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	2- No exception is thrown.	
	3-Applet1 builds the envelope response and it calls the postAsBERTLV() method		3-The envelope response is sent
	4-Applet1 calls all methods of the Envelope Response Handler (including the inherited method)	4- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method	
	5-Envelope mo short message control by sim is sent to the SIM	Applet1 finalizes	
	6-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	5- Applet1 is triggered	
	7-Applet1 builds a proactive command and it calls the send method	6- No exception is thrown	7- The proactive command
	8-Applet1 calls all methods of the Envelope Response Handler (including the inherited method)		is fetched and the Terminal Response is issued
		8- A Toolkit exception HANDLER NOT AVAILABLE is thrown for each method	
<u>26</u>	Envelope Response Handler availability with EVENT_UNRECOGNIZED_ENVELOPE		
	1-An unrecognized Envelope is sent to the SIM	1- Applet1 is triggered	
	2-EnvelopeResponseHandler.getTheHandler()		

Description	API/Framework Expectation	APDU Expectation
method is called by Applet1	2- No exception is thrown.	
3-Applet1 builds the envelope response and it calls the postAsBERTLV() or post method 4-Applet1 calls all methods of Envelope Response Handler (including the inherite method)	4- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method	3- The envelope response is sent
5-EnvelopeResponseHandler.getTheHandler(method is called	Applet1 finalizes 5- Applet2 is triggered.	
6-An unrecognized Envelope is sent to th SIM 7-EnvelopeResponseHandler.getTheHandler(HANDLER NOT AVAILABLE is	
method is called 8-Applet1 builds a proactive command and it calls the send() method	Applet2 finalizes	
9-Applet1 calls all methods of the Envelope Response Handler (including the inherited method)	7- No exception is thrown.	9- The proactive command is fetched and the Terminal response is issued
10- EnvelopeResponseHandler.getTheHandler() method is called by Applet2	9- A Toolkit exception HANDLER_NOT_AVAILABLE is thrown for each method Applet1 finalizes 10- Applet2 is triggered A Toolkit exception HANDLER_NOT_AVAILABLE is thrown.	

ld	Description	API/Framework Expectation	APDU Expectation
<u>27</u>	The envelope response is sent when a	•	•
	<pre>proactive session is ongoing 1-A formatted SMS PP envelope is sent to the SIM.</pre>	1- Applet1 is triggered.	
	2-Proactive command DISPLAY TEXT is built and it calls the send() method. 3-A call control by sim envelope is sent to the SIM.	O. Annierd in trianged	<u>2- 91 XX</u>
	4-EnvelopeResponseHandler.getTheHandler()	3- Applet1 is triggered	
	method is called by Applet1 5-Applet1 builds the envelope response and it calls the postAsBERTLV	4- No exception is thrown	
			5-The envelope response is sent 9F YY
			GET RESPONSE Data 91 XX Fetch DISPLAY TEXT
			Terminal Response DISPLAY TEXT
<u>28</u>	Envelope Response Handler availability with EVENT_UNFORMATTED_SMS_PP_ENV in		<u>DISPLAT TEXT</u>
	case of multi-triggering	1- Applet1 is triggered	
	1-A unformatted sms pp envelope is sent to the SIM 2-EnvelopeResponseHandler.getTheHandler()	2- No exception is thrown.	
	method is called by Applet1	3- Applet1 finalizes	
		4- Applet2 is triggered.	
	5-EnvelopeResponseHandler.getTheHandler() method is called by Applet 2	5- No Exception is thrown	6. The response is checked.
	6- Applet2 calls the post() method		
		Applet2 finalizes	
<u>29</u>	Envelope Response Handler availability with	1- Applet1 is triggered	
	EVENT_UNRECOGNIZED_ENVELOPE in case of multi-triggering	2- No exception is thrown.	
	1-An unrecognized Envelope is sent to the SIM		
	2-EnvelopeResponseHandler.getTheHandler() method is called by Applet1	3- Applet1 finalizes	

ld	Description	API/Framework Expectation	APDU Expectation
		4- Applet2 is triggered.	
		5- No Exception is thrown	
	5-EnvelopeResponseHandler.getTheHandler() method is called by Applet 2		
		Applet2 finalizes	6- The response is checked
	6- Applet2 calls the post() method		

6.3.1.4.4 Test Coverage

CRR Number	Test Case Number
CRRN1	14, 15, 16, 17, 18, 19
CRRN2	14, 15, 16, 17, 18, 19
CRRN3	14, 15, 16, 17, 18, 19
CRRC1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

6.3.1.4.4 Test Coverage

CRR Number	Test Case Number	
CRRN1	20, 21, 22, 23, 24, 25, 26, 27	
CRRN2	<u>20, 21, 22, 23, 24, 25</u>	
CRRN3	<u>20, 21, 22, 23, 24, 25</u>	
CRRC1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,	
	28, 29	

6.3.2 Handler Integrity

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.

<u>CRRN3</u>: 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.

<u>CRRN6</u> The Device Identity Simple TLV is used to store the information about the absolute record number in the <u>EFsms</u> file and the value of the <u>EFsms</u> record status byte.

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_

╛	Description	API/Framework Expectation	APDU Expectation
	Applet initialization and Envelope Handler	•	-
	integrity checks with		
	EVENT MENU SELECTION HELP REQUEST		
'	EVENI_MENU_JELEGIION_RELF_REQUESI		
1	Applet is registered to all events	4. N.L. averagette at the server	
4	efined in [7] except	1-No exception is thrown	
	VENT PROFILE DOWNLOAD and		
100	VENT_FROFILE_DOWNLOAD und VENT_STATUS_COMMAND.		
77			
0	sing the methods initMenuEntry() for VENT MENU SELECTION, allocateTimer()for		
25	VENT_TIMER_EXPIRATION, and setEventList()		
4			
I±	or the rest of the events.		
Ŧ	acilities supported		
	The sales was an allowed an allowed by the land of the sales was a sales	2- Applet is triggered	
	Envelope menu selection with help request		
+	s sent to the SIM	3-No exception is thrown.	
		O 140 CACOPHOLLIS HILOWIL.	
	EnvelopeHandler.getTheHandler() method is		
c	alled		
١,		4-No exception is thrown	
	Copy the contents of the envelope handler buffer 1 using EnvelopeHandler.copy()		
1.	n buffer I using EnvelopeHandler.copy()		
_			
	he EnvelopeHandler.findTLV() method is		
ď	alled with TAC_HELP_REQUEST		
Ļ	A properties command DIGDIAN MENUE is seen		F 04
-	A proactive command DISPLAY TEXT is sent		5-91 xx.
_	Envelope call control by sim is sent to	6- Applet is triggered	
	- Envelope call control by sim is sent to 		
Ö	IN		
B	nvelopeHandler.getTheHandler() method is		
	alled		
			
7		7-No exception is thrown and the	
ė	nvelope handler is the envelope call	handler contains the envelope call	
	ontrol using EnvelopeHandler.copy() and	control by SIM	
U	til.arrayCompare() methods	-	
Ī			
Ŧ	he EnvelopeHandler.findTLV() method is		
	alled with TAC DEVICE IDENTITIES		
Ĭ .			
C	all Control execution is finished.		
			-A proactive command-
			Display Text is fetched
1			The terminal Response
			DISPLAY TEXT is sent
			the SIM
C :	heck that the TAC_HELP_REQUEST is the TLV		

ld	Description	API/Framework Expectation	APDU Expectation
	8 The contents of EnvelopeHandler are	8-The contents of the envelope handler shall be the same as stored	
	compared with bufferl using	in buffer 1	
	Util.arrayCompare()	The state of the s	
1	Applet initialization and Envelope Handler		
-	integrity checks with		
	EVENT MENU SELECTION HELP REQUEST		
	1- Applet is registered to all events	1-No exception is thrown	
	defined in TS 43.019 [7] except	1-NO exception is thrown	
	EVENT_PROFILE_DOWNLOAD and EVENT_STATUS_COMMAND.		
	Using the methods initMenuEntry() for		
	EVENT_MENU_SELECTION, allocateTimer()for_		
	<pre>EVENT_TIMER_EXPIRATION, and setEventList() for the rest of the events.</pre>		
	Perform SIM initialization with all the		
	facilities supported		
	2-Envelope menu selection with help request	2- Applet is triggered	
	is sent to the SIM	3- No exception is thrown.	
	3-EnvelopeHandler.getTheHandler() method is		
	called		
		A No suspend of the disc	
	4-Copy the contents of the envelope handler	4- No exception is thrown	
	in buffer 1 using EnvelopeHandler.copy()		
	The EnvelopeHandler.findTLV() method is		
	called with TAG_HELP_REQUEST		
	5-A proactive command DISPLAY TEXT is sent		<u>5- 91 xx.</u>
		6- Applet is triggered	O OT XX.
	6-Envelope call control by sim is sent to SIM		
	EnvelopeHandler.getTheHandler() method is		
	called		
		7- No exception is thrown and the	
	7- It's checked that the contents of the envelope handler is the envelope call	handler contains the envelope call	
	control using EnvelopeHandler.copy() and	control by SIM	
	Util.arrayCompare() methods		
	The EnvelopeHandler.findTLV() method is		
	called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
	The state of the s		A proportive comment of
			A proactive command Display Text is fetched
			The terminal Response of
			DISPLAY TEXT is sent to the SIM
			uno Onvi
	Check that the TAG_HELP_REQUEST is the TLV selected		
	<u>serected</u>	8- The contents of the envelope	
	8-The contents of EnvelopeHandler are	handler shall be the same as stored	
	<pre>compared with buffer1 using Util.arrayCompare()</pre>	in buffer 1	
2	Envelope Handler integrity checks with EVENT_MENU_SELECTION		
	EVENT_WENU_SELECTION		
	1 An envelope menu selection is sent to	1- Applet is triggered	
	SIM		

	Description	API/Framework Expectation	APDU Expectation
		2-No exception is thrown.	
	2 EnvelopeHandler.getTheHandler() method is called		
		3-No exception is thrown.	
	3 Copy the contents of the envelope		
	EnvelopeHandler.copy()		
	The EnvelopeHandler.findTLV() method is		
е	called with TAG_ITEM_IDENTIFIER		4.04.307
4	1 A proactive command DISPLAY TEXT is sent		4-91 XX
<u> </u>	5 Envelope call control by sim is sent to	5Applet is triggered	
2	51N		
	EnvelopeHandler.getTheHandler() method is		
е	called		
Ę	5 It's checked the contents of the	6- No exception is thrown and the handler contains the envelope call	
€	envelope handler is the envelope call- control using EnvelopeHandler.copy() and	control by SIM	
	control using EnvelopeHandler.copy() and Jtil.arrayCompare() methods		
	The EnvelopeHandler.findTLV() method is		
e	called with TAG_DEVICE_IDENTITIES		
e	Call Control execution is finished.		Propositive command Displa
			Proactive command Displate Text is fetched
			The terminal Response of DISPLAY TEXT is sent to
			the SIM
	It's checked that the TAG_ITEM_IDENTIFIER		
i	is the TLV selected		
7	7 The contents of EnvelopeHandler are	7- The contents of the envelope	
	compared with bufferl using J til.arrayCompare()	handler shall be the same as stored in buffer 1	
2	Envelope Handler integrity checks with	In puner 1	
_	Livelope Handler Integrity Checks with		
	EVENT_MENU_SELECTION		
1		1- Applet is triggered	
1	EVENT_MENU_SELECTION 1-An envelope menu selection is sent to SIM	1- Applet is triggered	
	l-An envelope menu selection is sent to SIM		
2			
<u>2</u>	1-An envelope menu selection is sent to SIM 2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
200	1-An envelope menu selection is sent to SIM 2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler	2- No exception is thrown.	
2 2 2 3 <u>i</u>	2-Envelope menu selection is sent to SIM 2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy()	2- No exception is thrown.	
2 2 3 <u>i</u>	1-An envelope menu selection is sent to SIM 2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler	2- No exception is thrown.	
3 1 1	2-Envelope menu selection 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	2- No exception is thrown.	
3 <u>i</u> <u>T</u>	2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_ITEM_IDENTIFIER 4-A proactive command DISPLAY TEXT is sent	2- No exception is thrown.	4- 91 XX
2 c c c c c c c c c c c c c c c c c c c	2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_ITEM_IDENTIFIER	2- No exception is thrown. 3- No exception is thrown.	<u>4- 91 XX</u>
2 2 3 1 1 5 4 5 8	2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_ITEM_IDENTIFIER 4-A proactive command DISPLAY TEXT is sent 5-Envelope call control by sim is sent to SIM	2- No exception is thrown.	<u>4- 91 XX</u>
2 C 3 1 1 C 4 5 SS E	2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_ITEM_IDENTIFIER 4-A proactive command DISPLAY TEXT is sent 5-Envelope call control by sim is sent to	2- No exception is thrown. 3- No exception is thrown.	<u>4- 91 XX</u>
2 C 3 1 1 C 4 5 SS E	2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_ITEM_IDENTIFIER 4-A proactive command DISPLAY TEXT is sent 5-Envelope call control by sim is sent to SIM EnvelopeHandler.getTheHandler() method is	2- No exception is thrown. 3- No exception is thrown.	<u>4- 91 XX</u>
2 C 3 1 H O 4 5 1 2 E O 6	2-EnvelopeHandler.getTheHandler() method is called 3-Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_ITEM_IDENTIFIER 4-A proactive command DISPLAY TEXT is sent 5-Envelope call control by sim is sent to SIM EnvelopeHandler.getTheHandler() method is	2- No exception is thrown. 3- No exception is thrown.	<u>4- 91 XX</u>

ld	Description	API/Framework Expectation	APDU Expectation
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES Call Control execution is finished.	control by SIM	
	It's checked that the TAG_ITEM_IDENTIFIER is the TLV selected 7- The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM

ld	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		4-91 XX
	4 A proactive command DISPLAY TEXT is sent-		101700
	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 TAC_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Displ Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
	It's checked that the TAC_SMS_TPDU is the TLV selected	7. The contents of the envelope	
	7 The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	
<u>3</u>	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		<u>4- 91 XX</u>
	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		

ld	Description	API/Framework Expectation	APDU Expectation
	envelope handler is the envelope call	6- No exception is thrown and the	
	<pre>control using EnvelopeHandler.copy() and Util.arrayCompare methods</pre>	handler contains the envelope call control by SIM	
		CONTROL BY SHVI	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	<u>Call Control execution is finished.</u>		
			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		<u>uno onvi</u>
	TLV selected		
	7- The contents of EnvelopeHandler are	7. The contents of the convolute	
	compared with buffer1 using	7- The contents of the envelope handler shall be the same as stored	
	<pre>Util.arrayCompare()</pre>	in buffer 1	

1d 4	Description Envelope Handler integrity checks with EVENT_UNFORMATTED_SMS_PP_ENV	API/Framework Expectation	APDU Expectation
		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		4-91-XX
	4 A proactive command DISPLAY TEXT is sent		4-9+ AA
	5 Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is ealled		
	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 TAC_DEVICE_IDENTITIES		
	Call Control execution is finished.		Proactive command Displa
			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.	
<u>4</u>	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- 91 XX
	4-A proactive command DISPLAY TEXT is sent		
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		

ld	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	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 the envelope	
5	7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare() Envelope Handler integrity checks with	handler shall be the same as stored in buffer 1.	
	EVENT_UNFORMATTED_SMS_CB	1- Applet is triggered	
	sent to SIM	11 11 11 10 10	
	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 TAC_CELLBROADCAST_PAGE 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 ealled	6- No exception is thrown and the	
	6 It's checked that the contents of the envelope handler is the envelope call-control using EnvelopeHandler.copy and Util.arrayCompare() methods	handler contains the envelope call- control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAC_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 CELLBROADCAST PAGE is the TLV selected		
	7 The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare()	7—The contents of the envelope- handler shall be the same as stored in buffer 1.	

