ETSI TS 103 115 V9.05.0 (2017-09)

Smart Cards;

Test specification for UICC Application Programming

Interface for Java Card™ for Contactless Applications;

Test Environment and Annexes

(Release 9)

**Technical Specification**

Reference

RTS/SCP-00HCI\_API\_TESTv940

Keywords

API, NFC, Smart Card, testing

***ETSI***

650 Route des Lucioles

F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C

Association à but non lucratif enregistrée à la

Sous-Préfecture de Grasse (06) N° 7803/88

***Important notice***

The present document can be downloaded from:  
<http://www.etsi.org/standards-search>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at <http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:  
<https://portal.etsi.org/People/CommiteeSupportStaff.aspx>

***Copyright Notification***

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.  
The content of the PDF version shall not be modified without the written authorization of ETSI.  
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2017.

All rights reserved.

**DECT**TM, **PLUGTESTS**TM, **UMTS**TM and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.  
**3GPP**TM and **LTE**™ are Trade Marks of ETSI registered for the benefit of its Members and  
of the 3GPP Organizational Partners.  
**GSM**® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Contents

Intellectual Property Rights 7

Foreword 7

Modal verbs terminology 7

1 Scope 8

2 References 8

2.1 Normative references 8

2.2 Informative references 9

3 Definitions, symbols and abbreviations 9

3.1 Definitions 9

3.2 Symbols 9

3.3 Abbreviations 10

3.4 Formats 10

3.4.1 Format of the table of optional features 10

3.4.2 Format of the applicability table 11

3.4.3 Status and Notations 11

4 Applicability 12

4.1 Table of optional features 12

4.2 Applicability table 12

4.3 Information provided by the device supplier 15

4.4 Execution requirements 15

5 Test environment 16

5.1 Test environment description 16

5.2 Tests format 16

5.2.1 Test area reference 16

5.2.1.1 Conformance requirements 17

5.2.1.2 Test suite files 17

5.2.1.3 Initial conditions 17

5.2.1.4 Test procedure 17

5.3 Initial conditions 18

5.4 Package name 18

5.5 AID coding 18

5.6 Test equipment 19

5.6.1 Test tool 19

5.6.2 Java Software Development Kit 20

6 Test cases 20

6.1 Package uicc.hci.framework 20

6.1.1 Class HCIDevice 20

6.1.1.1 Method getHCIService 20

6.1.1.1.1 Conformance requirements 20

6.1.1.1.2 Test suite files 20

6.1.1.1.3 Initial conditions 20

6.1.1.1.4 Test procedure 21

6.1.1.2 Method getPowerMode 22

6.1.1.2.1 Conformance requirements 22

6.1.1.2.2 Test suite files 22

6.1.1.2.3 Initial conditions 22

6.1.1.2.4 Test procedure 23

6.1.1.3 Method isHCIServiceAvailable 23

6.1.1.3.1 Conformance requirements 23

6.1.1.3.2 Test suite files 24

6.1.1.3.3 Initial conditions 24

6.1.1.3.4 Test procedure 24

6.1.2 Interface HCIService 26

6.1.2.1 Method register 26

6.1.2.1.1 Conformance requirements 26

6.1.2.1.2 Test suite files 26

6.1.2.1.3 Initial conditions 26

6.1.2.1.4 Test procedure 27

6.1.2.2 Method deregister 28

6.1.2.2.1 Conformance requirements 28

6.1.2.2.2 Test Suite Files 28

6.1.2.2.3 Initial conditions 28

6.1.2.2.4 Test procedure 28

6.1.2.3 Method activateEvent 29

6.1.2.3.1 Conformance requirements 29

6.1.2.3.2 Test Suite Files 30

6.1.2.3.3 Initial conditions 30

6.1.2.3.4 Test procedure 31

6.1.2.4 Method deactivateEvent 33

6.1.2.4.1 Conformance requirements 33

6.1.2.4.2 Test suite files 34

6.1.2.4.3 Initial conditions 34

6.1.2.4.4 Test procedure 35

6.1.2.5 Method requestCallbackNotification 38

6.1.2.5.1 Conformance requirements 38

6.1.2.5.2 Test Suite Files 38

6.1.2.5.3 Initial conditions 38

6.1.2.5.4 Test procedure 38

6.1.2.6 Method getEventNotificationStatus 39

6.1.2.6.1 Conformance requirements 39

6.1.2.6.2 Test Suite Files 39

6.1.2.6.3 Initial conditions 39

6.1.2.6.4 Test procedure 40

6.1.3 Interface HCIMessage 43

6.1.3.1 Method isHeading 43

6.1.3.1.1 Conformance requirements 43

6.1.3.1.2 Test Suite Files 43

6.1.3.1.3 Initial conditions 43

6.1.3.1.4 Test procedure 44

6.1.3.2 Method isComplete 44

6.1.3.2.1 Conformance requirements 44

6.1.3.2.2 Test suite files 44

6.1.3.2.3 Initial condition 44

6.1.3.2.4 Test procedure 44

6.1.3.3 Method getType 45

6.1.3.3.1 Conformance requirements 45

6.1.3.3.2 Test Suite Files 45

6.1.3.3.3 Initial condition 45

6.1.3.3.4 Test procedure 46

6.1.3.4 Method getInstruction 46

6.1.3.4.1 Conformance requirements 46

6.1.3.4.2 Test Suite Files 46

6.1.3.4.3 initial condition 46

6.1.3.4.4 Test procedure 47

6.1.3.5 Method getReceiveOffset 47

6.1.3.5.1 Conformance requirements 47

6.1.3.6 Method getReceiveLength 47

6.1.3.6.1 Conformance requirements 47

6.1.3.6.2 Test Suite Files 48

6.1.3.6.3 Initial condition 48

6.1.3.6.4 Test procedure 48

6.1.3.7 Method getReceiveBuffer 48

6.1.3.7.1 Conformance requirements 48

6.1.3.7.2 Test suite files 49

6.1.3.7.3 Initial condition 49

6.1.3.7.4 Test procedure 49

6.1.4 Class HCIException 49

6.1.4.1 Method throwIt 49

6.1.4.1.1 Conformance requirements 49

6.1.4.1.2 Test suite files 49

6.1.4.1.3 Initial conditions 49

6.1.4.1.4 Test procedure 50

6.1.5 Interface HCIListener 50

6.1.5.1 Method onCallback 50

6.1.5.1.1 Conformance requirements 50

6.1.5.1.2 Test Suite Files 51

6.1.5.1.3 Initial conditions 51

6.1.5.1.4 Test procedure 52

6.2 Package uicc.hci.services 53

6.2.1 Package CardEmulation Service 53

6.2.1.1 Interface CardEmulationMessage 53

6.2.1.1.1 Method prepareAndSendGetParameterCommand 53

6.2.1.1.2 Method prepareAndSendSendDataEvent 56

6.2.1.1.2.3 Initial conditions 57

6.2.1.1.3 Method selectingMessage 58

6.2.1.1.3.3 Initial conditions 58

6.2.1.2 Interface CardEmulationService 59

6.2.1.2.1 Method getCardRFType 59

6.2.1.2.2 Test Suite Files 59

6.2.1.2.3 Initial conditions 59

6.2.1.3 Interface CardEmulationListener 60

6.2.1.3.1 Method onCallback 60

6.2.2 Package Connectivity Service 63

6.2.2.1 Interface ConnectivityService 63

6.2.2.1.1 Method prepareAndSendConnectivityEvent 63

6.2.2.1.2 Method prepareAndSendTransactionEvent(byte[] aid, short aidOffset, short aidLen, byte[] parameters, short parametersOffset, short parametersLen) 67

6.2.2.1.3 Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen) 71

6.2.3 Package Reader Service 74

6.2.3.1 Interface ReaderMessage 74

6.2.3.1.1 Method restartReaderModeProcedure 74

6.2.3.1.2 Method prepareAndSendWriteXchgDataCommand 77

6.2.3.1.3 Method prepareAndSendGetParameterCommand 82

6.2.3.2 Interface ReaderListener 86

6.2.3.2.1 Method onCallback 86

Annex A (normative): Class, methods and tests acronyms 89

A.1 HCI framework 89

A.1.1 Class HCIDevice 89

A.1.2 Interface HCIService 89

A.1.3 Interface HCIMessage 89

A.1.4 Interface HCIListener 89

A.1.5 Class HCIException 89

A.2 HCI Services 90

A.2.1 Package cardemulation 90

A.2.1.1 Interface CardEmulationListener 90

A.2.1.2 Interface CardEmulationMessage 90

A.2.1.3 Interface CardEmulationService 90

A.2.2 Package connectivity 90

A.2.2.1 Interface ConnectivityListener 90

A.2.2.2 Interface ConnectivityMessage 90

A.2.2.3 Interface ConnectivityService 90

A.2.3 Readermode 90

A.2.3.1 Interface RaederListener 90

A.2.3.2 Interface ReaderMessage 91

A.2.3.3 Interface ReaderService 91

Annex B (normative): AIDs - to be reserved 92

B.1 Package HCI framework 92

B.1.1 Class HCIDevice 92

B.1.2 Interface HCIService 92

B.1.3 Interface HCIMessage 93

B.1.4 Interface HCIListener 93

B.1.5 Class HCIException 93

B.2 HCI Services 93

B.2.1 Package cardemulation 93

B.2.1.1 Interface CardEmulationListener 93

B.2.1.2 Interface CardEmulationMessage 94

B.2.1.3 Interface CardEmulationService 94

B.2.2 Package connectivity 94

B.2.2.1 Interface ConnectivityListener 94

B.2.2.2 Interface ConnectivityMessage 94

B.2.2.3 Interface ConnectivityService 95

B.2.3 Package readermode 95

B.2.3.1 Interface ReaderMessage 95

B.2.3.2 Interface ReaderListener 95

B.2.3.3 Interface ReaderService 96

Annex C (normative): Requirements 97

C.1 Non-occurrence and out-of-scope requirements 97

C.1.1 Package uicc.hci 97

C.1.2 ETSI TS 102 705 prose part 97

C.2 FFS requirements 97

C.2.1 Package uicc.hci 97

C.2.2 ETSI TS 102 705 prose part 98

Annex D (normative): Test Specification for Java Card™ Platform HCI API for the UICC 99

Annex E (normative): Void 100

Annex F (informative): Void 101

Annex G (informative): Core specification version information 102

Annex H (informative): Change history 103

History 105

# Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server ([http://ipr.etsi.org](http://webapp.etsi.org/IPR/home.asp)).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

# Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Smart Card Platform (SCP).

The contents of the present document are subject to continuing work within TC SCP and may change following formal TC SCP approval. If TC SCP modifies the contents of the present document, it will then be republished by ETSI with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

0 early working draft;

1 presented to TC SCP for information;

2 presented to TC SCP for approval;

3 or greater indicates TC SCP approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

# Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](http://portal.etsi.org/Help/editHelp!/Howtostart/ETSIDraftingRules.aspx) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

# 1 Scope

The present document covers the minimum characteristics considered necessary in order to provide compliance to ETSI TS 102 705 [].

It specifies conformance test cases for the UICC Application Programming Interface for Java Card™ for contactless Applications.

# 2 References

## 2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non‑specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

* In the case of a reference to a TC SCP document, a non specific reference implicitly refers to the latest version of that document in the same Release as the present document.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI TS 102 705: "Smart Cards; UICC Application Programming Interface for Java Card™ for Contactless Applications".

[2] ISO/IEC 7816-3: "Identification cards - Integrated circuit cards - Part 3: Cards with contacts - Electrical interface and transmission protocols".

[3] ETSI TS 102 622: "Smart Cards; UICC - Contactless Front-end (CLF) Interface; Host Controller Interface (HCI)".

[4] ETSI TS 101 220: "Smart Cards; ETSI numbering system for telecommunication application providers".

[5] ETSI TS 102 221: "Smart Cards; UICC-Terminal interface; Physical and logical characteristics".

[6] ETSI TS 102 241: "Smart Cards; UICC Application Programming Interface (UICC API) for Java Card (TM)".

[7] ETSI TS 102 223: "Smart Cards; Card Application Toolkit (CAT)".

[8] ISO/IEC 9646-7: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 7: Implementation Conformance Statements".

[9] ETSI TS 102 226: "Smart Cards; Remote APDU structure for UICC based applications".

[10] GlobalPlatform: "GlobalPlatform Card Specification Version 2.2, Amendment C: Contactless Services" Version 1.0.

NOTE: See <http://www.globalplatform.org/>.

[11] Sun Microsystems "Application Programming Interface, Java Card™ Platform, 3.0.1 Classic Edition".

[12] Sun Microsystems "Runtime Environment Specification, Java Card™ Platform, 3.0.1 Classic Edition".

[13] Sun Microsystems "Virtual Machine Specification Java Card™ Platform, 3.0.1 Classic Edition".

NOTE: SUN Java Card Specifications can be downloaded at <http://www.oracle.com/technetwork/java/javame/javacard/download/overview/index.html>.

[14] ETSI TS 102 613: "Smart Cards; UICC - Contactless Front-end (CLF) Interface; Part 1: Physical and data link layer characteristics".

[15] Java Card API and Export File for Card Specification v2.2.1 (org.globalplatform) v1.5.

[16] Java Card Contactless API and Export File for Card Specification v2.2.1 (org.globalplatform.contactless) v1.1.

## 2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non‑specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

* In the case of a reference to a TC SCP document, a non specific reference implicitly refers to the latest version of that document in the same Release as the present document.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

Not applicable.

# 3 Definitions, symbols and abbreviations

## 3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

**applet installation parameters:** values for applet installation parameters in the Install(Install) command

**Conformance Requirement Reference (CRR):** description of the expected HCI API behaviour according to ETSI TS 102 705 []

**contactless mode:** is used as a generic term for "Card Emulation Mode" and "Reader Mode"

**HCP message:** message as specified in ETSI TS 102 622 []

NOTE: An HCP message can be of type "command", "event" or "response to a command".

**test case:** elementary test that checks for compliance with one or more Conformance Requirement References

**test procedure:** sequence of actions/commands to perform all the test cases defined in a test area

**test source file:** file containing methods that will load and install test applet in the card, execute and verify the test results, and restore the Default Initial Conditions on the UICC (when possible)

**RF Technology:** radio frequency technology supported by the HCI (ETSI TS 102 622 []) protocol specification

## 3.2 Symbols

For the purposes of the present document, the symbols given in ETSI TS 102 705 [] apply.

## 3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AID Application IDentifier

APDU Application Protocol Data Unit

NOTE: According to ISO/IEC 7816-3 [].

API Application Programming Interface

CAT Card Application Toolkit

CB Chaining Bit

CLF Contactless Front-end

NOTE: According to ETSI TS 102 622 [].

CRR Conformance Requirements Reference

CRRC Conformance Requirement Reference Context Error

CRRN Conformance Requirement Reference Normal

CRRP Conformance Requirement Reference Parameter Error

CRS Contactless Registry Services

CTR ConTRol

DUT Device Under Test

EVT EVenT

FFS For Further Study

GND GrouND

HCI Host Controller Interface

NOTE: According to ETSI TS 102 622 [].

HCP Host Controller Protocol

NOTE: According to ETSI TS 102 622 [].

INS INStruction

ISO International Organisation for Standardisation

JCRE Java Card™ Run-time Environment

RF Radio Frequency

SAA Service Availability and Access possibility for the different services

SDK Software Development Kit

SW Status Word

SWP Single Wire Protocol

TAR Toolkit Application Reference

## 3.4 Formats

### 3.4.1 Format of the table of optional features

The columns in table 4.1 have the following meaning.

|  |  |
| --- | --- |
| Column | Meaning |
| Option | The optional feature supported or not by the DUT. |
| Status | See clause 3.4.3. |
| Support | The support columns are to be filled in by the supplier of the implementation. The following common notations, defined in ISO/IEC 9646-7 [], are used for the support column in table 4.1.  Y or y supported by the implementation.  N or n not supported by the implementation.  N/A, n/a or - no answer required (allowed only if the status is N/A, directly or after evaluation of a conditional status). |
| Mnemonic | The mnemonic column contains mnemonic identifiers for each item. |

### 3.4.2 Format of the applicability table

The applicability of every test in table 4.2 is formally expressed by the use of Boolean expression defined in the following clause.

The columns in table 4.2 have the following meaning.

| Column | Meaning |
| --- | --- |
| Clause | The "Clause" column identifies the clause containing the test case referenced in the "Test case number and description" column. |
| Test case number and description | The "Test case number and description" column gives a reference to the test case number (along with the corresponding description) detailed in the present document and required to validate the DUT. |
| Release | The "Release" column gives the Release applicable and onwards, for the corresponding test case. |
| Execution requirements | The usage of the "Execution requirements" column is described in clause 4.4. |
| Rel-x UICC | For a given Release, the corresponding "Rel-x UICC" column lists the tests required for a DUT to be declared compliant to this Release. |
| Support | The "Support" column is blank in the proforma, and is to be completed by the manufacturer in respect of each particular requirement to indicate the choices, which have been made in the implementation. |

### 3.4.3 Status and Notations

The "Rel-x" columns show the status of the entries as follows:

The following notations, defined in ISO/IEC 9646‑7 [], are used for the status column:

M mandatory - the capability is required to be supported.