	Description	API/Framework Expectation	APDU Expectation
	Envelope Handler integrity checks with EVENT UNFORMATTED SMS CB		
-	A unformatted cellbroadcast envelope is ent to SIM	1- Applet is triggered	
	-EnvelopeHandler.getTheHandler() method is	2- No exception is thrown.	
-	alled		
3	-Copy the contents of the envelope handler	2. No exception is thrown	
i	n buffer 1 using EnvelopeHandler.copy()	5- NO exception is tillown.	
יי	The EnvelopeHandler.findTLV() method is		
	called with TAG_CELLBROADCAST_PAGE		4- 91 XX
4	-A proactive command DISPLAY TEXT is sent		4 01 ///
	-Envelope call control by sim is sent to	5- Applet is triggered	
	<u>SIM</u>		
	EnvelopeHandler.getTheHandler() method is called		
6	-It's checked that the contents of the	6- No exception is thrown and the handler contains the envelope call	
	envelope handler is the envelope call	control by SIM	
	til.arrayCompare() methods		
	The EnvelopeHandler.findTLV() method is		
	Call Control execution is finished.		
_	ari Control execution is limished.		
			Proactive command Displate Text is fetched
			The terminal Response of
			DISPLAY TEXT is sent to
			the SIM
_	t's checked that the		
7	AG_CELLBROADCAST_PAGE is the TLV selected	7- The contents of the envelope	
	- The contents of EnvelopeHandler are compared with buffer1 using	handler shall be the same as stored in buffer 1.	
	Util.arrayCompare()_	<u></u>	

ld 6	Description Envelope Handler integrity checks with EVENT_TIMER_EXPIRATION	API/Framework Expectation	APDU Expectation
		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_TIMER_ID		
	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 iscalled with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		Proactive command Displa Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	It's checked that the TAG_TIMER_ID is the		
	7 The contents of EnvelopeHandler are- compared with bufferl using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	
<u>6</u>	Envelope Handler integrity checks with EVENT_TIMER_EXPIRATION		
	1-A timer expiration 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_TIMER_ID		
	4-A proactive command DISPLAY TEXT is sent 5-Envelope call control by sim is sent to SIM	5- Applet is triggered	<u>4- 91 XX</u>
	EnvelopeHandler.getTheHandler() method is		

L	Description	API/Framework Expectation	APDU Expectation
	<u>called</u>		
	6-It's checked that the contents of the envelope handler is the envelope call	6- No exception is thrown and the	
	control using EnvelopeHandler.copy() and	handler contains the envelope call control by SIM	
	Util.arrayCompare() methods	CONTROL BY SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Disp 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		
	7- The contents of EnvelopeHandler are	7- The contents of the envelope	
	compared with bufferl using Util.arrayCompare()	handler shall be the same as stored	
	Envelope Handler integrity checks with	in buffer 1	
	EVENT_CALL_CONTROL_BY_SIM		
	1 A gell gentrel envelope is sent to CIM	1. Applet is triggered	
	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	3-No exception is thrown.	
	EnvelopeHandler.copy()		
	The EnvelopeHandler.findTLV() method is called with TAC_ADDRESS		
	4 A proactive command DISPLAY TEXT is sent		4-91-XX
			101700
	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	6- No exception is thrown and the	
	envelope handler is the envelope call control using EnvelopeHandler.copy() and	handler contains the envelope call- control by SIM-	
	Util.arrayCompare() methods	Control by Chivi	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Disp Text is fetched
			The tempiral Division
			The terminal Response of DISPLAY TEXT is sent to
			the SIM
	It's checked that the TAG_ADDRESS is the		
		<u> </u>	
	TLV selected	7- The contents of the envelope	
	TLV selected 7 The contents of EnvelopeHandler are compared with bufferl using	7- The contents of the envelope handler shall be the same as stored in buffer 1	

ld	Description	API/Framework Expectation	APDU Expectation
7	Envelope Handler integrity checks with EVENT CALL CONTROL BY SIM	AFI/Framework Expectation	AFDO Expectation
	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		<u>4- 91 XX</u>
	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 Beanance of
			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	
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 TAC_ADDRESS		4-91 XX
	4 A proactive command DISPLAY TEXT is sent		4***
	SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is called		

ld	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 TAC_DEVICE_IDENTITIES	6- No exception is thrown and the handler contains the envelope call-control by SIM	
	Call Control execution is finished.		Preactive 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 bufferl using Util.arrayCompare()	7—The contents of the envelope- handler shall be the same as stored in buffer 1.	
<u>8</u>	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- 91 XX
	4-A proactive command DISPLAY TEXT is sent		
	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.	

	Description	API/Framework Expectation	APDU Expectation
9	Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_MT_CALL		
	EVENT_DOWNEDAD_IIIT_GALE		
	1 A event download mt call envelope is sent	1-	
	CO SIM		
		O Nie ausantian is thereon	
	<pre>2 EnvelopeHandler.getTheHandler() method is called</pre>	2-No exception is thrown.	
	Carred		
	-3 Copy the contents of the envelope	3-No exception is thrown.	
	handler in buffer 1 using EnvelopeHandler.copy()		
	The EnvelopeHandler.findTLV() method is called with TAC_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	6- No exception is thrown and the	
	control using EnvelopeHandler.copy() and	handler contains the envelope call control by SIM	
	Util.arrayCompare() methods		
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Disp
			Text is fetched
			The terminal Response o
	It's checked that the TAG_ADDRESS is the		DISPLAY TEXT is sent to the SIM
	TLV selected		
	TLV selected 7 The contents of EnvelopeHandler are	7- The contents of the envelope- handler shall be the same as stored	
	TLV selected 7 The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare()		
<u>9</u>	The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare() Envelope Handler integrity checks with EVENT	handler shall be the same as stored	
<u>9</u>	The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare() Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD MT_CALL	handler shall be the same as stored in buffer 1	
9	The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare() Envelope Handler integrity checks with EVENT	handler shall be the same as stored in buffer 1	
9	The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare() Envelope Handler integrity checks with EVENT EVENT DOWNLOAD MT CALL 1-A event download mt call envelope is sent	handler shall be the same as stored in buffer 1	
<u>9</u>	The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare() Envelope Handler integrity checks with EVENT EVENT DOWNLOAD MT CALL 1-A event download mt call envelope is sent to SIM	handler shall be the same as stored in buffer 1 1- Applet is triggered	
9	The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare() Envelope Handler integrity checks with EVENT EVENT DOWNLOAD MT CALL 1-A event download mt call envelope is sent	handler shall be the same as stored in buffer 1 1- Applet is triggered	
9	The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare() 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	handler shall be the same as stored in buffer 1 1- Applet is triggered	
9	The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare() 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	handler shall be the same as stored in buffer 1 1- Applet is triggered 2- No exception is thrown.	DISPLAY TEXT is sent to the SIM
9	The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare() 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()	handler shall be the same as stored in buffer 1 1- Applet is triggered 2- No exception is thrown.	
9	The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare() 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	handler shall be the same as stored in buffer 1 1- Applet is triggered 2- No exception is thrown.	
9	The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare() 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	handler shall be the same as stored in buffer 1 1- Applet is triggered 2- No exception is thrown.	the SIM
<u>9</u>	The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare() 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	handler shall be the same as stored in buffer 1 1- Applet is triggered 2- No exception is thrown.	
9	The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare() 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	handler shall be the same as stored in buffer 1 1- Applet is triggered 2- No exception is thrown.	the SIM

ld	Description EnvelopeHandler.getTheHandler() method is	API/Framework Expectation	APDU Expectation
	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.	6- No exception is thrown and the handler contains the envelope call control by SIM	
	It's checked that the TAG_ADDRESS is the TLV selected 7- The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	Proactive command Disp Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
Ф	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 TAC_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 ealled		
	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 TAC_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Disp 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		

ld	Description	API/Framework Expectation 7- The contents of the envelope	APDU Expectation
	7 The contents of EnvelopeHandler are compared with bufferl using	handler shall be the same as stored in buffer 1.	
	Util.arrayCompare()	In buller 1.	
10	Envelope Handler integrity checks with EVENT EVENT DOWNLOAD CALL CONNECTED 1-A event download call connected envelope	1- Applet is triggered	
	is sent to SIM	O No constitution in the const	
	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	3- No exception is thrown.	
	called with TAG_ADDRESS 4-A proactive command DISPLAY TEXT is sent		<u>4- 91 XX</u>
	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	6- No exception is thrown and the	
	envelope handler is the envelope call control using EnvelopeHandler.copy and Util.arrayCompare() methods	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 bufferl using	7- The contents of the envelope handler shall be the same as stored in buffer 1.	
4.4	Util.arrayCompare()		
77	Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED 1 A event_download_call_disconnected_	1- Applet is triggered	
	envelope is sent to SIM		
	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	3-No exception is thrown.	
	ealled with TAG_ADDRESS 4 A proactive command DISPLAY TEXT is sent		4-91 XX

ld	Description	API/Framework Expectation 5- Applet is triggered	APDU Expectation
	5 Envelope call control by sim is sent to SIM	o- Appiet is triggered	
	EnvelopeHandler.getTheHandler() method is		
	called		
	6 It's checked that the contents of the	6- No exception is thrown and the	
	envelope handler is the envelope call- control using EnvelopeHandler.copy and	handler contains the envelope call- control by SIM	
	Util.arrayCompare() methods	COTATOL BY SHWI	
	The EnvelopeHandler.findTLV() method is		
	called with TAC_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			D
			Proactive command Displayers Text is fetched
			The terminal Response of
			DISPLAY TEXT is sent to
			the SIM
	It's checked that the TAC_ADDRESS is the		
	TLV selected	7. The contests of the seveles	
	7 The contents of EnvelopeHandler are	7- The contents of the envelope handler shall be the same as stored	
	<pre>compared with buffer1 using- Util.arrayCompare()</pre>	in buffer 1.	
1	Envelope Handler integrity checks with EVENT		
	EVENT DOWNLOAD CALL DISCONNECTED		
	1-A event download call disconnected	1- Applet is triggered	
	envelope is sent to SIM		
	2-EnvelopeHandler.getTheHandler() method is called	2- No exception is thrown.	
	<u></u>		
	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 04 VV
	4-A proactive command DISPLAY TEXT is sent		<u>4- 91 XX</u>
	5-Envelope call control by sim is sent to	5- Applet is triggered	
	SIM		
	EnvelopeHandler.getTheHandler() method is		
	called		
	6-It's checked that the contents of the	6- No exception is thrown and the	
	envelope handler is the envelope call	handler contains the envelope call	
	control using EnvelopeHandler.copy and Util.arrayCompare() methods	control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES		
	Call Control execution is finished.		
			Proactive command Disp
			Text is fetched
			The terminal Response on DISPLAY TEXT is sent to
			the SIM
	It's checked that the TAG_ADDRESS is the		
	TLV selected that the TAG_ADDRESS IS the		
	ITA BETECCEA		

ld	Description	API/Framework Expectation	APDU Expectation
Iu	7- The contents of EnvelopeHandler are	7- The contents of the envelope	APDO Expectation
	compared with buffer1 using	handler shall be the same as stored	
	Util.arrayCompare()	in buffer 1.	
12	Envelope Handler integrity checks with EVENT_		
	EVENT DOWNLOAD LOCATION STATUS		
	1 A event download location status envelope	1- Applet is triggered	
	is sent to SIM		
		2 No exception is thrown	
	2 EnvelopeHandler.getTheHandler() method is	2-140 exception is thrown.	
	carrea		
	-3 Copy the contents of the envelope handler in buffer 1 using	3-No exception is thrown.	
	EnvelopeHandler.copy()		
	The EnvelopeHandler.findTLV() method is		
	called with TAC_LOCATION_STATUS		4-91 XX
	4 A proactive command DISPLAY TEXT is sent		
	5 Envelope call control by sim is sent to SIM	5- Applet is triggered	
	DIT!		
	EnvelopeHandler.getTheHandler() method is called		
	carrea		
	6 It's checked that the contents of the	6- No exception is thrown and the	
	envelope handler is the envelope call	handler contains the envelope call	
	<pre>control using EnvelopeHandler.copy() and Util.arrayCompare() methods</pre>	control by SIM	
	The EnvelopeHandler.findTLV() method is		
	called with TAC_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_LOCATION_STATUS		
	is the TLV selected		
	7 The contents of EnvelopeHandler are	7- The contents of the envelope	
	compared with bufferl using	handler shall be the same as stored	
	Util.arrayCompare()	in buffer 1	
<u>12</u>	Envelope Handler integrity checks with EVENT EVENT_DOWNLOAD_LOCATION_STATUS		
	EVENT_DOWNLOAD_LOCATION_STATUS		
	1-A event download location status envelope	1- Applet is triggered	
	is sent to SIM		
	2-EnvelopeHandler.getTheHandler() method is	2- No exception is thrown.	
	<u>called</u>		
	3-Copy the contents of the envelope handler	3- No exception is thrown.	
	in buffer 1 using EnvelopeHandler.copy()	The strong and the st	
	The EnvelopeHandler findTIV/\ mothod is		
	The EnvelopeHandler.findTLV() method is called with TAG_LOCATION_STATUS		
	4-A proactive command DISPLAY TEXT is sent		<u>4-91 XX</u>
	5-Envelope call control by sim is sent to		
	SIM		

ld	Description	API/Framework Expectation	APDU Expectation
		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 The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES Call Control execution is finished.	6- No exception is thrown and the handler contains the envelope call control by SIM	
			Proactive command Display Text is fetched
	It's checked that the TAG_LOCATION_STATUS is the TLV selected		The terminal Response of DISPLAY TEXT is sent to the SIM
	7- The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	

ld	Description	API/Framework Expectation	APDU Expectation
	Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_USER_ACTIVITY	Ar // ramework Expectation	Al Do Expectation
	1 A event download user activity 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		4-91 XX
	4 A proactive command DISPLAY TEXT is sent 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 TAC_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 The contents of EnvelopeHandler are compared with bufferl using	7- The contents of the envelope handler shall be the same as stored in buffer 1	
	Util.arrayCompare()		
13	Envelope Handler integrity checks with EVENT EVENT DOWNLOAD USER ACTIVITY		
	1-A event download user activity 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	<u> </u>
	EnvelopeHandler.getTheHandler() method is		

ld	Description	API/Framework Expectation	APDU Expectation
	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
	It's checked that the TAG DEVICE IDENTITIES		The terminal Response of DISPLAY TEXT is sent to the SIM
	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	

ld	Description	API/Framework Expectation	APDU Expectation
14	Envelope Handler integrity checks with EVENT_ EVENT_DOWNLOAD_IDLE_SCREEN_AVAILAB LE	·	
	1 A event download idle screen available envelope is sent to SIM	1- Applet is triggered	
	2 EnvelopeHandler.getTheHandler() method is ealled	2-No exception is thrown.	
	3 Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is	3-No exception is thrown.	
	called with TAC_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 TAC_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 bufferl using Util.arrayCompare()	7- The contents of the envelope handler shall be the same as stored in buffer 1	
<u>14</u>	Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD IDLE SCREEN AVAILAB <u>LE</u>		
	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	<u>4- 91 XX</u>

melopetander getThetandler() method is called 6-Tit's checked that the contents of the envelope handler is the envelope call control using Envelopetandler (not) and provide petandler (not) method is called with TAG_DEVICE_IDENTITIES Call Control execution is finished. Proactive command Text is fetched The terminal Respondistry to the contents of Envelopetandler are compared with Envelopetandler are compared to the envelope handler shall be the same as stored in buffer 1 4. Envelopetandler getTheliandler() method is called with TAG_CALD_READER_STATUS 2. No exception is thrown. 3. No exception is thrown. 4. Applet is triggered 4. On exception is thrown. 4. Applet is triggered 4. On exception is thrown. 5. Envelopetandler are three contents of the envelope called three contents of the envelope called three contents of the envelope called three contents are sent to sent three called three contents of the envelope called three contents are envelope called three contents the envelope called three contents are envelope called three called control by SIM.	ctation
called with TAG_DEVICE_IDENTITIES Call Control execution is finished. Proactive command Text is fetched The terminal Respond DISPLAY TEXT is and the envelope handler and unit in the TAG_DEVICE_IDENTITIES is the TLV selected 7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare() 146 Envelope Handler integrity checks with EVENT_EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS 1	
The terminal Responding Display Text is a the TIV selected 7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare() 18 Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS 1 A event download card reader status—envelope is sent to SIM 2 EnvelopeHandler.getTheliandler() method is called 3 Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTIV() method is called with TAQ_CARD_READER_STATUS 4 A proactive command DISPLAY TEXT is sent 5 Envelope call control by sim is sent to SIM 5 EnvelopeHandler.getTheliandler() method is called 6 It's checked that the contents of the envelope call control using EnvelopeHandler.copy and Util.arrayCompare() method is called The EnvelopeHandler.findTIV() method is called The Envelope Called The Envelope Called The EnvelopeHandler.findTIV() method is called The Envelope Called The Envelope Called The EnvelopeHandler.findTIV() method is called The Envel	nd Displa
compared with buffer1 using Util.arrayCompare() T- The contents of the envelope handler shall be the same as stored in buffer 1 Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS 1 A event download card reader status 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.findTiv() 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 EnvelopeHandler.getTheHandler() method is called 6 It's checked that the contents of the envelope call control using EnvelopeHandler.copy and Util.arrayCompare() methods The EnvelopeHandler.findTiv() method is called on the envelope call the envelope call the EnvelopeHandler.findTiv() method is called the envelope called the EnvelopeHandler.findTiv() method is called the envelope called the envelope called the envelopeHandler.findTiv() method is called the envelopeHandler.findTiv() method is called the envelope called the envelopeHandler.findTiv() method is called the	
EVENT_DOWNLOAD_CARD_READER_STATUS 1 A event download card reader status— 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_CARD_READER_STATUS 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 6 No exception is thrown and the handler contains the envelope call 1- Applet is triggered 2- No exception is thrown.	
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_CARD_READER_STATUS 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 candler contains the envelope call-control using EnvelopeHandler.copy and Util.arrayCompare() methods The EnvelopeHandler.findTLV() method is candler contains the envelope call-control using EnvelopeHandler.copy and Util.arrayCompare() methods	
called 3 Copy the contents of the envelope handler in buffer 1 using EnvelopeHandler.copy() The EnvelopeHandler.findTLV() method is called with TAG_CARD_READER_STATUS 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 6 No exception is thrown. 3-No exception is thrown. 4-91 XX 5-Applet is triggered 6 No exception is thrown and the handler contents of the conten	
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 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 contains the envelope call control using EnvelopeHandler.copy and Util.arrayCompare() methods The EnvelopeHandler.findTLV() method is contains the envelope call contains the envelope c	
5- Applet is triggered EnvelopeHandler.getTheHandler() method is ealled 6	
5-Applet is triggered EnvelopeHandler.getTheHandler() method is ealled 6	
5-Applet is triggered EnvelopeHandler.getTheHandler() method is ealled 6	
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	
envelope handler is the envelope call- control using EnvelopeHandler.copy and Util.arrayCompare() methods The EnvelopeHandler.findTLV() method is— **The	
The EnvelopeHandler.findTLV() method is handler contains the envelope call	
called with TAG_DEVICE_IDENTITIES CONTOL BY SIM	
Proactive command Text is fetched	nd Displa
The terminal Respo	
It's shecked that the TAG_CARD_READER_STATUS is the TLV selected the TLV selected	
7 The contents of EnvelopeHandler are compared with bufferl using	

ld	Description Util.arrayCompare()	API/Framework Expectation	APDU Expectation
	otii.airaytompare()	in buffer 1	
<u>15</u>	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	<u>4- 91 XX</u>
	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	6- No exception is thrown and the handler contains the envelope call control by SIM	
	called with TAG_DEVICE_IDENTITIES	SOLITO BY SIM	
			Proactive command Display Text is fetched
	It's checked that the		The terminal Response of DISPLAY TEXT is sent to the SIM
	TAG_CARD_READER_STATUS is the TLV selected		LITE SHAL
	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	

1d 16	Description Envelope Handler integrity checks with UNRECOGNIZED_ENVELOPE	API/Framework Expectation	APDU Expectation
	1 A unrecognized envelope is sent to SIM	1- Applet is triggered	
	2 EnvelopeHandler.getTheHandler() method is ealled	2-No exception is thrown.	
	-3 Copy the contents of the envelope- handler in buffer 1 using EnvelopeHandler.copy()	3-No exception is thrown.	
	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 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	6-No exception is thrown and the handler contains the envelope call-control by SIM	
	The EnvelopeHandler.findTLV() method is called with TAC_DEVICE_IDENTITIES		
	Call Control execution is finished.		Proactive command Display Text is fetched
			The terminal Response of DISPLAY TEXT is sent to the SIM
	7 The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare()	7- The contents of the envelope- handler shall be the same as stored in buffer 1	
<u>16</u>	Envelope Handler integrity checks with UNRECOGNIZED ENVELOPE		
	1-A unrecognized 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() 4-A proactive command DISPLAY TEXT is sent	3- No exception is thrown.	
	5-Envelope call control by sim is sent to SIM	5- Applet is triggered	4- 91 XX
	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	6- No exception is thrown and the handler contains the envelope call control by SIM	

ld	Description	API/Framework Expectation	APDU Expectation
	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
	7- The contents of EnvelopeHandler are	7- The contents of the envelope	
	compared with buffer1 using Util.arrayCompare()	handler shall be the same as stored in buffer 1	

ld	Description	API/Framework Expectation	APDU Expectation
17	Envelope Handler integrity checks with		·
	EVENT_EVENT_DOWNLOAD_LANGUAGE_SEL ECTION		
	LOTION		
	1-A event download language selection 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_EVENT_LIST		
	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_EVENT_LIST 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	
18	Envelope Handler integrity checks with EVENT_EVENT_DOWNLOAD_BROWSER_TERM INATION		
	1-A event download browser termination 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_EVENT_LIST		
	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

ld	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
	It's checked that the TAG_EVENT_LIST is the TLV selected		the- SIM
	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	

I

ld	Description	API/Framework Expectation	APDU Expectation
19	Envelope Handler integrity checks with EVENT_FORMATTED_SMS_CB		20 Expodution
	1-An envelope SMS-CB formatted according to TS 23.048 [8] 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_CELL_BROADCAST_PAGE	3-No exception is thrown.	
	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	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	6- No exception is thrown and the	
	The EnvelopeHandler.findTLV() method is called with TAG_DEVICE_IDENTITIES	handler contains the envelope call control by SIM	
	Call Control execution is finished.		
	It's checked that the TAG_CELL_BROADCAST_PAGE is the TLV selected 7- The contents of EnvelopeHandler are compared with buffer1 using Util.arrayCompare()		Proactive command Display Text is fetched The terminal Response of - DISPLAY TEXT is sent to the –SIM
		7- The contents of the envelope handler shall be the same as stored in buffer 1	
<u>20</u>	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()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV() method is called with TAG_SMS_TPDU	2 140 CACCPRIOTES THOWIL.	
	4-A proactive command DISPLAY TEXT is sent		<u>4- 91 XX</u>

Description	API/Framework Expectation	APDU Expectation
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	
It's checked that the TAG_SMS_TPDU is the TLV selected 7- The contents of EnvelopeHandler are compared with bufferl using Util.arrayCompare()		Proactive command Display Text is fetched The terminal Response of DISPLAY TEXT is sent to the SIM
	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_SMS_TPDU Call Control execution is finished. It's checked that the TAG_SMS_TPDU is the TLV selected 7- The contents of EnvelopeHandler are compared with bufferl using	5-Envelope call control by sim is sent to SIM 5-Applet is triggered 6-No exception is thrown and the handler 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 Call Control execution is finished. It's checked that the TAG_SMS_TPDU is the TLV selected 7- The contents of EnvelopeHandler are compared with buffer1 using Thil arrayCompare()

ld	Description	API/Framework Expectation	APDU Expectation
<u>21</u>	Envelope Handler integrity checks with		
	EVENT UNFORMATTED SMS PP UPD		
	1- Update Record EFsms instruction single	1- Applet is triggered	
	and unformatted is sent to the SIM		
	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()		
	in butter 1 using EnvelopeHandler.copy()	3- No exception is thrown.	
	The EnvelopeHandler.findTLV method is		
	called with TAG_SMS_TPDU		4 04 VV
	4-A proactive command DISPLAY TEXT is sent		<u>4- 91 XX</u>
	5-Envelope call control by sim is sent to	- A - 1 - 1 - 1 - 1 - 1 - 1 - 1	
	SIM	5- Applet is triggered	
	EnvelopeHandler.getTheHandler() method is		
	<u>called</u>		
	6-It's checked that the contents of the envelope handler is the envelope call	6- No exception is thrown and the	
	control using EnvelopeHandler.copy() and	handler contains the envelope call control by SIM	
	Util.arrayCompare() methods	<u> </u>	
	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
	<pre>It's checked that the TAG_DEVICE_IDENTITIES is the TLV selected</pre>		
		7- The contents of the envelope	
	7- The contents of EnvelopeHandler are compared with buffer1 using	handler shall be the same as stored	
	Util.arrayCompare()	in buffer 1.	
22	Check the TLV list conversion for		
	EVENT FORMATTED SMS PP UPD		
	1- An EVENT_FORMATTED_SMS_PP_UPD is sent to	1- Applet is triggered	
	the SIM.	2- No exception is thrown.	
	2- The findTLV(tag == device identities Tag) is called.		
	<pre>3- The getValueByte(offset == 0) is called.</pre>	3- return the absolute record.	
	4- The getValueByte(offset == 1) is called.	4- return the record status	
	5- The findTLV(tag == address Tag) is	5- No exception is thrown.	
	called.		
	6- Check the content	7. No evention is through	
	7- The findTLV(tag == SMS TPDU Tag) is	7- No exception is thrown.	
	called.		

ld	Description	API/Framework Expectation	APDU Expectation
	8- Check the content		
<u>23</u>	Check TLV list conversion for EVENT FORMATTED SMS PP UPD		
	1- The getLength() method is called	return the Simple TLV list length	
	_ ine geometry meaned in edited	1. Tetari tile olimpie TEV list lengti	
<u>24</u>	Check TLV list conversion for EVENT_FORMATTED_SMS_PP_UPD		
	1- The getEnvelopeTag() method is called	1- return BTAG_SMS_PP_DOWNLOAD	
<u>25</u>	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	2- No exception is thrown.	
	<pre>Tag) is called. 3- The getValueByte(offset == 0) is called.</pre>	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		
26	Check TLV list conversion for		
<u>26</u>	EVENT_UNFORMATTED_SMS_PP_UPD		
	1- The getLength() method is called	1. return the Simple TLV list length	
<u>27</u>	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

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

6.3.2.4 EnvelopeResponseHandler

Test Area Reference: FWK HIN ERHD

6.3.2.4.1 Conformance Requirement

6.3.2.4.1.1 Normal Execution

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

6.3.2.4.2 Test Suite Files:

Test Script: FWK_HIN_ERHD_1.scr

Test Applet: FWK_HIN_ERHD_1.java

Load Script: FWK_HIN_ERHD_1.ldr

Cleanup Script: FWK_HIN_ERHD_1.clr

Parameter File: FWK HIN ERHD 1.par

6.3.2.4.3 Test Procedure

ld	<u>Description</u>	API/Framework Expectation	APDU Expectation
1	Applet1 is registered to EVENT UNRECOGNIZED ENVELOPE.		
		1- Applet 1 is triggered.	
	2- EnvelopeResponseHandler.getTheHandler()is		
	called by the Applet1.		
	<pre>3- EnvelopeResponseHandler.getLength()</pre>	2- The return value shall be 0.	
	method is called by Applet1		

6.3.2.4.4 Test Coverage

CRR Number	Test Case Number	
CRRN1	<u>1</u>	

6.3.3 Applet Triggering

6.3.3.3 EVENT_MENU_SELECTION_HELP_REQUEST

Test Area Reference: FWK_APT_EMSH-

6.3.3.3.1 Conformance Requirement

6.3.3.3.1.1 Normal Execution

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

6.3.3.1.2 Parameters error

No requirements.

6.3.3.1.3 Context Errors

No requirements. CCRN2: A toolkit applet shall be triggered by the

EVENT MENU SELECTION HELP REQUEST event only if the Menu Id corresponding to the Envelope Menu Selection Help Request received by the SIM Toolkit framework was registered with the *helpSupported* value set to true.

CCRN3: If at least one menuId of a Toolkit Applet registers to EVENT_MENU_SELECTION_HELP_REQUEST, the SET UP MENU proactive command sent by the SIM Toolkit Framework shall indicate to the ME that help information is available unless all the menus entries that support help are disabled.

6.3.3.3.2 Test Suite Files

Test Script: ____FWK_APT_EMSH_1.scr

Test Applet: FWK_APT_EMSH_1.java

FWK_APT_EMSH_2.java

Load Script: FWK_APT_EMSH_1.ldr

Cleanup Script: FWK_APT_EMSH_1.clr

Parameter File: FWK APT EMSH 1.java

FWK_APT_EMSH_2.java

FWK_APT_EMSH_3.java

Load Script: FWK_APT_EMSH_1.ldr

Cleanup Script: FWK_APT_EMSH_1.clr

Parameter File: FWK_APT_EMSH_1.par

6.3.3.3.3 Test Procedure

ld	Description	API/Framework Expectatio	n APDU Expectation
	Applet registration to		
EVENT_I	WENU_SELECTION_HELP_REQUEST		
	and triggering		
ToolkitRo	egistry.InitMenuEntry() method is		
	the constructor of applet1 and		
Applet2.			
For Apple	etl÷		
	/="Applet1"		
Offset=0			
Length=mc	enuEntry.length		
	erted=true		
IconQuali			
IconIdent	:ifier=0		
For Apple	2t2÷		
MenuEntry	/="Applet2"		
Offset=0			
	enuEntry.length		
	orted=true		
IconQuali			
IconIdent	:ifier=0		
event= EV	VENT_MENU_SELECTION_HELP_REQUEST	4	
1-Toolkit	Registry.isEventSet() is called	1-The command must return true.	
in constr	ructor.		
Perform S	SIM initialization the facility		
	ENU and without the facilities SET		
EVENT LIS	GT and POLL INTEVAL features		

I	d Description	API/Framework Expectation	APDU Expectation
2 м	- Item identifier =1 Cenu Selection Help Request envelope is	2- Applet1 is triggered and applet2	
9	ent to the SIM with the item identifier	is not triggered	
0	i a menu entry oi applet		
2	Itom identifier =2	Applet1 finalizes	
M M	enu Selection Help Request envelope is	3- Applet2 is triggered and applet1	
9	ent to the SIM with the item identifier for a menu entry of applet	is not triggered	
	Applet registration to	TeT.	
	EVENT_MENU_SELECTION_HELP_REQUI	<u>=51</u>	
	Applet1 and Applet2 are installed		
	ToolkitRegistry.InitMenuEntry() method called in the constructor of Applet1 ar		
	Applet2.		
	For Applet1 (item id 1):		
	<pre>MenuEntry="Applet1A" Offset=0</pre>		
	Length=menuEntry.length		
	<pre>HelpSupported=true IconQualifier=0</pre>		
	<pre>IconIdentifier=0</pre>		
	For Applet1 (item id 2): MenuEntry="Applet1B"		
	Offset=0 Length=menuEntry.length		
	HelpSupported=false		
	IconQualifier=0 IconIdentifier=0		
	event= EVENT_MENU_SELECTION_HELP_REQUES	C.T.	
	<pre>1- ToolkitRegistry.isEventSet() is call</pre>		
	in constructor.	1- The command shall return true.	
	For Applet2 (item id 3):	1- The command shall return true.	
	MenuEntry="Applet2A"		
	Offset=0 Length=menuEntry.length		
	HelpSupported=true		
	<pre>IconQualifier=0 IconIdentifier=0</pre>		
	For Applet2 (item id 4):		
	MenuEntry="Applet2B"		
	Offset=0 Length=menuEntry.length		
	HelpSupported=false IconQualifier=0		
	IconIdentifier=0		
	event= EVENT_MENU_SELECTION_HELP_REQUES	2- The command shall return true.	
	2- ToolkitRegistry.isEventSet() is call		
	in constructor.		
	Perform SIM initialization with the facility SET UP MENU and without the		
	facilities SET EVENT LIST and POLL		
	INTERVAL		
	3-Item identifier = 1	3- Applet1 is triggered and Applet2	
	Menu Selection Help Request envelope is	s is not triggered	
	sent to the SIM with item identifier 1 belonging to applet1		

ld	Description	API/Framework Expectation	APDU Expectation
	4-Item identifier = 2 Menu Selection Help Request envelope is sent to the SIM with item identifier 2 belonging to applet1	4 Applet1 and Applet2 are not triggered	
	5-Item identifier = 3 Menu Selection Help Request envelope is sent to the SIM with item identifier 3 belonging to applet2	5- Applet2 is triggered and Applet1 is not triggered	
	6-Item identifier = 4 Menu Selection Help Request envelope is sent to the SIM with item identifier 4 belonging to applet2	6- Applet2 and Applet1 are not triggered	
2	Applet deregistration to EVENT_MENU_SELECTION_HELP_REQUEST		
	Applet1 and Applet2 are deleted Applet3 is installed		
	ToolkitRegistry.InitMenuEntry() method is called in the constructor of Applet3. For Applet3 (item id 5): MenuEntry="Applet3A" Offset=0		
	Length=menuEntry.length HelpSupported=true IconQualifier=0 IconIdentifier=0		
	For Applet3 (item id 6): MenuEntry="Applet3B" Offset=0 Length=menuEntry.length HelpSupported=true IconQualifier=0 IconIdentifier=0		
	For Applet3 (item id 7): MenuEntry="Applet3C" Offset=0 Length=menuEntry.length HelpSupported=false IconQualifier=0 IconIdentifier=0		
	1. Perform SIM initialization with the facility SET UP MENU and without the facilities SET EVENT LIST and POLL INTERVAL		1. The SIM shall issue a SET UP MENU proactive command with Menu Entry
	2. Menu Selection Help Request envelope is sent to the SIM with item identifier 5 belonging to applet3		ID entry '05', '06' and '07', and Help supported set to true.
	3. ToolkitRegistry.disableMenuEntry() method for item id 5 is called by the Menu Selection Help Request Envelope.		3. The SIM shall issue a SET UP MENU proactive command with Menu Entry

ld	Description	API/Framework Expectation	APDU Expectation
	4. Menu Selection Help Request envelope is		ID entry '06' and '07', and
	sent to the SIM with item identifier 6		Help supported set to true.
	belonging to applet3		
	<pre>5. ToolkitRegistry.disableMenuEntry()</pre>		
	method for item id 6 is called by the Menu	4. Applet3 is triggered by	
	Selection Help Request Envelope.	EVENT MENU SELECTION HEL	
		P_REQUEST	
			5. The SIM shall issue a
			SET UP MENU proactive
			command with Menu Entry
			ID entry '07', and Help
			supported set to false.