O optional - the capability may be supported or not.

N/A not applicable - in the given context, it is impossible to use the capability.

X prohibited (excluded) - there is a requirement not to use this capability in the given context.

O.i qualified optional - for mutually exclusive or selectable options from a set. "i" is an integer which identifies an unique group of related optional items and the logic of their selection which is defined immediately following the table.

Ci conditional - the requirement on the capability ("M", "O", "X" or "N/A") depends on the support of other optional or conditional items. "i" is an integer identifying an unique conditional status expression which is defined immediately following the table. For nested conditional expressions, the syntax "IF ... THEN (IF ... THEN ... ELSE...) ELSE ..." is to be used to avoid ambiguities.

References to items

For each possible item answer (answer in the support column) there exists a unique reference, used, for example, in the conditional expressions. It is defined as the table identifier, followed by a solidus character "/", followed by the item number in the table. If there is more than one support column in a table, the columns are to be discriminated by letters (a, b, etc.), respectively.

EXAMPLE: 4.1/4 is the reference to the answer of item 4 in table 4.1.

The ID (identifier) of a test case consists of a main identifier and optionally a sub-identifier; for example, 2-1 and 3. A sub-identifier is used when there are multiple test cases with this same main identifier; otherwise, no sub-identifier is used. Reference to a main identifier when the relevant test cases also have sub-identifier are assumed to reference all of the test cases with that main identifier.

# 4 Applicability

## 4.1 Table of optional features

The device supplier shall state the support of possible options in table 4.1. See clause 3.4 for the format of table 4.1.

Table 4.1: Options

| Item | Option | Status | Support | Mnemonic |
| --- | --- | --- | --- | --- |
| 1 | Card emulation, Type A | O |  | O\_CE\_TYPE\_A |
| 2 | Card emulation, Type B | O |  | O\_CE\_TYPE\_B |
| 3 | Card emulation, Type B' | O |  | O\_CE\_TYPE\_B\_PRIME |
| 4 | Card emulation, Type F | O |  | O\_CE\_TYPE\_F |
| 5 | Reader Mode, Type A | O |  | O\_RM\_TYPE\_A |
| 6 | Reader Mode, Type B | O |  | O\_RM\_TYPE\_B |
| 7 | HCP message size greater than supported buffer size | O |  | O\_MSG\_GT\_BUF |

## 4.2 Applicability table

Table 4.2 specifies the applicability of each test case to the device under test. See clause 3.4 for the format of table 4.2.

Clause 4.4 should be referenced for usage of the execution requirements which are referenced in table 4.2 a) and described in table 4.2 c).