6.3.3.3.4 Test Coverage

CRR Number	Test Case Number	
CRRN1	1	
CRRN2	<u>1</u>	
CRRN3	<u>2</u>	

6.3.3.4 EVENT_FORMATTED_SMS_PP_ENV

Test Area Reference: FWK_APT_EFSE-

6.3.3.4.1 Conformance Requirement

6.3.3.4.1.1 Normal Execution

—CRRN1: The applet is triggered by the EVENT_FORMATTED_SMS_PP_ENV once:

- —it has been registered to this event; event,
- an envelope APDU carrying an SMS TPDUa Short Message Point to Point (Single or Concatenated) is received by Envelope APDU(s) and is formatted according to GSM 03.48 [8], is received; TS 23.048 [8].
- —the toolkit applet to be triggered is registered with the corresponding TAR in the SMS TPDU; TPDU,
- —the security is verified.

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

6.3.3.4.1.2 Parameters error No requirements. 6.3.3.4.1.3 No requirements. 6.3.3.4.2 **Test Suite Files** _FWK_APT_EFSE_1.scr Test Script: Test Applet: FWK_APT_EFSE_1.java **Load Script:** FWK_APT_EFSE_1.ldr Cleanup Script: FWK_APT_EFSE_1.clr Parameter File: FWK APT EFSE 1.java Load Script: FWK_APT_EFSE_1.ldr

Cleanup Script: FWK APT EFSE 1.clr

Parameter File: FWK_APT_EFSE_1.par

6.3.3.4.3 Test Procedure

	ld	Description	API/Framework Expectation	APDU Expectation
4		Applet registration to EVENT		
	F	ORMATTED_SMS_PP_ENV and triggering		
	App	let is registered to		
	EVE	NT_FORMATTED_SMS_PP_ENV and		
	EVE	NT_UNRECOGNIZED_ENVELOPE		
	1 7.	n Envelope EVENT FORMATTED SMS PP ENV		
	is	sent to the SIM.	1- Applet is triggered	
	<u>1</u>	Applet registration to EVENT		
		FORMATTED_SMS_PP_ENV and triggering	<u>ng</u>	
		Applet is registered to		
		EVENT_FORMATTED_SMS_PP_ENV and_ EVENT_UNRECOGNIZED_ENVELOPE		
		EVENT_UNRECOGNIZED_ENVELOPE		
		1- A Single Short Message SMS-PP Format	cted	
		Data Download is sent to the SIM.		
		2- A Concatenated Short Message SMS-PP		
		Formatted Data Download is sent to the		
		(composed of 2 Short Messages. The UDL		
		the first Short Message is 70 and for t second 70)		
		second 70)	2- Applet is triggered	
ļ				
2		Applet deregistration		
	Too	lkitRegistry.clearEvent() method is		
	cal	led for EVENT_FORMATTED_SMS_PP_ENV		
	2 7	formatted and an envelope is gent to		
	the	formatted sms pp envelope is sent to SIM.	1- Applet is not triggered	
	An :	unrecognized envelope is sent to the		
	sim			
	Too.	lkitRegistry.setEvent() method is led for EVENT FORMATTED SMS PP ENV		
	car	rea for hybrigholding broght him		
		n Envelope FORMATTED_SMS_PP_ENV is sent	2- Applet is triggered	
	+0	the SIM		
	2	Applet deregistration		
		ToolkitRegistry.clearEvent() method is called for EVENT FORMATTED SMS PP ENV	_	
		1- A Single Short Message SMS-PP Data Download is sent to the SIM	1- Applet is not triggered	
		2- A Concatenated Short Messages SMS-PH		
		Data Download is sent to the SIM (compo	osed	
		of 2 Short Messages. The UDL for the fi Short Message is 70 and for the second	irst	
		70).	- 2. Applet is not triggered	
			2- Applet is not triggered	
		An unregognized envelope is sent to the		
		An unrecognized envelope is sent to the sim	=	
		ToolkitRegistry.setEvent() method is		
		called for EVENT_FORMATTED_SMS_PP_ENV		
		3- A Single Short Messages SMS-PP Data		

ld	Description	API/Framework Expectation	APDU Expectation
	Download is sent to the SIM.		
	4- A Concatenated Short Messages SMS-PP Data Download is sent to the SIM (composed	3- Applet is triggered	
	of 2 Short Messages. The UDL for the first		
	Short Message is 70 and for the second		
	<u>70).</u>	4- Applet is triggered	

6.3.3.4.4 Test Coverage

CRR Number	Test Case Number	
CRRN1 (See note)	4	
CRRN1 (See note 1)	<u>1, 2</u>	
CRRN2	2	

NOTE: The security checks are not relevant to the test designed in this test area; they will be checked in subclause 6.3.6.

Note 1: The security checks are not relevant to the test designed in this test area; they will be checked in the "Framework Security Management" section.

6.3.3.5 EVENT_UNFORMATTED_SMS_PP_ENV

Test Area Reference: FWK_APT_EUSE-

6.3.3.5.1 Conformance Requirement

6.3.3.5.1.1 Normal Execution

—CRRN1: The applet isapplets registers are triggered by the EVENT_UNFORMATTED_SMS_PP_ENV once it-has registered to this event and an Unformatted Envelope DataDownLoad is received if no proactive session is ongoing.a Short Message Point to Point (Single or Concatenated) is received by Envelope APDU(s) and is unformatted.

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

6.3.3.5.1.2 Parameters error

No requirements.

6.3.3.5.1.3 Context Errors

No requirements.

6.3.3.5.2 Test Suite Files

Test Script: _____FWK_APT_EUSE_1.scr

Test Applet: FWK_APT_EUSE_1.java

Load Script: FWK_APT_EUSE_1.ldr

Cleanup Script: FWK_APT_EUSE_1.clr

Parameter File: FWK_APT_EUSE_1.java

Load Script: FWK_APT_EUSE_1.ldr

Cleanup Script: FWK APT EUSE 1.clr

Parameter File: FWK_APT_EUSE_1.par

6.3.3.5.3 Test Procedure

	ld	Description	API/Framework Expectation	APDU Expectation
4		Applet registration to- /ENT_UNFORMATTED_SMS_PP_ENV and triggering		
	App:	let is registered to the NT_UNFORMATTED_SMS_PP_ENV and STREET SMS_PP_ENV.		
	1 To	colkit Registry.isEventSet() method is led for EVENT_UNFORMATTED_SMS_PP_ENV is to the SIM.	1- The method returns true 2- Applet is triggered	
	1	Applet registration to EVENT UNFORMATTED SMS PP ENV an triggering	nd	
		Applet is registered to the EVENT_UNFORMATTED_SMS_PP_ENV and EVENT_FORMATTED_SMS_PP_ENV. 1-Toolkit Registry.isEventSet() method i		
		called for EVENT_UNFORMATTED_SMS_PP_ENV 2- A Single and Unformatted SMS-PP Data Download Envelope is sent to the SIM.		
		3- A Concatenated and Unformatted SMS-PD Data Download Envelope is sent to the SI (composed of 2 Short Messages. The UDL the first Short Message is 70 and for the second 70)	IM for 3- Applet is triggered	
2		Applet deregistration		
	cal :	lkit Registry.clearEvent()method is led for EVENT_UNFORMATTED_SMS_PP_ENV	1- Applet isn't triggered	
		Drmatted sms pp envelope is sent to the lkit Registry.setEvent() method is led for EVENT_UNFORMATTED_SMS_PP_ENV		
	2 Ar sent	a Envelope UNFORMATTED_SMS_PP_ENV is	2- Applet is triggered	
	2	Applet deregistration Toolkit Registry.clearEvent()method is called for EVENT_UNFORMATTED_SMS_PP_ENV		
		1- A Single and Unformatted SMS-PP Data Download Envelope is sent to the SIM.	1- Applet isn't triggered	

ld	Description	API/Framework Expectation	APDU Expectation
	2- A Concatenated and Unformatted SMS-PP Data Download Envelope is sent to the SIM (composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the second 70)	2- Applet isn't triggered	
	Applet is triggered by a EVENT_FORMATTED_SMS_PP_ENV		
	Toolkit Registry.setEvent() method is called for EVENT_UNFORMATTED_SMS_PP_ENV		
	3- A Single and Unformatted SMS-PP Data Download Envelope is sent to the SIM.	3- Applet is triggered	
	4- A Concatenated and Unformatted SMS-PP Data Download Envelope is sent to the SIM (composed of 2 Short Messages. The UDL for the first Short Message is 70 and for the	4- Applet is triggered	
	second 70)		

6.3.3.5.4 Test Coverage

CRR Number	Test Case Number	
CRRN1	1, 2	
CRRN2	2	

6.3.3.22 EVENT_FIRST_COMMAND_AFTER_SELECT

Test Area Reference: FWK_APT_EFCA

6.3.3.22.1 Conformance Requirement

6.3.3.22.1.1 Normal Execution

CRRN1: The applet is triggered by the EVENT_FIRST_COMMAND_AFTER_SELECT once it has registered to this event; Upon reception of the first command received by the GSM application after it has been selected, or after the ATR if it is the default application, and before the Status Word of the processed command has been sent back by the GSM application, the toolkit framework shall trigger all the toolkit applets registered to this event.

<u>CRRN2</u>: The applet is not triggered by the <u>EVENT_FIRST_COMMAND_AFTER_SELECT</u> once it has deregistered from this event.

CRRN3: If the first command received by the GSM application is a toolkit applet triggering command (e.g. TERMINAL PROFILE), the toolkit applets registered on the EVENT_FIRST_COMMAND_AFTER_SELECT event shall be triggered first.

6.3.3.22.2	Test Suite Files
Test Script:	FWK_APT_EFCA_1.scr
Test Applet:	FWK_APT_EFCA_1.java
	FWK_APT_EFCA_2.java
	FWK_APT_EFCA_3.java
	FWK_APT_EFCA_4.java

FWK APT EFCA 5.java

Load Script: FWK_APT_EFCA_1.ldr

Cleanup Script: FWK_APT_EFCA_1.clr

Parameter File: FWK_APT_EFCA_1.par

6.3.3.22.3 Test Procedure

ld	<u>Description</u>	API/Framework Expectation	APDU Expectation
<u>1</u>	Applets registration to		
	EVENT_FIRST_COMMAND_AFTER_SELECT		
	and triggering		
	Applet1 is registered to the EVENT_FIRST_COMMAND_AFTER_SELECT		
	Applet2 is registered to the EVENT_PROFILE_DOWNLOAD.		
	Applet3 is registered to EVENT_FORMATTED_SMS_PP_ENV.		
	1-Terminal Profile command is sent to the SIM. Applet1 deregisters from	1- Applet1 is triggered by EVENT_FIRST_COMMAND_AFTE	
	EVENT_FIRST_COMMAND_AFTER_SELECT.	R SELECT	
	2- Applet2 deregisters from EVENT PROFILE DOWNLOAD.	Applet1 finalizes Applet2 is triggered by EVENT PROFILE DOWNLOAD	
		Applet2 finalizes Applet3 is not triggered	
	3-Envelope(SMS-PP-DOWNLOAD) formatted is sent to the SIM	3-Applet3 is triggered.	
	4-Applet3 calls setEvent() on event_ EVENT_FIRST_COMMAND_AFTER_SELECT.		
2	Deregistered applets are not triggered	1-Applet3 is triggered. Applet1 and Applet2 are not triggered.	
	1-Reset then Terminal Profile command is sent to the SIM		
	2-Applet3 calls setEvent() on EVENT_PROFILE_DOWNLOAD.	2-Applet3 finalizes.	
3	Install a 4 th applet registered to EVENT FIRST COMMAND AFTER SELECT and EVENT_PROFILE_DOWNLOAD Applet4 is installed, with the same	1- Applet4 is triggered by EVENT_FIRST_COMMAND_AFTE R_SELECT.	
	priority level as Applet3.	Applet3 is triggered by	
	1-Reset then Terminal Profile command is sent to the SIM	EVENT FIRST COMMAND AFTE R_SELECT.	
		Applet4 is triggered by EVENT_PROFILE DOWNLOAD.	
	Delete all applets.	Applet3 is triggered by EVENT_PROFILE_DOWNLOAD.	
<u>4</u>	Check that the applet is triggered before the first SW is sent.		3-The SETUP MENU proactive command is
	<u> </u>	1	p. Caouto Communa la

ld	<u>Description</u>	API/Framework Expectation	APDU Expectation
	1-Install Applet 5. Applet 5 is registered with two entries in		fetched. There is only one item for Applet5.
	the menu entries list. Applet5 is also registered to EVENT_FIRST_COMMAND_AFTER_SELECT.		
	2-Reset and TERMINAL PROFILE.	2- Applet 5 is triggered	
	3-Applet disables a menu entry.		

[Note: Testing the triggering of an applet upon the first command after select is not possible.]

6.3.3.22.4 Test Coverage

CR Number	Test Case Number	
CRRN1	<u>1,2,3, 4</u>	
CRRN2	<u>3</u>	
CRRN3	<u>1, 4</u>	

6.3.3.23 EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE

Test Area Reference: FWK_APT_EDDA

6.3.3.23.1 Conformance Requirement

6.3.3.23.1.1 Normal Execution

<u>CRRN1</u>: For EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE, the framework shall only trigger the applet registered to this event with the appropriate channel identifier.

CRRN2: The registration to the EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE is effective once the toolkit applet has issued a successful OPEN CHANNEL proactive command, and valid till the first successful CLOSE CHANNEL or the end of card session.

<u>CRRN3</u>: When a Toolkit Applet has sent an OPEN CHANNEL proactive command and received a successful <u>TERMINAL RESPONSE</u>, the framework shall register the received channel identifier for the calling Toolkit Applet.

CRRN4: When a Toolkit Applet has sent a CLOSE CHANNEL proactive command and received a successful TERMINAL RESPONSE, the framework shall release the channel identifier contained in the command. A successful TERMINAL RESPONSE means that the result of the proactive command execution belongs to command performed category (i.e. General Result ='0x').

6.3.3.23.2 Test Suite Files

Test Script: FWK_APT_EDDA_1.scr

Test Applet: FWK_APT_EDDA_1.java

Load Script: FWK_APT_EDDA_1.ldr

Cleanup Script: FWK_APT_EDDA_1.clr

Parameter File: FWK_APT_EDDA_1.par

6.3.3.23.3 Test Procedure

	<u>Description</u>	API/Framework Expectation	APDU Expectation
	Applet registration to		
	EVENT_EVENT_DOWNLOAD_DATA_AVAILAB LE	1- Applet1 is triggered by	
	<u>LC</u>	Unformatted SMS PP envelope.	
	Applet1 is registered to Unformatted SMS		
	PP Envelope. 1- Unformatted SMS PP envelope is sent to		
	the SIM.		
	2- Applet calls setEvent() with the event		
	EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE.		
		2- Applet1 finalizes.	
	3- An envelope Event Download Data	3- Applet1 is not triggered.	
	Available is sent to the SIM Channel Status = 81 00		
	4. Unformatted GMG DD annual and in cont. to		
	4- Unformatted SMS PP envelope is sent to the SIM.	4- Applet1 is triggered by	
	F Application in the company of the	Unformatted SMS PP envelope.	
	5- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init()		
	method.		
			6- OPEN CHANNEL
	6- send() method is called to register to		proactive command is fetched.
	this event.		ICIOTICA.
		7- Applet1 finalizes.	Unsuccessful TERMINAL
		8- Applet1 is not triggered.	RESPONSE of OPEN CHANNEL is sent to the
	8- An envelope Event Download Data Available is sent to the SIM with Channel	o- Applet i is not triggered.	SIM.
	Status = 01 00.		
		O Applett is triggered by	
	9- Unformatted SMS PP envelope is sent to	9- Applet1 is triggered by EVENT_UNFORMATTED_SMS_P	
	the SIM.	P ENV.	
	10- Applet1 builds a proactive command		
	OPEN CHANNEL calling ProactiveHandler.init() method.		
	Floactivenandier.init() method.		
	11- send() method is called to register to		
	this event.	12- Applet1 finalizes.	
			11- OPEN CHANNEL
			proactive command is
			fetched. Successful TERMINAL
			RESPONSE of OPEN
			CHANNEL is sent to the
-	Applet triggering to		SIM with Channel Id = 01.
	EVENT EVENT DOWNLOAD DATA AVAILAB		
	<u>LE</u>		
	1- An envelope Event Download Data		
	Available is sent to the SIM Channel Status = 81 00.		
	Channel Status - 01 UU.	1- Applet1 is triggered.	
1	Applet deregistration to EVENT_EVENT_		
	DOWNLOAD DATA AVAILABLE		
			1- OPEN CHANNEL
	<pre>0- Unformatted SMS PP envelope is sent to the SIM.</pre>	0- Applet1 is triggered.	proactive command is fetched.
	1 Applot1 initializes and sands an ODDY		Successful terminal
	1- Applet1 initializes and sends an OPEN CHANNEL proactive command.		response is sent, with
			channelId=02.
	2- Applet1 builds a CLOSE CHANNEL		l .

ld	Description	API/Framework Expectation	APDU Expectation
	Proactive Command calling ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods. 3- An envelope Event Download Data Available is sent to the SIM. Channel Status = 82 00	3- Applet1 is triggered.	2- CLOSE CHANNEL proactive command is fetched. Unsuccessful TERMINAL RESPONSE of CLOSE CHANNEL is sent to the SIM.
	4- Applet1 builds a CLOSE CHANNEL Proactive Command calling ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods.	5- Applet1 finalizes.	4- CLOSE CHANNEL proactive command is fetched. Successful TERMINAL RESPONSE of CLOSE CHANNEL is sent to the SIM with Channel Id = 02.
4	Applet triggering to EVENT EVENT DOWNLOAD DATA AVAILAB LE 1- An envelope Event Download Data Available is sent to the SIM Channel Status = 82 00.	1- Applet1 is not triggered.	
<u>5</u>	Applet1 not triggered after a reset 0- Applet1 is triggered by an unformatted SMS PP Envelope 1- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init() method. 2- send() method is called to register to this event. 3- isEventSet() method is called. 4- Reset the card. 5- An envelope Event Download Data Available is sent to the SIM Channel Status = 82 00.	3- returns true. 5- Applet1 is not triggered.	1- OPEN CHANNEL proactive command is fetched. 2- Successful TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM with Channel Id = 02.