Table 4.2 a): Applicability of tests

| Clause | Test case number and description | Release | Execution requirements | Rel-9  UICC | Support |
| --- | --- | --- | --- | --- | --- |
| 6.1.1.1 | Method getHCIService (ID2, ID4) | Rel-9 | SAA1 | M |  |
| 6.1.1.1 | Method getHCIService (ID1-1) | Rel-9 | SAA1 | M |  |
| 6.1.1.1 | Method getHCIService (ID1-2) | Rel-9 | SAA2,  SAA1 | M |  |
| 6.1.1.1 | Method getHCIService (ID1-3) | Rel-9 | SAA3 | M |  |
| 6.1.1.1 | Method getHCIService (ID3-1) | Rel-9 | SAA6 | M |  |
| 6.1.1.1 | Method getHCIService (ID3-2) | Rel-9 | SAA1,  SAA7 | M |  |
| 6.1.1.1 | Method getHCIService (ID3-3) | Rel-9 | SAA8 | M |  |
| 6.1.1.1 | Method getHCIService (ID5-1) | Rel-9 | SAA4,  SAA1 | M |  |
| 6.1.1.1 | Method getHCIService (ID5-2) | Rel-9 | SAA3,  SAA5 | M |  |
| 6.1.1.2 | Method getPowerMode | Rel-9 | SAA1 | M |  |
| 6.1.1.3 | Method isHCIServiceAvailable (ID1-1, ID3-1) | Rel-9 | SAA1 | M |  |
| 6.1.1.3 | Method isHCIServiceAvailable (ID1-2) | Rel-9 | SAA2,  SAA1 | M |  |
| 6.1.1.3 | Method isHCIServiceAvailable (ID1-3) | Rel-9 | SAA3 | M |  |
| 6.1.1.3 | Method isHCIServiceAvailable (ID2) | Rel-9 |  | M |  |
| 6.1.1.3 | Method isHCIServiceAvailable (ID3-2) | Rel-9 | SAA3 | M |  |
| 6.1.1.3 | Method isHCIServiceAvailable (ID4-1) | Rel-9 | SAA4 | M |  |
| 6.1.1.3 | Method isHCIServiceAvailable (ID4-2) | Rel-9 | SAA5 | M |  |
| 6.1.1.3 | Method isHCIServiceAvailable (ID5-1) | Rel-9 | SAA10 | M |  |
| 6.1.1.3 | Method isHCIServiceAvailable (ID5-2) | Rel-9 | SAA11 | M |  |
| 6.1.2.1 | Method register (ID1-1, ID3-1, ID4) | Rel-9 | SAA1 | M |  |
| 6.1.2.1 | Method register (ID1-2, ID2, ID3-2) | Rel-9 | SAA1,  SAA2 | M |  |
| 6.1.2.1 | Method register (ID1-3, ID3-3) | Rel-9 | SAA3 | M |  |
| 6.1.2.2 | Method deregister (ID1-1, ID1-3) | Rel-9 | SAA1 | M |  |
| 6.1.2.2 | Method deregister (ID1-2, ID2) | Rel-9 | SAA1,  SAA2 | M |  |
| 6.1.2.3 | Method activateEvent (ID1, ID4, ID6, ID7, ID8, ID9, ID10) | Rel-9 | SAA1 | M |  |
| 6.1.2.3 | Method activateEvent (ID2) | Rel-9 | SAA3 | M |  |
| 6.1.2.3 | Method activateEvent (ID3) | Rel-9 | SAA1,  SAA2 | M |  |
| 6.1.2.3 | Method activateEvent (ID5-1) | Rel-9 | SAA10,  SAA3 | M |  |
| 6.1.2.3 | Method activateEvent (ID5-2) | Rel-9 | SAA11,  SAA3 | M |  |
| 6.1.2.4 | Method deactivateEvent (ID1, ID4, ID5) | Rel-9 | SAA1 | M |  |
| 6.1.2.4 | Method deactivateEvent (ID2) | Rel-9 | SAA3 | M |  |
| 6.1.2.4 | Method deactivateEvent (ID7) | Rel-9 | SAA3,  SAA1 | M |  |
| 6.1.2.4 | Method deactivateEvent (ID3, ID6) | Rel-9 | SAA1,  SAA2 | M |  |
| 6.1.2.5 | Method requestCallbackNotification | Rel-9 | SAA1 | M |  |
| 6.1.2.6 | Method getEventNotificationStatus(ID1, ID2, ID5-1) | Rel-9 | SAA1 | M |  |
| 6.1.2.6 | Method getEventNotificationStatus(ID3,ID4, ID5-2) | Rel-9 | SAA3 | M |  |
| 6.1.2.6 | Method getEventNotificationStatus(ID5-3, ID6, ID7) | Rel-9 | SAA2,  SAA1 | M |  |
| 6.1.3.1 | Method isHeading | Rel-9 | SAA1 | M |  |
| 6.1.3.2 | Method isComplete | Rel-9 | SAA1 | M |  |
| 6.1.3.3 | Method getType (ID1) | Rel-9 | SAA1 | M |  |
| 6.1.3.3 | Method getType (ID2) | Rel-9 | SAA2,  SAA1 | M |  |
| 6.1.3.4 | Method getInstruction | Rel-9 | SAA1 | M |  |
| 6.1.3.6 | Method getReceiveLength | Rel-9 | SAA1 | M |  |
| 6.1.3.7 | Method getReceiveBuffer | Rel-9 | SAA1 | M |  |
| 6.1.4.1 | Method throwIt | Rel-9 | SAA1 | M |  |
| 6.1.5.1 | Method onCallback (ID1, ID2) | Rel-9 | SAA1 | M |  |
| 6.1.5.1 | Method onCallback (ID4) | Rel-9 | SAA1 | C002 |  |
| 6.1.5.1 | Method onCallback (ID5) | Rel-9 | SAA1 | C001 |  |
| 6.2.1.1.1 | Method prepareAndSendGetParameterCommand (ID1) | Rel-9 | SAA1 | C001 |  |
| 6.2.1.1.1 | Method prepareAndSendGetParameterCommand (ID2) | Rel-9 | SAA1 | C002 |  |
| 6.2.1.1.1 | Method prepareAndSendGetParameterCommand (ID3-1) | Rel-9 | SAA1 | C001  C007 |  |
| 6.2.1.1.1 | Method prepareAndSendGetParameterCommand (ID3-2) | Rel-9 | SAA1 | C002  C007 |  |
| 6.2.1.1.2 | Method prepareAndSendSendDataEvent (ID1) | Rel-9 | SAA1 | C001 |  |
| 6.2.1.1.2 | Method prepareAndSendSendDataEvent (ID2) | Rel-9 | SAA1 | C002 |  |
| 6.2.1.1.2 | Method prepareAndSendSendDataEvent (ID4, ID5) | Rel-9 | SAA1 | M |  |
| 6.2.1.1.2 | Method prepareAndSendSendDataEvent (ID3) | Rel-9 | SAA1 | C007 |  |
| 6.2.1.1.3 | Method selectingMessage | Rel-9 | SAA1 | M |  |
| 6.2.1.2.1 | Method getCardRFType (ID1) | Rel-9 | SAA1 | C001 |  |
| 6.2.1.2.1 | Method getCardRFType (ID2) | Rel-9 | SAA1 | C002 |  |
| 6.2.1.2.1 | Method getCardRFType (ID4) | Rel-9 | SAA1 | C003 |  |
| 6.2.1.2.1 | Method getCardRFType (ID3) | Rel-9 | SAA1 | C004 |  |
| 6.2.1.3.1 | Method onCallback (ID1,ID2, ID3, ID7-2) | Rel-9 | SAA1 | C001 |  |
| 6.2.1.3.1 | Method onCallback (ID4, ID5, ID6, ID7-3) | Rel-9 | SAA1 | C002 |  |
| 6.2.1.3.1 | Method onCallback (ID7-1, ID7-4) | Rel-9 | SAA1 | M |  |
| 6.2.2.1.1 | Method prepareAndSendConnectivityEvent (ID1, ID3, ID4, ID5) | Rel-9 | SAA2,  SAA1 | M |  |
| 6.2.2.1.1 | Method prepareAndSendConnectivityEvent (ID2-1) | Rel-9 | SAA2,  SAA10 | M |  |
| 6.2.2.1.1 | Method prepareAndSendConnectivityEvent (ID2-2) | Rel-9 | SAA2,  SAA11 | M |  |
| 6.2.2.1.1 | Method prepareAndSendConnectivityEvent (ID6) | Rel-9 | SAA2,  SAA1,  SAA9 | M |  |
| 6.2.2.1.1 | Method prepareAndSendConnectivityEvent (ID7) | Rel-9 | SAA2,  SAA3 | M |  |
| 6.2.2.1.2 | Method prepareAndSendTransactionEvent byte[] aid, short aidOffset, short aidLen, byte[] parameters, short parametersOffset, short parametersLen) (ID1, ID3, ID4) | Rel-9 | SAA1,  SAA2 | M |  |
| 6.2.2.1.2 | Method prepareAndSendTransactionEvent byte[] aid, short aidOffset, short aidLen, byte[] parameters, short parametersOffset, short parametersLen) (ID2-1) | Rel-9 | SAA2,  SAA10 | M |  |
| 6.2.2.1.2 | Method prepareAndSendTransactionEvent byte[] aid, short aidOffset, short aidLen, byte[] parameters, short parametersOffset, short parametersLen) (ID2-2) | Rel-9 | SAA2,  SAA11 | M |  |
| 6.2.2.1.2 | Method prepareAndSendTransactionEvent byte[] aid, short aidOffset, short aidLen, byte[] parameters, short parametersOffset, short parametersLen) (ID6) | Rel-9 | SAA2,  SAA3 | M |  |
| 6.2.2.1.3 | Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen) (ID1, ID4) | Rel-9 | SAA2,  SAA1 | M |  |
| 6.2.2.1.3 | Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen) (ID3) | Rel-9 | SAA2 | M |  |
| 6.2.2.1.3 | Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen) (ID2-1) | Rel-9 | SAA2,  SAA10 | M |  |
| 6.2.2.1.3 | Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen) (ID2-2) | Rel-9 | SAA2,  SAA11 | M |  |
| 6.2.2.1.3 | Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen) (ID5) | Rel-9 | SAA2,  SAA3 | M |  |
| 6.2.3.1.1 | Method restartReaderModeProcedure (ID1) | Rel-9 | SAA3 | C005 |  |
| 6.2.3.1.1 | Method restartReaderModeProcedure (ID2) | Rel-9 | SAA3 | C006 |  |
| 6.2.3.1.1 | Method restartReaderModeProcedure (ID3, ID4) | Rel-9 | SAA3 | M |  |
| 6.2.3.1.1 | Method restartReaderModeProcedure (ID5-2) | Rel-9 | SAA3,  SAA11 | M |  |
| 6.2.3.1.2 | Method prepareAndSendWriteXchgDataCommand (ID1) | Rel-9 | SAA3 | C005 |  |
| 6.2.3.1.2 | Method prepareAndSendWriteXchgDataCommand (ID2) | Rel-9 | SAA3 | C006 |  |
| 6.2.3.1.2 | Method prepareAndSendWriteXchgDataCommand (ID4, ID5, ID6, ID7) | Rel-9 | SAA3 | M |  |
| 6.2.3.1.2 | Method prepareAndSendWriteXchgDataCommand (ID3-2) | Rel-9 | SAA3,  SAA11 | M |  |
| 6.2.3.1.3 | Method prepareAndSendGetParameterCommand (ID1) | Rel-9 | SAA3 | C005 |  |
| 6.2.3.1.3 | Method prepareAndSendGetParameterCommand (ID2) | Rel-9 | SAA3 | C006 |  |
| 6.2.3.1.3 | Method prepareAndSendGetParameterCommand (ID3-2) | Rel-9 | SAA3,  SAA11 | M |  |
| 6.2.3.1.3 | Method prepareAndSendGetParameterCommand (ID5) | Rel-9 | SAA3 | M |  |
| 6.2.3.2.1 | Method onCallback (ID1, ID2, ID4) | Rel-9 | SAA3 | C005 |  |
| 6.2.3.2.1 | Method onCallback (ID5, ID6, ID8) | Rel-9 | SAA3 | C006 |  |

Table 4.2 b): Conditional items referenced by table 4.2 a)

| Conditional item | Description |
| --- | --- |
| C001 | IF O\_CE\_TYPE\_A THEN M ELSE N/A |
| C002 | IF O\_CE\_TYPE\_B THEN M ELSE N/A |
| C003 | IF O\_CE\_TYPE\_B\_PRIME THEN M ELSE N/A |
| C004 | IF O\_CE\_TYPE\_F THEN M ELSE N/A |
| C005 | IF O\_RM\_TYPE\_A THEN M ELSE N/A |
| C006 | IF O\_RM\_TYPE\_B THEN M ELSE N/A |
| C007 | IF O\_MSG\_GT\_BUF THEN M ELSE N/A |

Table 4.2 c): Execution requirements referenced by table 4.2 a)

| Execution requirement | Description |
| --- | --- |
| SAA1 | Card emulation service is available |
| SAA2 | Connectivity service is available |
| SAA3 | Reader Mode service is available |
| SAA4 | Access not allowed for Applet for Card Emulation service |
| SAA5 | Access not allowed for Applet for Reader service |
| SAA6 | Card emulation service is not available |
| SAA7 | Connectivity service is not available |
| SAA8 | Reader Mode service not available |
| SAA9 | Proactive functionality in ETSI TS 102 241 [] is supported |
| SAA10 | Toolkit application is available and registered to the ENVELOPE (EVENT DOWNLOAD - Contactless state request) and allowed to switch on/off HCI interface |
| SAA11 | No CRS on the card and test applet acting like CRS can be loaded |

## 4.3 Information provided by the device supplier

The device supplier shall provide the information indicated in table 4.3.