6.3.3.23.4 Test Coverage

CR Number	Test Case Number
CRRN1	<u>2</u>
CRRN2	<u>1, 4, 5</u>
CRRN3	<u>1</u>
CRRN4	<u>3</u>

6.3.3.24 EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS

Test Area Reference: FWK APT EDCS

6.3.3.24.1 Conformance Requirement

6.3.3.24.1.1 Normal Execution

<u>CRRN1:</u> For EVENT EVENT DOWNLOAD CHANNEL STATUS, the framework shall only trigger the applet registered to this event with the appropriate channel identifier.

CRRN2: The registration to the EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS is effective once the toolkit applet has issued a successful OPEN CHANNEL proactive command, and valid till the first successful CLOSE CHANNEL or the end of the card session.

<u>CRRN3</u>: When a Toolkit Applet has sent an OPEN CHANNEL proactive command and received a successful TERMINAL RESPONSE, the framework shall register the received channel identifier for the calling Toolkit Applet.

CRRN4: When a Toolkit Applet has sent a CLOSE CHANNEL proactive command and received a successful TERMINAL RESPONSE, the framework shall release the channel identifier contained in the command. A successful TERMINAL RESPONSE means that the result of the proactive command execution belongs to command performed category (i.e. General Result ='0x').

6.3.3.24.2	Test Suite Files		
Test Script:	FWK APT EDCS 1.scr		
Test Applet:	FWK_APT_EDCS_1.java		
Load Script:	FWK_APT_EDCS_1.ldr		
Cleanup Script:	FWK_APT_EDCS_1.clr		
Parameter File:	FWK_APT_EDCS_1.par		

6.3.3.24.3 Test Procedure

Description	API/Framework Expectation	APDU Expectation
Applet registration to EVENT_EVENT_DOWNLOAD_CHANNEL_STAT	1. Applet1 is triggered by	
US	Unformatted SMS PP envelope	
Applet1 is registered to Unformatted SMS PP Envelope. 1-Unformatted SMS PP envelope is sent to the SIM. 2-The applet calls setEvent() with EVENT_EVENT_DOWNLOAD_CHANNEL_STATUS.	2- Applet1 finalizes.	6- OPEN CHANNEL proactive command is fetched. Unsuccessful TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM.
3- An envelope Event Download Channel Status is sent to the SIM. Channel Status = 81 00 4-Unformatted SMS PP envelope is sent to the SIM. 5- Applet1 builds a proactive command OPEN	3- Applet1 is not triggered. 4- Applet1 is triggered by Unformatted SMS PP envelope.	11- OPEN CHANNEL proactive command is fetched. Successful TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM with Channel Id = 01.
CHANNEL calling ProactiveHandler.init() method. 6- send() method is called to register to this event. 8- An envelope Event Download Data Available is sent to the SIM with Channel Status = 01 00.	7- Applet finalizes. 8- Applet1 is not triggered.	
9- Unformatted SMS PP envelope is sent to the SIM. 10- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init() method. 11- send() method is called to register to	9- Applet1 is triggered by EVENT_UNFORMATTED_SMS_P P_ENV.	
Applet triggering to EVENT_EVENT_DOWNLOAD_CHANNEL STATUS	12- Applet1 finalizes.	
1- An envelope Event Download Channel Status is sent to the SIM. Channel Status = 81 00	1- Applet1 is triggered.	
Applet deregistration to EVENT EVENT DOWNLOAD CHANNEL STATUS 0- Unformatted SMS PP envelope is sent to the SIM. 1-Applet1 initializes and sends an OPEN CHANNEL proactive command.	0- Applet1 is triggered.	OPEN CHANNEL proactive command is fetched. Successful terminal response is sent, with channelld=02.
2- Applet1 builds a CLOSE CHANNEL Proactive Command calling ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods.	3- The applet is triggered.	2-CLOSE CHANNEL proactive command is fetched. Unsuccessful TERMINAL

ld	Description	API/Framework Expectation	APDU Expectation
	3-An envelope Event Download Channel Status is sent to the SIM. Channel Status = 82 00 4- Applet1 builds a Close Channel Proactive Command calling ProactiveHandler.initCloseChannel() and ProactiveHandler.send() methods.	5- Applet1 finalizes.	RESPONSE of CLOSE CHANNEL is sent to the SIM. 4- CLOSE CHANNEL proactive command is fetched. Successful TERMINAL RESPONSE of CLOSE CHANNEL is sent to the SIM with Channel Id = 02.
<u>4</u> <u>5</u>	Applet triggering to EVENT EVENT DOWNLOAD CHANNEL STATUS 1- An envelope Event Download Channel Status is sent to the SIM. Channel Status = 82 00 Applet1 not triggered after a reset	Applet1 is not triggered.	
	0- Applet1 is triggered by an unformatted SMS PP Envelope. 1- Applet1 builds a proactive command OPEN CHANNEL calling ProactiveHandler.init() method. 2- send() method is called to register to this event. 3- isEventSet() method is called. 4- Reset the card. 5- An envelope Event Download Data Available is sent to the SIM Channel Status = 82 00.	3- returns true. 5- Applet1 is not triggered.	1- OPEN CHANNEL proactive command is fetched. 2- Successful TERMINAL RESPONSE of OPEN CHANNEL is sent to the SIM with Channel Id = 02.

6.3.3.24.4 Test Coverage

CR Number	Test Case Number
CRRN1	<u>2</u>
CRRN2	<u>1, 4, 5</u>
CRRN3	<u>1</u>
CRRN4	3

6.3.3.25 EVENT FORMATTED SMS PP UPD

Test Area Reference: FWK_APT_EFSU

6.3.3.25.1 Conformance Requirement

6.3.3.25.1.1 Normal Execution

CRRN1: The applet is triggered by the EVENT_FORMATTED_SMS_PP_UPD once:

- it has been registered to this event,
- a Short Message Point to Point (Single or Concatenated) is received by Update Record EFsms APDU(s) and is formatted according to TS 23.048 [8],
- the toolkit applet to be triggered is registered with the corresponding TAR in the SMS TPDU,

<u>CRRN2</u>: The applets are not triggered by the EVENT FORMATTED SMS PP UPD once it has deregistered from this event.

6.3.3.25.2	Test Suite Files		
Test Script:	FWK_APT_EFSU_1.scr		
Test Applet:	FWK_APT_EFSU_1.java		
Load Script:	FWK_APT_EFSU_1.ldr		
Cleanup Script:	FWK_APT_EFSU_1.clr		
Parameter File:	FWK_APT_EFSU_1.par		

6.3.3.25.3 Test Procedure

6.3.3.25.4 Test Coverage

<u>ld</u>	Desc	<u>cription</u>	API/Framework Expectation	APDU Expectation
1	Applet regist	ration to EVENT		
	FORMATTED_SMS_	PP_UPD and triggering		
	Applet is registered EVENT_FORMATTED_SMS_			
	EVENT_UNRECOGNIZED_E			
	1 marallala production	in Process (Control of the		
	called for EVENT_FOR	isEventSet() method is MATTED_SMS_PP_UPD	1- The method returns true.	
		nt to Point Single and		
	Formatted is received EFsms APDU.	d by Update Record		
	Ersins Arbo.		2- Applet is triggered.	
	3. Short Message Poi	nt to Doint		
	Concatenated Formatt			
	Update Record EFsms			
	Concatenated Message Short Messages. The		3- Applet is triggered on reception	
	Short Message is 70		of the last concatenated SMS	
	70).			
<u>2</u>	Applet de	eregistration eregistration		
	ToolkitRegistry.clea			
	called for EVENT_FOR	MATTED_SMS_PP_UPD		
		nt to Point Single and		
	Formatted is received EFsms APDU.	ed by Update Record		
	Ersiis Ardo.		1- Applet is not triggered	
	2. Short Message Poi	nt to Point		
	Concatenated and For	matted is received by		
	Update Record EFsms Concatenated Message			
	Short Messages. The		2- Applet is not triggered	
	Short Message is 70	and for the second		
	70).			
	An unrecognized enve	elope is sent to the		
		1. mm (1.7. mm 1.3 2 1.		
	ToolkitRegistry.setEcalled for EVENT_FOR		3- Applet is triggered	
	3. Short Message Poil Formatted is receive	nt to Point Single and do by Update Record		
	EFsms APDU.			
	4. Short Message Poi	nt to Point	4- Applet is triggered on reception	
	Concatenated Formatt	ed is received by	of the last concatenated SMS.	
	Update Record EFsms Concatenated Message			
	Short Messages. The	UDL for the first		
	Short Message is 70 70).	and for the second		
	/ · / ·			
		CRR Number	Test Case Number	
<u> </u> 		CRRN1 (See note1) CRRN2	1,2	
-		<u>UNRINZ</u>	<u> </u>	

Note 1: The security checks are not relevant to the test designed in this test area; they will be checked in the "Framework Security Management" section.

6.3.3.26 EVENT_UNFORMATTED_SMS_PP_UPD

Test Area Reference: FWK_APT_EUSU

6.3.3.26.1 Conformance Requirement

6.3.3.26.1.1 Normal Execution

CRRN1: The applets registers are triggered by the EVENT_UNFORMATTED_SMS_PP_UPD once a Short Message Point to Point (Single or Concatenated) is received by Update Record EFsms APDU(s) and is unformatted.

<u>CRRN2</u>: The applets are not triggered by the <u>EVENT_UNFORMATTED_SMS_PP_UPD</u> once it has deregistered from this event.

6.3.3.26.2 Test Suite Files

Test Script: FWK APT EUSU 1.scr

Test Applet: FWK_APT_EUSU_1.java

Load Script: FWK_APT_EUSU_1.ldr

Cleanup Script: FWK_APT_EUSU_1.clr

Parameter File: FWK_APT_EUSU_1.par

6.3.3.26.3 Test Procedure

<u>ld</u>	<u>Description</u>	API/Framework Expectation	APDU Expectation
1	Applet registration to EVENT		
	UNFORMATTED_SMS_PP_UPD and triggering		
	Applet is registered to EVENT_UNFORMATTED_SMS_PP_UPD and		
	EVENT_UNRECOGNIZED_ENVELOPE		
	1. Toolkit Registry.isEventSet() method is		
	called for EVENT_UNFORMATTED_SMS_PP_UPD	1- Applet is not triggered	
	2. Short Message Point to Point Single and		
	Unformatted is received by Update Record EFsms APDU	2- Applet is triggered.	
	EFSIIIS APDU		
	3. Short Message Point to Point		
	Concatenated and Unformatted is received	3- Applet is triggered on reception	
	by Update Record EFsms APDU (The Concatenated Message is composed of 2	of the last concatenated SMS.	
	Short Messages. The UDL for the first		
	Short Message is 70 and for the second		
	70).		
2	Applet deregistration		
	ToolkitRegistry.clearEvent() method is		
	called for EVENT_UNFORMATTED_SMS_PP_UPD		
	1. Short Message Point to Point Single and		
	Unformatted is received by Update Record EFsms APDU		
	EFSIIIS APDU	- Applet is not triggered	
	Charle Mannage Balan to Balan		
	2. Short Message Point to Point Concatenated and Unformatted is received		
	by Update Record EFsms APDU(s) (The		
	Concatenated Message is composed of 2 Short Messages. The UDL for the first	2- Applet is not triggered.	
	Short Message is 70 and for the second		
	70).		
	An unrecognized envelope is sent to the		
	sim		
	ToolkitPegistry gotFyont() method is		
	ToolkitRegistry.setEvent() method is called for EVENT_UNFORMATTED_SMS_PP_UPD	3- Applet is triggered	
	2 Chart Manager Balance Science 2		
	3. Short Message Point to Point Single and Unformatted is received by Update Record	-	
	EFsms APDU		
	4. Short Message Point to Point	4- Applet is triggered on reception	
	Concatenated and Unformatted is received	of the last concatenated SMS	
	by Update Record EFsms APDU(s) (The Concatenated Message is composed of 2		
	Short Messages. The UDL for the first		
	Short Message is 70 and for the second		
	70).		
L	1	1	

6.3.3.26.4 Test Coverage

CRR Number	Test Case Number
CRRN1	<u>1,2</u>
CRRN2	<u>2</u>

6.3.4 Proactive Command Sending by the STF

6.3.4.3 Proactive Command Control

Test Area Reference: FWK_PCS_PCCO

6.3.4.1.1 Conformance Requirements

6.3.4.1.1.1 Normal Execution

CRRN1: The SIM Toolkit Framework shall prevent the toolkit applet to issue the following proactive commands:

SET UP MENU, SET UP EVENT LIST, POLL INTERVAL, POLLING OFF. If an applet attempts to issue such a command, the SIM Toolkit Framework shall throw an exception.

CRRN2: The SIM Toolkit Framework shall prevent a toolkit applet to issue a TIMER MANAGEMENT proactive command using a timer identifier, which is not allocated to it. If an applet attempts to issue such a command, the SIM Toolkit Framework shall throw an exception.

CRRN3: The SIM Toolkit Framework shall prevent a toolkit applet to issue a SEND DATA, RECEIVE DATA and CLOSE CHANNEL proactive commands using a channel identifier, which is not allocated to it. If an applet attempts to issue such a command the SIM Toolkit Framework shall throw an exception.

CRRN4: The SIM Toolkit Framework shall prevent a toolkit applet to issue an OPEN CHANNEL proactive command if it exceeds the maximum number of channel allocated to this applet. If an applet attempts to issue such a command the SIM Toolkit Framework shall throw an exception.

6.3.4.1.2 Test Suite Files

Test Script: FWK PCS PCCO 1.scr

Test Applet: FWK PCS PCCO 1.java

FWK PCS PCCO 2.java

FWK PCS PCCO 3.java

Load Script: FWK PCS PCCO 1.ldr

Cleanup Script: FWK PCS PCCO 1.clr

Parameter File: FWK PCS PCCO 1.par

6.3.4.1.3 Test Procedure

ld	<u>Description</u>	API/Framework Expectation	APDU Expectation
0	Applets installation		
	Applet1 is installed with 4 timers maximum, 0		
	channel maximum and 1 menu.		
	Applet2 is installed with 8 timers maximum, 3		
	<u>channels maximum.</u>		
	Applet3 is installed with 1 channel maximum.		
<u>1</u>	STK Proactive Commands		
			1- 90 00 (no proactive
	1- Send a formatted envelope with the TAR of	1- Applet1 is triggered	command is sent)
	Applet1		
	2- Applet1 builds and sends a SET UP MENU	2- COMMAND_NOT_ALLOWED	
	proactive command	toolkit exception is thrown	

ld	Description	API/Framework Expectation	APDU Expectation
<u>IQ</u>	3- Applet1 builds and sends a SET UP EVENT	3- COMMAND_NOT_ALLOWED	APDU Expectation
	LIST proactive command	toolkit exception is thrown	
	4- Applet1 builds and sends a POLL INTERVAL	4- COMMAND NOT ALLOWED	
	proactive command	toolkit exception is thrown 5- COMMAND_NOT_ALLOWED	
	5- Applet1 builds and sends a POLLING OFF proactive command	toolkit exception is thrown	
2	TIMER MANAGEMENT Proactive command	toonat oxooption to timown	
_			
	1- Send a formatted envelope with the TAR of	1- Applet2 is triggered	
	Applet2 2- Applet2 allocates 8 timers by calling	2- No exception is thrown	
	allocateTimer() method and release the 3 timers	2- No exception is thrown	
	from id 1 to 3.		
	3- Send a formatted envelope with the TAR of	3- Applet1 is triggered	
	Applet1		
	4- Applet1 allocates 3 timers (Id 1 to 3) by calling allocateTimer() method 3 times	4- No exception is thrown	
	5- Send a formatted envelope with the TAR of	5- Applet2 is triggered	
	Applet2	5- Appletz is triggered	
	6- Applet2 releases timers of Id 4 to 7	6- No exception is thrown	
	7- Send a formatted envelope with the TAR of	7- Applet1 is triggered	
	Applet1	9. No expention is thrown	
	8- For each of the 3 timers allocated by Applet1 (Id 1to 3) a TIMER MANAGEMENT proactive session	o- NO exception is thrown	
	is performed		
	9- For other timers (Id 4 to 8), Applet1 builds and	9- COMMAND NOT ALLOWED	
	sends a TIMER MANAGEMENT proactive	toolkit exception is thrown	
	<u>command</u>		
			8- 3 TIMER MANAGEMENT
			proactive commands are
			<u>fetched</u>
			9- The Status word of the
			last previous Terminal Response is 90 00 (no more
			Response is 90 00 (no more
			proactive command is sent)
3	No Channel allowed		proactive command is sent)
3			
3	1- Send a formatted envelope with the TAR of	1- Applet1 is triggered	1- 90 00 (no proactive
3	1- Send a formatted envelope with the TAR of Applet1		
3	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN	2- COMMAND_NOT_ALLOWED	1- 90 00 (no proactive
3	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED	1- 90 00 (no proactive
3	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown	1- 90 00 (no proactive
3	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED	1- 90 00 (no proactive
3	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown	1- 90 00 (no proactive
3	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown	1- 90 00 (no proactive
3	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED	1- 90 00 (no proactive
	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown	1- 90 00 (no proactive
3	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED	1- 90 00 (no proactive
	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown	1- 90 00 (no proactive
	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command 4 Channels allowed 1- Send a formatted envelope with the TAR of Applet3	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 1- Applet3 is triggered	1- 90 00 (no proactive
	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command 4 Channels allowed 1- Send a formatted envelope with the TAR of Applet3 2- Applet3 builds and sends a CSD OPEN	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown	1- 90 00 (no proactive
	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command 4 Channels allowed 1- Send a formatted envelope with the TAR of Applet3 2- Applet3 builds and sends a CSD OPEN CHANNEL proactive command	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 1- Applet3 is triggered	1- 90 00 (no proactive command is sent)
	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command 4 Channels allowed 1- Send a formatted envelope with the TAR of Applet3 2- Applet3 builds and sends a CSD OPEN CHANNEL proactive command 3- Send a Fetch and Terminal Response OK on	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 1- Applet3 is triggered	1- 90 00 (no proactive command is sent) 2- 91 1C 3- OPEN CHANNEL
	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command 4 Channels allowed 1- Send a formatted envelope with the TAR of Applet3 2- Applet3 builds and sends a CSD OPEN CHANNEL proactive command	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 1- Applet3 is triggered	1- 90 00 (no proactive command is sent)
	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command 4 Channels allowed 1- Send a formatted envelope with the TAR of Applet3 2- Applet3 builds and sends a CSD OPEN CHANNEL proactive command 3- Send a Fetch and Terminal Response OK on channel 7	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 1- Applet3 is triggered 2- No exception is thrown	1- 90 00 (no proactive command is sent) 2- 91 1C 3- OPEN CHANNEL
	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command 4 Channels allowed 1- Send a formatted envelope with the TAR of Applet3 2- Applet3 builds and sends a CSD OPEN CHANNEL proactive command 3- Send a Fetch and Terminal Response OK on channel 7	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 1- Applet3 is triggered	1- 90 00 (no proactive command is sent) 2- 91 1C 3- OPEN CHANNEL
	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command 4 Channels allowed 1- Send a formatted envelope with the TAR of Applet3 2- Applet3 builds and sends a CSD OPEN CHANNEL proactive command 3- Send a Fetch and Terminal Response OK on channel 7 4- Send a formatted envelope with the TAR of Applet2	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 1- Applet3 is triggered 2- No exception is thrown	2- 91 1C 3- OPEN CHANNEL proactive
	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command 4 Channels allowed 1- Send a formatted envelope with the TAR of Applet3 2- Applet3 builds and sends a CSD OPEN CHANNEL proactive command 3- Send a Fetch and Terminal Response OK on channel 7	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 1- Applet3 is triggered 2- No exception is thrown	1- 90 00 (no proactive command is sent) 2- 91 1C 3- OPEN CHANNEL
	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command 4 Channels allowed 1- Send a formatted envelope with the TAR of Applet3 2- Applet3 builds and sends a CSD OPEN CHANNEL proactive command 3- Send a Fetch and Terminal Response OK on channel 7 4- Send a formatted envelope with the TAR of Applet2 5- Applet2 builds and sends a CSD OPEN	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 1- Applet3 is triggered 2- No exception is thrown	2- 91 1C 3- OPEN CHANNEL proactive
	1- Send a formatted envelope with the TAR of Applet1 2- Applet1 builds and sends a CSD OPEN CHANNEL proactive command 3- Applet1 builds and sends a GPRS OPEN CHANNEL proactive command 4Applet1 builds and sends a SEND DATA proactive command 5- Applet1 builds and sends a RECEIVE DATA proactive command 6- Applet1 builds and sends a CLOSE CHANNEL proactive command 4 Channels allowed 1- Send a formatted envelope with the TAR of Applet3 2- Applet3 builds and sends a CSD OPEN CHANNEL proactive command 3- Send a Fetch and Terminal Response OK on channel 7 4- Send a formatted envelope with the TAR of Applet2 5- Applet2 builds and sends a CSD OPEN CHANNEL proactive command	2- COMMAND_NOT_ALLOWED toolkit exception is thrown 3- COMMAND_NOT_ALLOWED toolkit exception is thrown 4- COMMAND_NOT_ALLOWED toolkit exception is thrown 5- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 6- COMMAND_NOT_ALLOWED toolkit exception is thrown 1- Applet3 is triggered 2- No exception is thrown	2- 91 1C 3- OPEN CHANNEL proactive 5- 91 1C

<u>ld</u>	<u>Description</u>	API/Framework Expectation	APDU Expectation
	7- Applet2 builds and sends a GPRS OPEN	7- No exception is thrown	7- 91 17
	CHANNEL proactive command		
	8- Send Fetch and Terminal Response OK on		8- OPEN CHANNEL
	channel 2		proactive command is
			fetched, SW = 91 1C on the
			Terminal Response
	9- For each channel id from 3 to 7, Applet2 builds	9- COMMAND NOT ALLOWED	
	and sends a SEND DATA proactive command	toolkit exception is thrown	
	10- For each channel id from 3 to 7, Applet2 builds	10- COMMAND_NOT_ALLOWED	
		toolkit exception is thrown	
	11- For each channel id from 3 to 7, Applet2 builds	11- COMMAND_NOT_ALLOWED	
	and sends a CLOSE CHANNEL proactive	toolkit exception is thrown	
	command		
	12- Applet2 builds and sends a CSD OPEN	12- No exception is thrown	
	CHANNEL proactive command		
	13- Fetch and Terminal Response OK on channel		13- OPEN CHANNEL
	<u>3</u>		proactive command is
			<u>fetched</u>
	14- Applet2 builds and sends an OPEN CHANNEL	14- COMMAND_NOT_ALLOWED	14- 90 00 expected to the
	proactive command	toolkit exception is thrown	previous Terminal
			Response (no proactive
			command is sent)

6.3.4.1.4 Test Coverage

CRR number	Test case number
<u>N1</u>	<u>1</u>
<u>N2</u>	<u>2</u>
<u>N3</u>	<u>3,4</u>
<u>N4</u>	<u>3,4</u>

6.3.6 Framework Security Management

Security Parameters

The table that follows contains the security parameters that shall be used when the 03.48 TS 23.048 [8]-security is required in the test cases developed in the current section.

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

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

6.3.6.1 Input Data

Test Area Reference: FWK_FWS_INDA

6.3.6.1.1 Conformance Requirements

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 <u>TS 23.048 [8]</u>, 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 TS 23.048 [8], the SIM Toolkit Framework shall verify the security of the cell broadcast page.

<u>CRRN4</u>: If the SIM receives an <u>Update Record EFsms instruction formatted according to TS 23.048[8], the SIM Toolkit Framework shall verify the security of the SMS.</u>

CRRN5: The STF shall provide the input data deciphered.

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

ld	Description	API/Framework Expectation	APDU Expectation
4	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:	The applet is triggered.	
	Cryptographic checksum;		
	Data = 01		
	<u>1d</u>	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:	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:

ld	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	1- The applet 1 is triggered and the	
	sent to the SIM with this features: Ciphering;	value integrity is checked.	Envelope with status word
	Cryptographic checksum;		9000
	No proof of receipt;		
	TAR of Applet 1;		
	<u>Data = 01</u>		
	O Charle Manner and the state of an i		
	2- Short Message concatenated and formatted is sent to the SIM by an		
	Envelope (SMS PP) with these features:		2- The SIM answers to the
	Ciphering;	2- The applet 1 is triggered and the	
	Cryptographic checksum;	value integrity is checked	9000
	No proof of receipt;		
	TAR of Applet 1;		
	Data length is 150.		
_	Triangular to different contets with different		
2	Triggering two different applets with different security		
	Security		
	Applet2 is installed		
	1 Envelope(SMS PP) formatted is sent to	1- Applet1 is triggered	
	the SIM with this features:	, Applet is triggered	1- The SIM answers to the
	Ciphering;		Envelope with status word
	Cryptographic checksum; No proof of receipt;		9000
	TAR of Applet 1		
	Data = 02		
	2 Envelope(SMS PP) formatted is sent to		
	the SIM with this features:	2- Applet2 is triggered	
	No ciphering;	30000	2- The SIM answers to the
	Cryptographic checksum;		Envelope with status word
	TAR of Applet 2		9000
	Data = 03		
3	Envelope(SMS-PP) formatted with wrong	No applet is triggered	1- The SIM answers to the
	cryptographic checksum		Envelope with status word
			9000
	No ciphering;		
	Wrong Cryptographic checksum;		
	No proof of receipt; TAR of Applet 1		
	TAR of Applet I Data = 04		
	Data - 01	1	1

ld	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	1- Applet 1 is triggered and the value integrity is checked	1- The SIM answers to the Envelope with status words 9000
	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	2- Applet 1 is triggered and the value integrity is checked	2- The SIM answers to the Envelope with status words 9000
	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	3- Applet 2 is triggered and the	3- The SIM answers to the Envelope with status words 9000
	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	value integrity is checked	4- The SIM answers to the Envelope with status words 9000
	Data length = 150.	4- Applet 2 is triggered and the value integrity is checked	
3	Envelope(SMS-PP) formatted with wrong cryptographic checksum		1- The SIM answers to the Envelope with status words 9000
	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	1- No applet is triggered.	
	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	2- No applet is triggered.	

ld	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;	1- Applet3 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 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 Data = 02	1- Applet3 is triggered	1- The SIM answers to the Envelope with status words
	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 Data = 03	2- Applet4 is triggered	2- The SIM answers to the Envelope with status word 9000

ld	Description	API/Framework Expectation	APDU Expectation
5	Triggering two different applets with different		7.1. 2.0 <u>2.1.p.</u> 00.tu
	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 Data = 02	1- Applet3 is triggered and the value integrity is checked	1- The SIM answers to the Envelope with status words 9000
	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 Data = 03	2- Applet4 is triggered and the value integrity is checked	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:	No applet is triggered	The SIM answers to the Envelope with status words-9000
	TAR of Applet 3 Data = 01		
<u>6</u>	Envelope(SMS-CB) formatted with wrong cryptographic checksum No ciphering; Wrong Cryptographic checksum; No proof of receipt;	No applet is triggered	1- The SIM answers to the Envelope with status words 9000
	TAR of Applet 3 Data = 04		
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 applet5 is triggered and the value integrity is checked. 2- The 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:	1- Applet5 is triggered and the value integrity is checked.	1- The SIM answers to the Update Record EFsms instruction with status words
	Ciphering; Cryptographic checksum;		9000

ld	Description	API/Framework Expectation	APDU Expectation
	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;	2- Applet5 is triggered and the value integrity is checked.	2- The SIM answers to the Update Record EFsms instruction with status words 9000
	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	3- Applet6 is triggered and the value integrity is checked.	3- The SIM answers to the Update Record EFsms instruction with status words 9000
	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.	4- Applet6 is triggered and the value integrity is checked.	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 instruction with these features:No ciphering; Wrong Cryptographic checksum; No proof of receipt; TAR of Applet5 Data = 07	1- No applet is triggered.	1- The SIM answers to the Update Record EFsms instruction with status words 9000
	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	2- No applet is triggered.	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	
CRRN2	<u>3,6,9</u>	
CRRN3	4,5,6	
CRRN4	<u>7,8,9</u>	
CRRN5	1,2,4,5,7,8	

6.3.7 Envelope Response Posting

6.3.7.4 EVENT_FORMATTED_SMS_PP_ENV

Test Area Reference: FWK_ERP_EFSE

6.3.7.4.1 Conformance Requirement

6.3.7.4.1.1 Normal Execution

<u>CRRN1</u>: If PoR is required a SMS-DELIVER REPORT is sent by the SIM, when the post() or the postAsBERTLV() method is invoked and if bit 6 of the second octet of SPI is set to 0.

CRRN2: If PoR is required a SMS-SUBMIT is sent by the SIM, when the post() or the postAsBERTLV() method is invoked and if bit 6 of the second octet of SPI is set to 1. In this case the statusType method parameter is meaningless. The SIM Toolkit Framework shall build and issue a Send Short Message proactive command as defined in TS 11.14 [4].

6.3.7.4.2	Test Suite Files
Test Script:	FWK_ERP_EFSE_1.scr
Test Applet:	FWK_ERP_EFSE_1.java
	FWK_ERP_EFSE _2.java
Load Script:	FWK_ERP_EFSE _1.ldr
Cleanup Script:	FWK_ERP_EFSE_1.clr
Parameter File:	FWK ERP EFSE 1.par

6.3.7.4.3 Test Procedure

<u>ld</u>	<u>Description</u>	API/Framework Expectation	APDU Expectation
1	SMS DELIVER REPORT		
	1- A formatted sms pp envelope with SMS Deliver Report required is sent to the SIM with bit 6 of SPI2 set to 0.	1- Applet1 is triggered	
	2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	2- No exception is thrown.	
	3- Applet1 builds the answer and calls the post() method with StatusType=SW1_RP_ACK	Applet1 finalizes	3- ME receives 9FXX and checks the response
	4- A formatted sms pp envelope with SMS Deliver Report required is sent to the SIM with bit 6 of SPI2 set to 0.		
	5- EnvelopeResponseHandler.getTheHandler() method is called by Applet1 6- Applet1 builds the answer and calls the postAsBERTLV() method with	4- Applet1 is triggered	5- ME receives 9FXX and checks the response
	StatusType=SW1_RP_ACK	5- No exception is thrown.	
		Applet1 finalizes	
2	SMS-SUBMIT 1- A formatted sms pp envelope with SMS Submit required is sent to the SIM with bit 6 of SPI2 set to 1.	1- Applet1 is triggered	
	2- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	2- No exception is thrown.	
	3-Applet1 builds the answer and calls the post() method with StatusType=SW1_RP_ACK	Applet1 finalizes	3- ME receives a Send Short Message proactive command.
	4- A formatted sms pp envelope with SMS Submit required is sent to the SIM with bit 6 of SPI2 set to 1.	4- Applet1 is triggered	
	5- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	5- No exception is thrown	

	<u>Description</u>	API/Framework Expectation	APDU Expectation
<u>6</u>	G-Applet1 builds the answer and calls the bost() method with StatusType=SW1_RP_ERROR	Applet1 finalizes	6- ME receives a Send Short Message proactive command.
<u>s</u>	7- A formatted sms pp envelope with SMS Submit required is sent to the SIM with Dit 6 of SPI2 set to 1. 8- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	7- Applet1 is triggered	
r	O-Applet1 builds the answer and calls the costAsBERTLV() method with StatusType=SW1_RP_ACK	8- No exception is thrown.	9- ME receives a Send Short Message proactive command.
	10- A formatted sms pp envelope with SMS	Applet1 finalizes	
h 1	Submit required is sent to the SIM with pit 6 of SPI2 set to 1. L1- EnvelopeResponseHandler.getTheHandler() method is called by Applet1	10- Applet1 is triggered	
<u>1</u>	12-Applet1 builds the answer and calls the costAsBERTLV () method with StatusType=SW1_RP_ERROR	11- No exception is thrown.	12- ME receives a Send Short Message proactive command.

6.3.7.4.4 Test Coverage

CRR Number	Test Case Number
CRRN1	<u>1</u>
CRRN2	2

6.3.8 Toolkit Installation

6.3.8.1 Timers Allocation

Test Area Reference: FWK_TIN_TMAL-

6.3.8.1.1 Conformance Requirements

6.3.8.1.1.1 Normal execution

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

6.3.8.1.1.2 Parameters error

No requirements.