Table 4.3: Information provided by device suppliers

| Item | Description | Value | Mnemonic |
| --- | --- | --- | --- |
| 1 | Presence of an application associated to an contactless applet that can be launched in the terminal host using EVT\_TRANSACTION |  | AID, Parameters |
| 2 | Supported received message buffer size |  |  |

## 4.4 Execution requirements

Table 4.2, Applicability of tests, specifies execution requirement (SAAn) for several test cases, to define the service availability and access possibility for the different services. For these test cases, the availability of the different services shall be guaranteed in order to execute the corresponding test procedure against the DUT. In case of absence of a particular (SAAn), the corresponding test case should not be carried out.

# 5 Test environment

This clause specifies requirements that shall be met and the testing rules that shall be followed during the test procedure.

## 5.1 Test environment description

The general architecture for the test environment is:



Figure 5.1

## 5.2 Tests format

### 5.2.1 Test area reference

Each test area is referenced as follows:

For HCI Framework and HCI services Testing: 'Api\_[package name]\_[class name]\_[method name]' where

* package name:
* uicc.hci.framework: '1'.
* uicc.hci.services: '2'.
* class name:
* yyy: 3 letters for each class/interface.
* See annex A for full classes/interfaces acronyms list.
* method name:
* zzz[input parameters]:
* See annex A for full methods name acronyms list.

#### 5.2.1.1 Conformance requirements

The conformance requirements are expressed in the following way:

* Method prototype as listed in ETSI TS 102 705 [].
* Normal execution:

- Contains normal execution and correct parameters limit values, each referenced as a Conformance Requirement Reference Normal (CRRN).

* Parameters error:

- Contains parameter errors and incorrect parameter limit values, each referenced as a Conformance Requirement Reference Parameter Error (CRRP).

* Context error:

- Contains errors due to the context the method is used in, each referenced as a Conformance Requirement Reference Context Error (CRRC).

#### 5.2.1.2 Test suite files

Each test suite files contains a table to indicate the correspondence between the applet names and the different test cases.

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| [Test Area Reference]\_[Test applet number].java | Test case |

The test applets and the related Cap Files use the following naming convention:

* Test Applet: [Test Area Reference]\_[Test applet number].java
* Cap File: [Test Area Reference].cap

The applet numbers start from '1'.

The Cap File format is described in Java Card™ Virtual Machine Specification [].

All files from the same test area are located in the same subfolder.

#### 5.2.1.3 Initial conditions

In addition to the general precondtions defined in clause 5.3, this clause defines the initial conditions prior to the execution of each test case; i.e. for each ID.

#### 5.2.1.4 Test procedure

Each test procedure contains a table to indicate the expected responses form the API and APDU as follows.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case | | | | | |
| Id | HCI commands | API Description | API Expectation | HCI Response | CRR |
|  | *Commands sent on HCI interface* | *Test Case detailed description* | *API expected behaviour* | *Expected behaviour at APDU/HCI level* | *Conformance Requirements Reference* |

The ID of a test case consists of the identifier in the ID column and (if present) the sub-identifier in the HCI commands column. Examples of valid IDs are 2-1 and 3.

## 5.3 Initial conditions

Unless otherwise specified, test cases shall be executed in full power mode only.

The Initial Conditions are a set of general prerequisites for the UICC prior to the execution of testing. For each test procedure described in the present document, the following rules apply to the Initial Conditions:

* unless otherwise stated, before installing the applet(s) relevant to the current test procedure, all packages specific to other test procedures shall not be present.

Prior to any test:

* For test cases run in full power mode, the test environment shall have the UICC powered on and performed the session initialization and RF registries updates as described in ETSI TS 102 622 [].
* For test cases run in low power mode, the test environment shall power on the UICC, activate the SWP interface, perform the session initialization and RF registries updates as described in ETSI TS 102 622 [], power down the UICC and power it up again in low power mode.
* The test can be executed once the HCI interface is idle; i.e. no further communication is expected.

## 5.4 Package name

Java packages integrating this Test Suite shall follow this naming convention:

**uicc.hci.test.framework.[Test Area Reference]:** Java Card packages containing Test Area References for the ETSI TS 102 705 [] uicc.hci.framework package.

**uicc.hci.test.services.cardemulation.[Test Area Reference]:** Java Card packages containing Test Area References for the ETSI TS 102 705 [] uicc.hci.services.cardemulation package.

**uicc.hci.test.services.connectivity.[Test Area Reference]:** Java Card packages containing Test Area References for the ETSI TS 102 705 [] uicc.hci.services.connectivity package.

**uicc.hci.test.services.readermode.[Test Area Reference]:** Java Card packages containing Test Area References for the ETSI TS 102 705 [] uicc.hci.services.readermode package.

where the Test Area Reference is written in lower case.

EXAMPLE: The package *../uicc.hci.test.framework.[Test Area Reference]* creates the following directory structure *../uicc/hci/test/framework/[Test Area Reference]/Api\_1\_...\_[1..n].\*,* where '*Api\_1\_...\_[1..n].\*'* are the different test applets Java source files used in *[Test Area Reference]*.

## 5.5 AID coding

The AID coding for the Test Packages, Applet classes and Applets shall be as specified in ETSI TS 101 220 []. In addition, the following TAR and Application Provider specific data values are defined for use within the present document:

AID coding

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte 1 | |  | | Byte 12 | | Byte 13 | | Byte 14 | | Byte 15 | | Byte 16 | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | Application Provider specific data | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | TAR | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  | Specified in ETSI TS 101 220 [] | | | | | |

TAR coding (3 bytes / 24 bits):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| b1 | |  | | b4 | | b5 | |  | | b8 | | b9 | |  | | b16 | |  | |  | |  | | b17 | |  | | b24 | |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Applet class/instance number |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Package number |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Core package identifier |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | Test Part Identifier |

Applet instance number, Applet Class number, Package number:

* For package AID, package number shall start from 0 and class and instance numbers shall be 0.
* For class AID, package number is the number of the class package, class number shall start from 1 and instance shall be 0.
* For instance AID, package and class number are the number of class and package of which instance belongs, and instance number shall start from 1.

Test Part and Core Package Identifier

* 0000 0000 reserved (as TAR= '00.00.00' is reserved for Issuer Security Domain).
* 0010 0001 uicc.hci.test.framework.
* 0010 0100 uicc.hci.test.services.
* 0010 0101 uicc.hci.test.services.cardemulation.
* 0010 0110 uicc.hci.test.services.connectivity.
* 0010 0111 uicc.hci.test.services.readermode.

Application Provider specific data (1 byte):

* '00' for Package.
* '01' for Applet class.
* '02' for Applet Instance.

EXAMPLE: Annex B.

## 5.6 Test equipment

These clauses recommend a minimum specification for each of the items of test equipment referenced in the tests.

### 5.6.1 Test tool

This test tool shall meet the following requirements:

* be able to send and receive commands using the underlying HCI resources (e.g. gates and pipes) defined by the HCI protocol as specified in ETSI TS 102 622 [];
* be able to send and receive the commands correctly on the lower layer; i.e ETSI TS 102 613 [];
* the result of I/O commands shall be presented at the application layer;
* be able to provide results of the tests;
* shall send and/or compare all data specified in test file.

### 5.6.2 Java Software Development Kit

Java Card™ software development kit (SDK) version supported by Java Card 3.0.1 specifications ([], [],[]) is 1.5.

# 6 Test cases

## 6.1 Package uicc.hci.framework

### 6.1.1 Class HCIDevice

#### 6.1.1.1 Method getHCIService

Test Area Reference: Api\_1\_Hdv\_Gsr.

##### 6.1.1.1.1 Conformance requirements

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

public static HCIService getHCIService(short serviceID)

throws HCIException,

javacard.framework.SystemException

6.1.1.1.1.1 Normal execution

* CRRN1: used to retrieve instances that implement one of HCIService subinterfaces defined in the present document.
* CRRN2: returns reference to the permanent JCRE entry point object of the HCIService.
* CRRN3: returns null if not yet registered.

6.1.1.1.1.2 Parameter errors

* CRRP1: javacard.framework.SystemException is thrown with the reason code ILLEGAL\_VALUE if the serviceID does not match with the predefined values.

6.1.1.1.1.3 Context errors

* CRRC1: HCIException thrown with reason HCI\_ACCESS\_NOT\_GRANTED if access to the indicated service is not permitted for the Applet.
* CRRC2: HCIException is thrown with reason HCI\_SERVICE\_NOT\_AVAILABLE if the requested service type is not available.