6.3.8.1.1.3 Context errors

—CRRC1_: Allocated timers shall not exceed the maximum number of timers allowed for this applet instance defined during installation.

—CRRC2_: The total number of timers allocated for all the applets shall not exceed 8. If the maximum number of timers required is greater than '08' (maximum numbers of timers specified in TS 11.14 [4], the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

6.3.8.1.2 Test suite files

Test Script:	FWK_TIN_TMAL_1.scr
Test Applet:	FWK_TIN_TMAL_1.java
	FWK_TIN_TMAL_2.java
	FWK_TIN_TMAL_3.java
Load Script:	FWK_TIN_TMAL_1.ldr
Cleanup Script:	FWK_TIN_TMAL_1.clr
Parameter File:	FWK TIN TMAL 1.java
	FWK TIN TMAL 2.java
	FWK_TIN_TMAL_3.java
Load Script:	FWK_TIN_TMAL_1.ldr
Cleanup Script:	FWK TIN TMAL 1.clr
Parameter File:	FWK_TIN_TMAL_1.par

6.3.8.1.3 Test Procedure

ld	Description		API/Framework Expectation	n	APDU Expectation
	More than 8 timers at the		,		•
in	nstantiation of applet1: check				
	at applet1 is not installed or				
	that it is not possible to				
	allocate more than 8 timers.			#	ne SIM answers to the
				Env	elope with status words 90-00
Ins	tall for install of applet1 with- imum 9 timers allocated.				50 00
max:		Sha	Il throw a ToolkitException with	4	2 behaviours may be
app:			on NO TIMER AVAILABLE	-	expected :
	33		on the 9 th allocateTimer()	1	. applet1 is not found,
		o,	on the commence ()		status word 6X XX
					2. applet1 has been
	lot1 is solosted				talled and only 8 timers
app .	let1 is selected				are allocated
1	More than 8 timers at the instantiation of	F			a. 5 a 5 a 5
<u> </u>	applet1: check that applet1 is not installed				
	Install for install of applet1 with				
	maximum 9 timers allocated, requesting	a			The SIM answers to the
	POR to be sent via SMS-DELIVER-REPORT.				Envelope with status words
					9Fxx
					A GET RESPONSE is sent
					and the additional data in
					the PoR is checked. It must
					be 01 6A 80.
R	leset the card and delete instance of applet1				
	Reset the card				
2	Good installation of applet2				
	Install for install of applet2 (maximum timers allocated).	4			The SIM answers to the Envelope with status words 90 00
3	Allocate 4 timers Applet2		No exception shall be thrown.		
4	Allocate one more timer Applet2		Shall throw a ToolkitException reason NO_TIMER_AVAILABL		
5	Good installation of applet3				
	Install for install of applet3 (maximum timers allocated).	. 8			The SIM answers to the Envelope with status words 90 00
6	Allocate 4 timers Applet3		No exception shall be thrown.		
7	Allegate and mark Corre		Chall throws - T1125	; 41	
7	Allocate one more timer Applet3		Shall throw a ToolkitException reason NO_TIMER_AVAILABL		
8	Check that each timerId (allocated by applet and applet3) is between 1 and 8 and is differ from each other				

6.3.8.1.4 Test Coverage

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

6.3.8.8 Channel Allocation

Test Area Reference: FWK_TIN_CHAL

6.3.8.7.1 Conformance Requirements

6.3.8.7.1.1 Normal execution

<u>CRRN1</u>: One toolkit applet can register to several channels, but a channel can only be allocated to one toolkit <u>applet</u>.

6.3.8.7.1.2 Context errors

CRRC1: Allocated channels shall not exceed the maximum number of channels allowed for this applet instance.

CRRC2: The total number of channels allocated for all the applets shall not exceed 7. If the maximum number of channels required is greater than '07' (maximum numbers of channels specified in TS 11.14 [4]), the card shall return the Status Word '6A80', incorrect parameters in data field, to the Install(Install) command.

6.3.8.7.2 Test suite files

Test Script:	FWK TIN CHAL 1.scr
Test Applet:	FWK_TIN_CHAL_1.java
	FWK_TIN_CHAL_2.java
	FWK_TIN_CHAL_3.java
Load Script:	FWK_TIN_CHAL_1.ldr
Cleanup Script:	FWK_TIN_CHAL_1.clr
Parameter File:	FWK_TIN_CHAL_1.par

6.3.8.7.3 Test Procedure

ld	<u>Description</u>	API/Framework Expectation	APDU Expectation
1	More than 7 channels at the instantiation of applet1: check that applet1 is not installed 1-Install for install of applet1 with maximum 8 channels allocated. A POR is asked to be sent via SMS-DELIVER-REPORT.	THE PROPERTY OF THE PROPERTY O	1- The SIM answers to the Envelope with status words 9Fxx. A GET RESPONSE is sent and the additional data in the PoR is checked. It must be 01 6A 80.
	Reset the card		

		1	
2	Good installation of applet2		The SIM answers to the
	Install for install of applet2 (maximum 4		Envelope with status words
	<u>channels allocated).</u>		90 00
2	Onen 4 chemnele	No avecation about he through	ODEN CHANNEL propries
3	Open 4 channels	No exception shall be thrown.	OPEN CHANNEL proactive
	Applet2		command are fetched.
	Annalata hadda a marantina samuand ODEN		Successful TERMINAL
	Applet2 builds a proactive command OPEN_CHANNEL 4 times, calling init() and send()		RESPONSE of OPEN
	methods.	-	CHANNEL are sent to the
			SIM with Channel Id = 01 to
			04
4	Open one more channel	Shall throw a ToolkitException with	<u> </u>
	Applet2	reason	
		COMMAND NOT ALLOWED	
	Applet2 builds a proactive command OPEN	SOMMAN AND INCOMPANIES	
	CHANNEL once again, calling init() and		
_	send() methods.		
<u>5</u>	Good installation of applet3		
	Install for install of applet3 (maximum 7		The CIM enguero to the
	channels allocated).		The SIM answers to the Envelope with status words
			90 00
<u>6</u>	Open 3 channels	No exception shall be thrown.	OPEN CHANNEL proactive
	Applet3		command is fetched.
	Applet3 builds a proactive command OPEN_CHANNEL 3 times, calling init() and send()		Successful TERMINAL
	methods.	-	RESPONSE of OPEN
	<u></u>		CHANNEL are sent to the
			SIM with Channel Id from 05
7	Onen ene ware all arreal		ODEN CHANNEL propries
7	Open one more channel	No expention shall be through	OPEN CHANNEL proactive
	Applet3	No exception shall be thrown.	command is fetched.
	handata builda a musartina samusud OPTN		Unsuccessful Terminal Response is sent to the SIM
	Applet3 builds a proactive command OPEN CHANNEL once again, calling init() and		Response is sent to the SIM with 'No Channel Available'
	send() methods.		as Additional Information on
			Result.
			Result.

6.3.8.1.4 Test Coverage

CRR number	Test case number
<u>N1</u>	<u>2,3</u>
<u>C1</u>	<u>1, 7</u>
C2	4,5,6

6.3.8.8 Minimum Security Level

Test Area Reference: FWK TIN MSL

6.3.8.8.1 Conformance Requirements

6.3.8.8.1.1 Normal execution

<u>CRRN1: The Receiving Entity shall check the Minimum Security Level during processing the security of the Command Packet.</u>

CRRN2: The Receiving Entity shall reject the message if the MSL check fails.

<u>CRRN3</u>: If the MSL check fails, a Response Packet with the 'Insufficient Security Level' Response Status Code shall be sent if required.

CRRN4: If the length of the Minimum Security Level field is greater than zero, the Minimum Security Level is used to specify the minimum level of security to be applied to Secured Packets. The first byte shall be the MSL Parameter, other bytes shall be the MSL Data.

<u>CRRN5</u>: If the length of the Minimum Security Level field is zero, no minimum security level check shall be performed by the receiving entity.

<u>CRRN6</u>: If no Minimum Security Level field is present (no MSL length, no MSL parameter and no MSL data), no minimum security level check shall be performed by the receiving entity.

<u>CRRN7</u>: If the Maximum number of channels field is included in the command data then the Length of Minimum Security Level field shall also be included.

CRRN8: If an optional parameter is included, then all the previous parameters shall be included also

6.3.8.8.2 Test suite files

Test Script: FWK_TIN_MSL_1.scr

Test Applet: FWK_TIN_MSL_1.java

Load Script: FWK_TIN_MSL_1.ldr

Cleanup Script: FWK_TIN_MSL_1.clr

Parameter File: FWK_TIN_MSL_1.par

6.3.8.8.3 Test Procedure

ld	<u>Description</u>	API/Framework Expectation	APDU Expectation
1	Installation with MSL length of 0		
	1- Install (install) applet with a MSL length = 0 2- Send formatted SMS PP env with no RC/CC/DS, no Ciphering and counter mode 0 (not checked) 3- Send a formatted SMS PP env with CC, ciphering and counter mode 1 (counter available and no checking) 4- Delete the applet instance	2- Applet is triggered 3- Applet is triggered	1- 9000
2	Installation without MSL field		
	1- Install (install) applet without MSL field (no MSL length, no MSL parameter and no data) 2- Send formatted SMS PP env with no RC/CC/DS, no Ciphering and counter mode 0 (not checked) 3- Send a formatted SMS PP env with CC, ciphering and counter mode 1 counter available and no checking) 4- Delete the applet instance	2- Applet is triggered 3- Applet is triggered	1- 9000

6.3.8.9.4 Test Coverage

CRR number	Test case number	
CRRN1	Not applicable	
CRRN2	Not applicable	
CRRN3	Not applicable	
CRRN4	Not applicable	
CRRN5	<u>1</u>	
CRRN6	<u>2</u>	
CRRN7	Not testable	
CRRN8	Not testable	

6.3.11 Concatenated SMS

6.3.11.1 Concatenation processing

6.3.11.1 Conformance Requirements:

6.3.11.1.1 Normal execution

- <u>CRRN1: The SIM Toolkit Framework shall link single Short Messages together to re-assemble the original message</u> before any further processing.
- <u>CRRN2</u>: The concatenation control headers used to re-assemble the short messages in the correct order shall not be present in the SMS TPDU.
- <u>CRRN3: The TP-elements of the SMS TPDU and the Address (TS-Service-Centre-Address) shall correspond to the ones in the last received Short Message (independently of the Sequence number of Information-Element-Data).</u>
- <u>CRRN4: The original Short Message shall be placed in one SMS TPDU TLV (with TP-UDL field coded on one octet) included in the EnvelopeHandler.</u>
- <u>CRRN5</u>: The SIM Toolkit Framework shall be able to process messages with the following properties:

 The Information Element Identifier is equal to the 8-bit reference number

 It contains uncompressed 8 bit data or uncompressed UCS2 data.

6.3.11.2 Test Suite Files

Test Script: FWK_CSM_PROC_1.scr

Test Applet: FWK_CSM_PROC_1.java

Load Script: FWK_CSM_PROC_1.ldr

Cleanup Script: FWK_CSM_PROC_1.clr

Parameter File: FWK_CSM_PROC_1.par

6.3.11.3 Test Procedure

<u>ld</u>	<u>Description</u>	API/Framework Expectation	APDU Expectation
	Applet registration to EVENT FORMATTED SMS PP ENV and triggering		
	Applet is registered to EVENT_FORMATTED_SMS_PP_ENV and EVENT_UNFORMATTED_SMS_PP_ENV		
	A concatenated formatted SMS_PP short message is sent to the SIM (composed of two segments).		
<u>1</u>	The second segment of a concatenated short message is sent to the SIM.	Applet is not triggered.	
2	The first segment of the concatenated short message is sent to the SIM	Applet is triggered.	
<u>3</u>	Call the EnvelopeHanlder.getTheHandler()	No exception is thrown.	
4	Call the EnvelopeHandler.findTLV()to select the Dev Id, the adress and the TPDU TLV and the EnvelopeHandler.compareValue() to check each content.	Check that the message has been re-assembled in the correct order. Check that TP-UDL field is coded one octet. Check that the concatenation control header is not present in the message. Check the integrity of the message.	

5	A new concatenated formatted short message is	Applet is triggered.	
	sent to the SIM composed of two segments. The		
	Address field of the first segment is different		
	from the address field in the second segment.		
<u>6</u>	Call the EnvelopeHandler.getTheHandler()	No exception is thrown.	
<u>7</u>	Call the EnvelopeHandler.findTLV()to select the	Check that the address field of	
	the address TLV and the	the message is equal to the	
	EnvelopeHandler.compareValue() to check its	address field of the second	
	content.	segment.	
8	A new concatenated formatted short message is	Applet is triggered.	
_	sent to the SIM composed of two segments. Some		
	TP_elements of the TP_DU of the first segment are		
	different from the TP elements in the second		
	segment.		
<u>9</u>	Call the EnvelopeHandler.getTheHandler()	No exception is thrown.	
<u>10</u>	Call the EnvelopeHandler.findTLV()to select the	Check that the TP elements of	
	the TP DU TLV and the	the message are equal to the	
	EnvelopeHandler.compareValue() to check its TP	ones of the second segment.	
44	elements.	A - t - t - t	
<u>11</u>	Send a concatenated formatted short message (composed of 2 segment) with uncompressed 8 bits	Applet is triggered.	
	data.		
	Applet registration to		
	EVENT UNFORMATTED SMS PP ENV and triggering		
	EVERT ON ORMATTED ONO IT ERV and diggering		
	Same test as 1 but with an unformatted SMS_PP		
	envelope.		
	A concatenated unformatted SMS_PP short message		
	is sent to the SIM (composed of two segments).		
<u>12</u>	The second segment of a concatenated short	Applet is not triggered.	
	message is sent to the SIM.		
13	The first segment of the concatenated short	Applet is triggered.	
13	message is sent to the SIM	Applet is triggered.	
4.4	Call the EnvelopeHanlder.getTheHandler()	No secondina in the second	
<u>14</u>	Call the Envelopenantuel.getimenanuler()	No exception is thrown.	
4.5	Call the EnvelopeHandler.findTLV()to select the	Object the state of the supplier of the state of the stat	
<u>15</u>	Dev Id, the adress and the TPDU TLV and the	Check that the message has	
	EnvelopeHandler.compareValue() to check each	been re-assembled in the	
	content.	correct order. Check that TP-	
	<u> </u>	UDL field is coded one octet.	
		Check that the concatenation	
		control header is not present in	
		the message. Check the	
		integrity of the message.	
<u>16</u>	A new concatenated formatted short message is	Applet is triggered.	
	sent to the SIM composed of two segments. The		
	Address field of the first segment is different		
4=	from the address field in the second segment.	No supporting the disc	
<u>17</u>	Call the EnvelopeHandler.getTheHandler()	No exception is thrown.	
4.5	Gold the problem of the Committee of the		
<u>18</u>	Call the EnvelopeHandler.findTLV()to select the	Check that the address field of	
	the address TLV and the EnvelopeHandler.compareValue() to check its	the message is equal to the	
	EnvelopeHandler.compareValue() to check its content.	address field of the second	
		segment.	
<u>19</u>	A new concatenated unformatted short message is	Applet is triggered.	
	sent to the SIM composed of two segments. Some		
	TP_elements of the TP_DU of the first segment are different from the TP_elements in the second		
	segment.		
20	Call the EnvelopeHandler.getTheHandler()	No exception is thrown.	
20		140 ONOOPHOIT IS HITOWIT.	
21	Call the EnvelopeHandler.findTLV()to select the	Check that the TP elements of	
<u>21</u>	the TP DU TLV and the		
	<u> </u>	the message are equal to the	
	EnvelopeHandler.compareValue() to check its TP	and of the second seament	
	EnvelopeHandler.compareValue() to check its TP elements.	ones of the second segment.	
22			
22	elements.	ones of the second segment. Applet is triggered.	

6.3.11.4 Test Coverage

CRR number	Test case number
<u>N1</u>	1,2, 3, 5, 6, 8, 9, 12, 13,
	<u>14, 16, 17, 19, 20</u>
<u>N2</u>	<u>4,15</u>
<u>N3</u>	<u>7,10, 18, 21</u>
<u>N4</u>	<u>4,15</u>
<u>N5</u>	11,22

Annex A (normative): Class and Methods AID numbering and acronyms

A.2.4 EnvelopeHandler methods

Method Name	Acronyms	Numbering on 6 bits
<pre>byte_Byte_getEnvelopeTag()</pre>	GENT	000001
<pre>byte_Byte getItemIdentifier()</pre>	GIID	000010
<pre>short_Short_getSecuredDataLength()</pre>	GSDL	000011
<pre>short getSecuredDataOffset()</pre>	GSDO	000100
EnvelopeHandler getTheHandler()	GTHD	000101
<pre>short getTPUDLOffset()</pre>	GTPO	000110
Short getCapacity()	GCAP	010010
Short getUserDataLength()	GUDL	010011
Byte getChannelIdentifier()	GCID	010100
Inhorited Method News Wisudles die		
Inherited Method Name: ViewHandler	ODDVO DOO	000444
Byte compareValue(short valueOffset,byte[] compareBuffer, short compareOffset, short compareLength)	CPRVS_BSS	000111
Short copy(byte[] dstBuffer,short dstOffset,short dstLengt h)	COPY_BSS	001000
Short copyValue(short valueOffset, byte[] dstBuffer,short dstOffset,short dstLength)	CPYVS_BSS	001001
<pre>-Byte findAndCompareValue(byte tag,byte[] compareBuffer,sh ort compareOffset)</pre>	FACRB_BS	001010
-Byte findAndCompareValue(byte tag,byte occurrence,-short valueOffset,byte[] compareBuffer,short compareOffset,short compareLength)	FACRBBS_BSS	001011
-Short FindAndCopyValue(byte tag,byte occurrence,short valu eOffset, byte[] dstBuffer, short dstOffset, short dstLength)	FACYBBS_BSS	001100
-Short findAndCopyValue(byte tag,byte[] dstBuffer,short dst Offset)	FACYB_BS	001101
Byte FindTLV(byte tag,byte occurrence)	FINDBB	001110
Short GetLength()	GLEN	001111
-Byte GetValueByte(short valueOffset)	GVBYS	010000
-Short GetValueLength()	GVLE	010001

A.2.5 EnvelopeResponseHandler methods

Method Name	Acronym	Numbering on 6 bits
EnvelopeResponseHandler getTheHandler()	GTHD	000001
Void post(byte statusType)	POSTB	000010
Void postAsBERTLV(byte statusType, byte tag)	POSTBB	000011
Short getCapacity()	<u>GCAP</u>	<u>010101</u>
Inherited Method Name: EditHandler	1004 000	200400
Void appendArray(byte[] buffer, short offset, short length)	APDA_BSS	000100
Void appendTLV(byte tag, byte value)	APTLBB	000101
Void appendTLV(byte tag, byte[] value, short	APTLB_BSS	000110
valueOffset, short valueLength)		
Void appendTLV(byte tag, byte value1, byte value2)	APTLBBB	000111
Void appendTLV(byte tag, byte value1, byte[] value2,	APTLBB_BSS	001000
<pre>short value2Offset, short value2Length) Void clear()</pre>	CLER	001001
Void Clear()	CLER	001001
Inherited Method Name: ViewHandler		
Byte	CPRVS_BSS	001010
compareValue(short valueOffset,byte[] compareBuffer,	5 <u>_</u>	
short compareOffset, short compareLength)		
Short Copy(byte[] dstBuffer,short dstOffset,short dstLengt	COPY_BSS	001011
h)		
Short	CPYVS BSS	001100
CopyValue(short valueOffset,		
<pre>byte[] dstBuffer,short dstOffset,short dstLength)</pre>		
- Byte	FACRB BS	001101
FindAndCompareValue(byte tag,byte[] compareBuffer,sh	THORD_BO	001101
ort compareOffset)		
D. L.	EAODD DO	004404
<pre>Byte findAndCompareValue(byte tag,byte[] compareBuffer,sh</pre>	FACRB_BS	<u>001101</u>
ort compareOffset)		
-Byte findAndCompareValue(byte tag,byte occurence, -	FACRBBS_BSS	001110
<pre>short valueOffset,byte[] compareBuffer,short compare Offset,short compareLength)</pre>		
-Short-	FACYBBS BSS	001111
FindAndCopyValue(byte tag,byte occurence,short value		
Offset, byte[] dstBuffer, short dstOffset,		
short dstLength) Short	FACYBBS BSS	001111
findAndCopyValue(byte tag,byte occurence,short value	FACTBBS_BSS	001111
Offset, byte[] dstBuffer, short dstOffset,		
short dstLength)	510V5 50	
-Short findAndCopyValue(byte tag,byte[] dstBuffer,short dst	FACYB_BS	010000
Offset)		
Byte	FINDBB	010001
FindTLV(byte tag, byte occurrence)		
Byte findTLV(byte tag,byte occurrence)	<u>FINDBB</u>	<u>010001</u>
Short	GLEN	010010
GetLength()		
-Byte	GVBYS	010011
GetValueByte(short valueOffset)	CV/DV/C	040044
Byte getValueByte(short valueOffset)	<u>GVBYS</u>	<u>010011</u>
-Short	GVLE	010100
GetValueLength()		
Short	<u>GVLE</u>	<u>010100</u>
getValueLength()		<u> </u>

A.2.7 ProactiveHandler methods

Method Name	Acronyms	Numbering on 6 bits
GetTheHandler()	GTHD	000001
ProactiveHandler getTheHandler()	GTHD	000001
<pre>Init(byte type, byte qualifier, byte dstDevice)</pre>	INITBBB	000010
Void init(byte type, byte qualifier, byte dstDevice)	INITBBB	000010
<pre>InitDisplayText(byte qualifier, byte des, byte[] buffer, short offset, short length)</pre>	INDTBB_BSS	000011
Void initDisplayText(byte qualifier, byte dcs, byte[] buffer, short offset, short length)	INDTBB BSS	000011
<pre>InitGetInkey(byte qualifier, byte des, byte[] buffer, short offset, short length)</pre>	INGKBB_BSS	000100
Void initGetInkey(byte qualifier, byte dcs, byte[] buffer, short offset, short length)	INGKBB_BSS	000100
<pre>InitGetInput(byte qualifier, byte dcs, byte[] buffer, short offset, short length, short minRespLength, short maxRespLength)</pre>	INGPBB_BSSSS	000101
Void initGetInput(byte qualifier, byte dcs, byte[] buffer, short offset, short length, short minRespLength, short maxRespLength)	INGPBB BSSSS	<u>000101</u>
Byte send()	SEND	000110
Short getCapacity()	GCAP	011000
Void initCloseChannel(byte bChannelIdentifier)	<u>ICCHB</u>	<u>011001</u>
Inherited Method Name: EditHandler		
<pre>Void appendArray(byte[] buffer, short offset, short length)</pre>	APDA_BSS	000111
Void appendTLV(byte tag, byte value)	APTLBB	001000
Void appendTLV(byte tag, byte[] value, short valueOffset, short valueLength)	APTLB_BSS	001001
Void appendTLV(byte tag, byte value1, byte value2)	APTLBBB	001010
<pre>Void appendTLV(byte tag, byte value1, byte[] value2, short value2Offset, short value2Length)</pre>	APTLBB_BSS	001011
Void clear()	CLER	001100

Inherited Method Name: ViewHandler		
Byte	CPRVS_BSS	001101
CompareValue(short valueOffset,byte[] compareBuffer,s		
hort compareOffset, short compareLength) Byte	CPRVS_BSS	001101
compareValue(short valueOffset,byte[] compareBuffer,	CPRVS_BSS	<u>001101</u>
short compareOffset, short compareLength)		
Short	COPY_BSS	001110
Copy(byte[] dstBuffer,short dstOffset,short dstLength		
Short	COPY_BSS	001110
<pre>copy(byte[] dstBuffer,short dstOffset,short dstLengt</pre>	COPY BSS	001110
h)		
Short	CPYVS_BSS	001111
CopyValue(short valueOffset,		
<pre>byte[] dstBuffer,short dstOffset,short dstLength)</pre>		
Short	CPYVS_BSS	001111
copyValue(short valueOffset,	<u> </u>	<u>001111</u>
<pre>byte[] dstBuffer,short dstOffset,short dstLength)</pre>		
	EAODD DO	040000
-Byte FindAndCompareValue(byte tag,byte[] compareBuffer,sho	FACRB_BS	010000
rt compareOffset)		
Byte	FACRB_BS	010000
<pre>findAndCompareValue(byte tag,byte[] compareBuffer,sh</pre>		
ort compareOffset)		
-Byte findAndCompareValue(byte tag,byte occurence, -	FACRBBS BSS	010001
short valueOffset,byte[] compareBuffer,short compareO	1 AONDDO_DOO	010001
ffset, short compareLength)		
-Short-	FACYBBS_BSS	010010
FindAndCopyValue(byte tag,byte occurence,short value0 ffset, byte[] dstBuffer, short dstOffset,		
short dstLength)		
Short	FACYBBS BSS	010010
findAndCopyValue(byte tag,byte occurence,short value	 	<u> </u>
Offset, byte[] dstBuffer, short dstOffset,		
short dstLength)	FACYD DO	040044
-Short findAndCopyValue(byte tag,byte[] dstBuffer,short dstO	FACYB_BS	010011
ffset)		
Byte	FINDBB	010100
FindTLV(byte tag, byte occurrence)		
Byte find II / byte tog byte oggymnenge)	<u>FINDBB</u>	<u>010100</u>
findTLV(byte tag,byte occurrence)	GLEN	010101
GetLength()	OLLIV	010101
Short	GLEN	010101
getLength()		<u></u>
Byte	GVBYS	010110
CetValueByte(short_valueOffset)	CVPVC	010110
<pre>Byte getValueByte(short valueOffset)</pre>	<u>GVBYS</u>	<u>010110</u>
-Short	GVLE	010111
CetValueLength()	3.22	
		040444
Short getValueLength()	<u>GVLE</u>	<u>010111</u>

A.2.8 ProactiveResponseHandler methods

Method Name	Acronyms	Numbering on 6 bits
<pre>Short CopyAdditionalInformation(byte[] dstBuffer,</pre>	CPAI_BSS	000001
<pre>short dstOffset, short dstLength)</pre>	_	
<pre>Short copyAdditionalInformation(byte[] dstBuffer,</pre>	CPAL BSS	<u>000001</u>
<pre>short dstOffset, short dstLength)</pre>		
Short copyTextString(byte[] dstBuffer, short	CPTS_BS	000010
dstOffset)		
Short getAdditionalInformationLength()	GTIL	000011
Byte getGeneralResult()	GTGR	000100
Byte getItemIdentifier()	GTII	000101
Byte getTextStringCodingScheme()	GTCS	000110
Short getTextStringLength()	GTTL	000111
GetTheHandler()	GTHD	001000
ProactiveResponseHandler getTheHandler()	<u>GTHD</u>	<u>001000</u>
Short getCapacity()	GCAP	<u>010100</u>
Byte getChannelIdentifier()	GCID	<u>010101</u>
Short copyChannelData(byte[] dstBuffer,	CCHD BSS	010110
short dstOffset, short dstLength)		

Inherited Method Name: ViewHandler		
Byte CompareValue(short valueOffset,byte[] compareBuffer,s hort compareOffset, short compareLength)	CPRVS_BSS	001001
Short Copy(byte[] dstBuffer,short dstOffset,short dstLength)	COPY_BSS	001010
Short CopyValue(short valueOffset, byte[] dstBuffer,short dstOffset,short dstLength)	CPYVS_BSS	001011
Byte FindAndCompareValue(byte tag,byte[] compareBuffer,sho rt compareOffset)	FACRB_BS	001100
Byte findAndCompareValue(byte tag,byte occurence, short valueOffset,byte[] compareBuffer,short compareOffset,short compareLength)	FACRBBS_BSS	001101
Short FindAndCopyValue(byte tag,byte occurence,short valueO ffset, byte[] dstBuffer, short dstOffset, short dstLength)	FACYBBS_BSS	001110
Short findAndCopyValue(byte tag,byte[] dstBuffer,short dstO ffset)	FACYB_BS	001111
Byte FindTLV(byte tag,byte occurrence)	FINDBB	010000
Short GetLength()	GLEN	010001
Byte GetValueByte(short valueOffset)	GVBYS	010010
Short GetValueLength()	GVLE	010011

B.4 Style and formatting

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

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

CMD COMMAND [EXPECTED DATA] (EXPECTED STATUS)

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

F.1 Toolkit Installation Parameters (TIN)

Test Area within the chapter	Acronyms	Numbering on 6 bits
Timer allocation	TMAL	000001
Item identifier	ITID	000010
Item position	ITPO	000011
Access conditions	ACCO	000100
Priority level	PRLV	000101
Maximum length for each menu entry	MLME	000110
Number of menu entries	NBME	000111
Memory space	MESP	001000
<u>Channel Allocation</u>	<u>CHAL</u>	001001
Minimum Security Level	MSL	001010

F.3 Handler Integrity (HIN)

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

F.4 Applet Triggering (APT)

Test Area within the chapter	Acronyms	Numbering on 6 bits
EVENT_PROFILE_DOWNLOAD	EPDW	000001
EVENT_MENU_SELECTION	EMSE	000010
EVENT_MENU_SELECTION_HELP_REQUEST	EMSH	000011
EVENT_FORMATTED_SMS_PP_ENV	EFSE	000100
EVENT_UNFORMATTED_SMS_PP_ENV	EUSE	000101
EVENT_CALL_CONTROL_BY_SIM	ECCN	000110
EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM	EMCN	000111
EVENT_TIMER_EXPIRATION	ETEX	001000
EVENT_UNFORMATTED_SMS_CB	EUCB	001001
EVENT_EVENT_DOWNLOAD_MT_CALL	EDMC	001010
EVENT_EVENT_DOWNLOAD_CALL_CONNECTED	EDCC	001011
EVENT_EVENT_DOWNLOAD_CALL_DISCONNECTED	EDCD	001100
EVENT_EVENT_DOWNLOAD_LOCATION_STATUS	EDLS	001101
EVENT_EVENT_DOWNLOAD_USER_ACTIVITY	EDUA	001110
EVENT_EVENT_DOWNLOAD_IDLE_SCREEN_AVAILABLE	EDIS	001111
EVENT_EVENT_DOWNLOAD_CARD_READER_STATUS	EDCR	010000
RFU (EVENT_UNRECOGNIZED_ENVELOPE)	(EUEN)	010001
EVENT_UNRECOGNIZED_ENVELOPE	EUEV	010001
EVENT_STATUS_COMMAND	ESTC	010010
EVENT_EVENT_DOWNLOAD_LANGUAGE_SELECTION	EDLG	010011
EVENT_EVENT_DOWNLOAD_BROWSER_TERMINATION—	EDBT	010100
EVENT_FORMATTED_SMS_CB	EFCB	010101
EVENT FIRST COMMAND AFTER SELECT	<u>EFCA</u>	010110
EVENT_EVENT_DOWNLOAD_DATA_AVAILABLE	EDDA	010111
EVENT EVENT DOWNLOAD CHANNEL STATUS	<u>EDCS</u>	011000
EVENT FORMATTED SMS PP UPD	<u>EFSU</u>	011001
EVENT UNFORMATTED SMS PP UPD	<u>EUSU</u>	011010

F.5 Proactive Command Sending (PCS)

Test Area within the chapter	Acronyms	Numbering on 6 bits
System Proactive commands	SPCO	000001
Interaction with GSM commands	IGCO	000010
Errors during proactive command sending	EPCS	000011
Proactive Command Control	PCCO	000100

F.6 Envelope Response Posting (ERP)

Test Area within the chapter	Acronyms	Numbering on 6 bits
EVENT_CALL_CONTROL_BY_SIM	ECCN	000001
EVENT_MO_SHORT_MESSAGE_CONTROL_BY_SIM	EMCN	000010
EVENT_UNRECOGNIZED_ENVELOPE	EUEN	000011
EVENT_FORMATTED_SMS_PP_ENV	<u>EFSE</u>	000010

F.11 Concatenation processing (PROC)

Test Area within the chapter	<u>Acronyms</u>	Numbering on 6 bits
Concatenation processing	PROC	000001

G.2.5 INSTALL(install) Section

Here are the parameters to be included in the Install(Install) command (as specified in GSM 03.48 TS 23.048 [8])

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

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

G.3 Full example

```
[CONVERT]

PackageAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 00

PackageName = sim.test.access.api_1_svw_updrbs

PackageVersion = 1.0
```

```
AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 01
AppletClassName = API_1_SVW_UPDRBS_1
AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 02
AppletClassName = API_1_SVW_UPDRBS_2
[INSTALL(load)]
PackageNonVolatileMemSize = 0D27
;InstallationNonVolatileMemSize = 0400
;InstallationVolatileMemSize = 0000
[LOAD]
MaxLoadCommandDataLength = 6C ; max value
[INSTALL(install)]
AppletClassAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 01
InstanceAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 01
InstallationNonVolatileMemSize = 0400
InstallationVolatileMemSize = 0000
AccessDomain = 00
PriorityLevel = FF
MaxNumberOfTimers = 00
MaxMenuEntryTextLength = 10
MaxNumberOfMenuEntries = 01
MenuEntriesPositionIdentifier = 0001
AppletSpecificParameters =
[INSTALL(install)]
AppletClassAID = A0 00 00 00 30 00 02 FF FF FF 89 00 00 01 02
InstanceAID = A0 00 00 00 30 00 02 FF FF FF FF 89 00 00 01 02
InstallationNonVolatileMemSize = 0200
InstallationVolatileMemSize = 0000
MenuEntriesPositionIdentifier = 0002
MaxNumberOfChannels = 05
MSLFieldLength = 00
MSLParameter =
```

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