##### 6.1.1.1.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hdv\_Gsr\_1.java | 1 |
| Api\_1\_Hdv\_Gsr\_2.java | 2 |
| Api\_1\_Hdv\_Gsr\_1.java | 3 |
| Api\_1\_Hdv\_Gsr\_1.java | 4 |
| Api\_1\_Hdv\_Gsr\_1.java | 5 |

##### 6.1.1.1.3 Initial conditions

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI Interface.

##### 6.1.1.1.4 Test procedure

| **Test case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| ID | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Select supported and registered service** | | | | |
| 1 -EVT\_SEND\_DATA (INS = '01') | getHCIService()  serviceID = CARD\_EMULATION\_SERVICE\_ID | No exception shall be thrown | EVT\_SEND\_DATA (SW - '90 00') | N1, N2 |
| 2 - EVT\_SEND\_DATA (INS = '02') | getHCIService()  serviceID = CONNECTIVITY\_SERVICE\_ID | No exception shall be thrown | EVT\_SEND\_DATA (SW - '90 00') | N1, N2 |
| 3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here  - send APDU (INS = '03') | getHCIService()  serviceID = READER\_SERVICE\_ID | No exception shall be thrown | SW - '90 00' | N1, N2 |
| 2 | **Applet not registered** | | | | |
| EVT\_SEND\_DATA (INS = '01') | Applet.register() has not yet been invoked.  getHCIService()  serviceID = CARD\_EMULATION\_SERVICE\_ID | Return: Null | EVT\_SEND\_DATA (SW - '90 00') | N3 |
| 3 | **Select not availabile service** | | | | |
| 1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here  - send APDU (INS = '01') | getHCIService()  serviceID = CardEmulationService | Shall throw uicc.hci.framework.HCIException with error code HCI\_SERVICE\_NOT\_AVAILABLE | SW - '90 01' | C2 |
| 2 - EVT\_SEND\_DATA (INS ='02') | getHCIService()  serviceID = ConnectivityService | Shall throw uicc.hci.framework.HCIException with error code HCI\_SERVICE\_NOT\_AVAILABLE | EVT\_SEND\_DATA (SW - '90 01') | C2 |
| 3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here  - send APDU (INS = '03') | getHCIService()  serviceID = ReaderService | Shall throw uicc.hci.framework.HCIException with error code HCI\_SERVICE\_NOT\_AVAILABLE | SW - '90 01' | C2 |
| 4 | **Select undefined service** | | | | |
| EVT\_SEND\_DATA (INS = '04') | getHCIService()  serviceID = -1 | Shall throw javacard.framework.SystemException with error code ILLEGAL\_VALUE | EVT\_SEND\_DATA (SW - '90 00') | P1 |
| 5 | **Access not granted** | | | | |
| 1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here  - send APDU (INS = '01') | getHCIService()  seviceID = CardEmulationService | Shall throw  uicc.hci.framework.HCIException with error code HCI\_ACCESS\_NOT\_GRANTED | SW - '90 02' | C1 |
| 2 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here  - send APDU (INS='03') | getHCIService()  serviceID =  ReaderService | Shall throw  uicc.hci.framework.HCIException with error code HCI\_ACCESS\_NOT\_GRANTED | SW - '90 02' | C1 |

#### 6.1.1.2 Method getPowerMode

Test Area Reference: Api\_1\_Hdv\_Gpm.

##### 6.1.1.2.1 Conformance requirements

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

public static byte getPowerMode()

6.1.1.2.1.1 Normal execution

- CRRN1: returns the current power mode of the UICC as defined in ETSI TS 102 613 [].

- CRRN2: returns -1 if the power mode cannot be retrieved.

- CRRN3: Applets communicating through the process() method shall also be able to use the API services defined in ETSI TS 102 705 [] which do not require a CardEmulationListener registration (e.g. requesting the power mode or connectivity service).

NOTE: Development of test cases for CRRN2 is FFS.

6.1.1.2.1.2 Parameter errors

* None.

6.1.1.2.1.3 Context errors

* None.

##### 6.1.1.2.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hdv\_Gpm\_1.java | 1 |
| Api\_1\_Hdv\_Gpm\_1.java | 2 |
| Api\_1\_Hdv\_Gpm\_1.java | 3 |

##### 6.1.1.2.3 Initial conditions

* The UICC is not powered up.

##### 6.1.1.2.4 Test procedure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | | | | | |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Verify power mode** | | | | |
| - Activate the UICC in full power mode.  - EVT\_FIELD\_ON  - EVT\_CARD\_ACTIVATED  - EVT\_SEND\_DATA (select applet)  - EVT\_SEND\_DATA(INS='05') | getPowerMode() | return = FULL\_POWER\_MODE | EVT\_SEND\_DATA (SW - '90 03') | N1, N3 |
| 2 | **Verify power mode** | | | | |
| Activate the UICC in low power mode.  - EVT\_FIELD\_ON  - EVT\_CARD\_ACTIVATED  - EVT\_SEND\_DATA (select applet)  - EVT\_SEND\_DATA(INS='05') | getPowerMode() | return = LOW\_POWER\_MODE | EVT\_SEND\_DATA (SW - '90 04') | N1, N3 |
| 3 | **ISO interface is activated, SWP interface is not activated** | | | | |
| Send command on ISO interface to select applet.  - send APDU (INS='01')  - send APDU (INS='05') | getPowerMode() | return = FULL\_POWER\_MODE | - SW - '90 00'  SW - '90 03' | N1  N3 |

#### 6.1.1.3 Method isHCIServiceAvailable

Test Area Reference: Api\_1\_Hdv\_Isa.

##### 6.1.1.3.1 Conformance requirements

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

public static byte isHCIServiceAvailable(short serviceID)

6.1.1.3.1.1 Normal execution

* CRRN1: this method is used to retrieve information about the availability of a HCIService more related to user interface.
* CRRN2: this method returns (byte)0x00 if the service is available and can be used, otherwise bit values set to:
* bit 0 set: interface not supported by terminal;
* bit 1 set: service not supported by terminal;
* bit 2 set: access not allowed for Applet;
* bit 3 set: HCI interface currently disabled in UICC.

6.1.1.3.1.2 Parameter errors

* None.

6.1.1.3.1.3 Context errors

* None.

##### 6.1.1.3.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hdv\_Isa\_1.java | 1 |
| Api\_1\_Hdv\_Isa\_1.java | 2 |
| Api\_1\_Hdv\_Isa\_1.java | 3 |
| Api\_1\_Hdv\_Isa\_1.java | 4 |
| Api\_1\_Hdv\_Isa\_2.java | 5-1 |
| Api\_1\_Hdv\_Isa\_3.java | 5-2 |

##### 6.1.1.3.3 Initial conditions

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI interface.

##### 6.1.1.3.4 Test procedure

| Test Case | | | | | |
| --- | --- | --- | --- | --- | --- |
| ID | HCI/APDU Command | API Description | API Expectation | HCI/APDU Response | CRR |
| 1 | Retrieve information of an available service | | | | |
|  | 1 - EVT\_SEND\_DATA (INS = '06') | isHCIServiceAvailable()  serviceID =  CARD\_EMULATION\_SERVICE\_ID | 0x00 | EVT\_SEND\_DATA (SW ‑ '90 00', data byte set to '00') | N1,  N2 |
|  | 2 - EVT\_SEND\_DATA (INS = '07') | isHCIServiceAvailable()  serviceID =  CONNECTIVITY\_SERVICE\_ID | 0x00 | EVT\_SEND\_DATA (SW ‑ '90 00', data byte set to '00') | N1,  N2 |
|  | 3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.3.1.3 not applicable here  - send APDU INS='08' | isHCIServiceAvailable()  serviceID =  READER\_SERVICE\_ID | 0x00 | SW - '90 00', data byte set to '00' | N1,  N2 |
| 2 | **Interface not supported by terminal** | | | | |
|  | Precondition: SWP interface not supported; the initial conditions in clause 6.1.1.3.1.3 not applicable here, and C6 should be connected to GND:  Sending command on ISO interface:  - select applet  - send APDU (INS = '06') | isHCIServiceAvailable() | bit 0 set | - SW - '90 00'  - SW - '90 00', bit 0 set in the data byte | N2 |
| 3 | **Service not supported by terminal** | | | | |
|  | 1 - Precondition: disable Connectivity service support in terminal; i.e. connectivity gate is not available in terminal pipe creation is not possible and the  terminal shall indicate in TERMINAL PROFILE  HCI Connectivity is not supported.  - EVT\_SEND\_DATA (INS = '07') | isHCIServiceAvailable()  serviceID =  CONNECTIVITY\_SERVICE\_ID | bit 1 set | EVT\_SEND\_DATA (SW ‑ '90 00', bit 1 set in the data byte) | N2 |
|  | 2 - Precondition: disable Reader service support in terminal; i.e. reader RF gates are not available in terminal  Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here  -  - send APDU (INS='08') | isHCIServiceAvailable()  serviceID =  READER\_SERVICE\_ID | bit 1 set | SW - '90 00', bit 1 set in the data byte | N2 |
| 4 | **Access not allowed for Applet** | | | | |
|  | 1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here  - send APDU (INS = '06') | isHCIServiceAvailable()  serviceID =  CARD\_EMULATION\_SERVICE\_ID | bit 2 set | SW - '90 00', bit 2 set in the data byte | N2 |
|  | 2 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here  - send APDU (INS = '08') | isHCIServiceAvailable()  serviceID =  READER\_SERVICE\_ID | bit 2 set | SW - '90 00', bit 2 set in the data byte | N2 |
| 5 | **HCI interface currently disabled in UICC** | | | | |
|  | 1 - Precondition: The terminal shall indicate the support of class r by setting the 26th byte, 'b2' and the 31st byte, 'b1' in the terminal profile and disable the contactless functionality in the UICC as defined in ETSI TS 102 223 [].  The initial conditions in clause 6.1.1.3.3 are not applicable here.  Send on ISO interface the following commands:  - Send APDU to select the applet.  - Send APDU (INS = '06') | isHCIServiceAvailable()  serviceID =  CARD\_EMULATION\_SERVICE\_ID | bit 3 set | SW - '90 00', bit 3 set in the data byte | N2 |
|  | 2 - Precondition: The contactless interface is disabled in the UICC as defined in Global Platform Amendment C.  The initial conditions in clause 6.1.1.3.3 are not applicable here.  Send on ISO interface send the following commands:  - Send APDU to select the applet.  - Send APDU (INS = '06')  - Postcondition:  The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C | setCommunicationInterface() API method of  Global Platform Amendment C [] is used to disable HCI interface  isHCIServiceAvailable()  serviceID =  CARD\_EMULATION\_SERVICE\_ID  setCommunicationInterface() API method of  Global Platform Amendment C [] is used to enable again HCI interface | bit 3 set | SW - '90 00', bit 3 set in the data byte | N2 |

### 6.1.2 Interface HCIService

#### 6.1.2.1 Method register

Test Area Reference: Api\_1\_Hsr\_Reg.

##### 6.1.2.1.1 Conformance requirements

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

void register(HCIListener listener)

throws HCIException

6.1.2.1.1.1 Normal execution

* CRRN1: registers a Listener object to the Service instance.
* CRRN2: a certain listener type can only be registered to the same service type.
* CRRN3: The Registration of Listener Interfaces and activation of events shall be persistent.

6.1.2.1.1.2 Parameter errors

* CRRP1: throws HCIException in case the listener Object registered to the service is not implementing the corresponding interface of the HCIService instance with error code:
* HCIException.HCI\_WRONG\_LISTENER\_TYPE.

6.1.2.1.1.3 Context errors

* CRRC1: throws HCIException with error code HCI\_LISTENER\_ALREADY\_REGISTERED in case a listener Object of the same type has already been registered.

##### 6.1.2.1.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hsr\_Reg\_1.java | 1-1 |
| Api\_1\_Hsr\_Reg\_3.java | 1-2 |
| Api\_1\_Hsr\_Reg\_5.java | 1-3 |
| Api\_1\_Hsr\_Reg\_1.java | 2 |
| Api\_1\_Hsr\_Reg\_2.java | 3-1 |
| Api\_1\_Hsr\_Reg\_4.java | 3-2 |
| Api\_1\_Hsr\_Reg\_6.java | 3-3 |
| Api\_1\_Hsr\_Reg\_1.java | 4 |

##### 6.1.2.1.3 Initial conditions

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI interface.

##### 6.1.2.1.4 Test procedure

|  | | Test case | | | |
| --- | --- | --- | --- | --- | --- |
| ID | HCI Command | Description | API Expectation | HCI Response | CRR |
| 1 | **Register Listener to a service** | | | | |
| 1 - EVT\_SEND\_DATA (INS = '09') | register()  Service = CardEmulationService  Listener = CardEmulationListener | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1 |
| 2 - EVT\_SEND\_DATA (INS = '09') | register()  Service = ConnectivityService  Listener = ConnectivityListener | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1 |
| 3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.1.3 not applicable here  - send APDU (INS='09') | register()  Service = ReaderService  Listener = ReaderListener | No exception shall be thrown | SW - '90 00' | N1 |
| 2 | **Register wrong listener type** | | | | |
| EVT\_SEND\_DATA (INS = '0A') | register()  Service = ConnectivityService  Listener = CardEmulationListener | Shall throw uicc.hci.framework.HCIException with error code HCI\_WRONG\_LISTENER\_TYPE | EVT\_SEND\_DATA (SW ‑ '90 00') | N2, P1 |
| 3 | **Listener already registered** | | | | |
|  | 1 - EVT\_SEND\_DATA (INS = '0B') | register()  Service = CardEmulationService  Listener = CardEmulationListener | Shall throw uicc.hci.framework.HCIException with error code HCI\_LISTENER\_ALREADY\_REGISTERED | EVT\_SEND\_DATA (SW ‑ '90 00') | C1 |
|  | 2 - EVT\_SEND\_DATA (INS = '0B') | register()  Service = ConnectivityService  Listener = ConnectivityListener | Shall throw uicc.hci.framework.HCIException with error code HCI\_LISTENER\_ALREADY\_REGISTERED | EVT\_SEND\_DATA (SW ‑ '90 00') | C1 |
|  | 3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.1.3 not applicable here  - send APDU (INS='0B') | register()  Service = ReaderService  Listener = ReaderListener | Shall throw uicc.hci.framework.HCIException with error code HCI\_LISTENER\_ALREADY\_REGISTERED | SW - '90 00' | C1 |
| 4 | **Persistent Listener Registration** | | | | |
| - Send  EVT\_SEND\_DATA (INS = '09')  - EVT\_FIELD\_OFF  - EVT\_FIELD\_ON  - EVT\_CARD\_ACTIVATED  - Select the applet  - Send EVT\_SEND\_DATA (INS = '09') | register()  Service = CardEmulationService  Listener = CardEmulationListener | Shall throw uicc.hci.framework.HCIException with error code HCI\_LISTENER\_ALREADY\_REGISTERED | EVT\_SEND\_DATA (SW - '6F 01') | N3 |

#### 6.1.2.2 Method deregister

Test Area Reference: Api\_1\_Hsr\_Drg.

##### 6.1.2.2.1 Conformance requirements

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

void deregister(HCIListener listener)

6.1.2.2.1.1 Normal execution

* CRRN1: deregisters a Listener object from the Service instance.
* CRRN2: Calling deregister with an argument that does not identify a currently registered HCIListener has no effect.
* CRRN3: Applets communicating through the process() method shall also be able to use the API services defined in ETSI TS 102 705 [] which do not require a CardEmulationListener registration (e.g. requesting the power mode or connectivity service).

6.1.2.2.1.2 Parameter errors

* None.

6.1.2.2.1.3 Context errors

* None.

##### 6.1.2.2.2 Test Suite Files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hsr\_Drg\_1.java | 1 |
| Api\_1\_Hsr\_Drg\_2.java | 2 |

##### 6.1.2.2.3 Initial conditions

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI interface.

##### 6.1.2.2.4 Test procedure

| Test case | | | | | |
| --- | --- | --- | --- | --- | --- |
| ID | HCI Command | API Description | API Expectation | HCI Response | CRR |
| 1 | **Deregister Listener to a Service - CardEmulation** | | | | |
|  | 1 - EVT\_SEND\_DATA (INS = '01') | register()  Service = CardEmulationService  Listener = CardEmulationListener  deregister()  Service = CardEmulationService  Listener = CardEmulationListener | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1,  N3 |
|  | 2 - EVT\_SEND\_DATA (INS = '02') | deregister()  Service = ConnectivityService  Listener = CardEmulationListener | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N2 |
|  | 3 - EVT\_SEND\_DATA (INS = '03') | deregister()  Service = CardEmulationService  Listener = CardEmulationListener | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N2,  N3 |
| 2 | **Deregister Listener to a Service - Connectivity** | | | | |
| 1 - EVT\_SEND\_DATA (INS = '01') | register()  Service = ConnectivityService  Listener = ConnectivityListener  deregister()  Service = ConnectivityService  Listener = ConnectivityListener | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1 |
| 2 - EVT\_SEND\_DATA (INS = '02') | deregister()  Service = CardEmulationService  Listener = ConnectivityListener | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N2 |
| 3 - EVT\_SEND\_DATA (INS = '03') | deregister()  Service = ConnectivityService  Listener = ConnectivityListener | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N2 |

#### 6.1.2.3 Method activateEvent

Test Area Reference: Api\_1\_Hsr\_Ace.

##### 6.1.2.3.1 Conformance requirements

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

void activateEvent(byte event)

throws HCIException

6.1.2.3.1.1 Normal execution

* CRRN1: This method is used to activate an event.
* CRRN2: for all service interfaces HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED is sent to notify the applet when the UICC failed to transmit a message sent by CLF. (See NOTE.)
* CRRN3: for all service interfaces HCIListener.EVENT\_HCI\_RECEPTION\_FAILED is sent to notify the applet when the UICC failed to receive a message sent by CLF. (See NOTE.)
* CRRN4: Possible values for event for Card emulation service are:
* EVENT\_GET\_PARAMETER\_RESPONSE.
* EVENT\_ON\_SEND\_DATA.
* EVENT\_FIELD\_OFF.
* CRRN5: Possible values for event for Reader service are:
* EVENT\_GET\_PARAMETER\_RESPONSE.
* EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE.
* EVENT\_TARGET\_DISCOVERED.
* CRRN6: Possible value for event for Connectivity service is EVENT\_STAND\_BY.
* CRRN7: The Registration of Listener Interfaces and activation of events shall be persistent.
* CRRN8: When the contactless interface is disabled (cf. "state of contactless functionality" in ETSI TS 102 223 [] and setCommunicationInterface() API method of "GlobalPlatform Amendment C" []), the Contactless Framework shall throw an HCIException with reason code HCI\_CURRENTLY\_DISABLED.
* CRRN9: The Contactless Framework shall raise an EVENT\_FIELD\_OFF if this event is activated for this Applet instance, before the invocation of the deselect() method of the Applet instance.
* CRRN10: After the EVENT\_FIELD\_OFF event the Applet instance shall not be triggered by any other event until the Applet instance is selected again.
* CRRN11: If the current application protocol is APDU based the HCI framework shall handle an application session termination according to ETSI TS 102 221 [] independent of the interface used for message exchange.
* CRRN12: Applet selection and deselection shall be performed by the Contactless Framework according to the rules defined in the "Java Card™ Runtime Environment Specification, 3.0 Classic Edition" [] and in "GlobalPlatform Amendment C" [].
* CRRN13: The select() method of the Applet instance shall always be invoked for an Applet selection according to the rules given in "Java Card™ Runtime Environment Specification, 3.0 Classic Edition" [].
* CRRN14: If the HCI event EVT\_FIELD\_OFF or EVT\_CARD\_DEACTIVATED defined by the HCI protocol as specified in ETSI TS 101 220 [] is received by the Contactless Framework and the UICC is still powered, the Applet instance shall be deselected according to "GlobalPlatform Amendment C" [].

NOTE: HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED and HCIListener.EVENT\_HCI\_RECEPTION\_FAILED are deprecated in TS 102 705 [1]. However, as existing applets may still invoke the method with these events as a parameter, the invocation of the method with these events as a parameter is still tested.

6.1.2.3.1.2 Parameter errors

- CRRP1: HCIException with reason code HCI\_WRONG\_EVENT\_TYPE if a wrong event was activated for this service instance.

- CRRP2: HCIException with reason code HCI\_CURRENTLY\_DISABLED if the interface to the contactless frontend (CLF) is currently disabled and the event is ReaderListener.EVENT\_TARGET\_DISCOVERED.

- CRRP3: HCIException with reason code HCI\_CONDITIONS\_NOT\_SATISFIED if one or more conditions to activate the event are not satisfied.

NOTE: Development of test cases for CRRP3 is FFS.

6.1.2.3.1.3 Context errors

* None.

##### 6.1.2.3.2 Test Suite Files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hsr\_Ace\_1.java | 1 |
| Api\_1\_Hsr\_Ace\_3.java | 2 |
| Api\_1\_Hsr\_Ace\_2.java | 3 |
| Api\_1\_Hsr\_Ace\_1.java | 4 |
| Api\_1\_Hsr\_Ace\_7.java | 5-1 |
| Api\_1\_Hsr\_Ace\_9.java | 5-2 |
| Api\_1\_Hsr\_Ace\_4.java | 6 |
| Api\_1\_Hsr\_Ace\_5.java | 7 |
| Api\_1\_Hsr\_Ace\_6.java | 8 |
| Api\_1\_Hsr\_Ace\_8.java | 9 |
| Api\_1\_Hsr\_Ace\_6.java | 10 |

##### 6.1.2.3.3 Initial conditions

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI Interface.

##### 6.1.2.3.4 Test procedure

| **Test Case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI Command** | **Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Event activation - CardEmulationService** | | | | |
|  | 1 - EVT\_SEND\_DATA (INS = '10') | HCIListener = CardEmulationListener  HCIService = CardEmulationService  activateEvent()  event = HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1,  N2 |
|  | 2 - EVT\_SEND\_DATA (INS = '11') | HCIListener = CardEmulationListener  HCIService = CardEmulationService  activateEvent()  event = HCIListener.EVENT\_HCI\_RECEPTION\_FAILED | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1,  N3 |
|  | 3 - EVT\_SEND\_DATA (INS = '12') | HCIListener = CardEmulationListener  HCIService = CardEmulationService  activateEvent()  event = CardEmulationListener.EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1,  N4 |
|  | 4 - EVT\_SEND\_DATA (INS = '14') | HCIListener = CardEmulationListener  HCIService = CardEmulationService  activateEvent()  event =  CardEmulationListener.EVENT\_ON\_SEND\_DATA | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1,  N4 |
| 2 | **Event activation - ReaderService** | | | | |
| 1- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.3.3 not applicable here  - Send APDU INS='10' | HCIService = ReaderService  activateEvent()  event = HCIListener.EVENT\_HCI\_RECEPTION\_FAILED | No exception shall be thrown. | SW - '90 00' | N1,  N3 |
| 2- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.3.3 not applicable here  - Send APDU INS='11' | HCIService = ReaderService  activateEvent()  event = EVENT\_TARGET\_DISCOVERED | No exception shall be thrown. | SW - '90 00' | N1,  N5 |
| 3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.3.3 not applicable here  - Send APDU INS='12' | HCIService = ReaderService  activateEvent()  event = EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE | No exception shall be thrown. | SW - '90 00' | N1,  N5 |
| 4 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.3.3 not applicable here  - Send APDU INS='13' | HCIService = ReaderService  activateEvent()  event = HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED | No exception shall be thrown. | SW - '90 00' | N1,  N2. |
| 5 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.3.3 not applicable here  - Send APDU INS='14' | HCIService = ReaderService  activateEvent()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown. | SW - '90 00' | N1,  N5 |
| 3 | **Event activation - ConnectivityListener** | | | | |
| 1 - EVT\_SEND\_DATA (INS = '10') | HCIListener = ConnectivityListener  HCIService = ConnectivityService  activateEvent()  event = HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED | No exception shall be thrown. | EVT\_SEND\_DATA (SW ‑ '90 00') | N1,  N2 |
| 2 - EVT\_SEND\_DATA (INS = '11') | HCIListener = ConnectivityListener  HCIService = ConnectivityService  activateEvent()  event = HCIListener.EVENT\_HCI\_RECEPTION\_FAILED | No exception shall be thrown. | EVT\_SEND\_DATA (SW ‑ '90 00') | N1,  N3 |
| 3 - EVT\_SEND\_DATA (INS = '12') | HCIListener = ConnectivityListener  HCIService = ConnectivityService  activateEvent()  event = EVENT\_STAND\_BY | No exception shall be thrown. | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N6 |
| 4 | **Wrong event type** | | | | |
| EVT\_SEND\_DATA (INS = '16') | HCIListener = CardEmulationListener  HCIService = CardEmulationService  activateEvent()  event value = 0x02 | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | P1 |
| 5 | **HCI is disabled** | | | | |
| 1- Precondition:  The terminal shall indicate the support of class r by setting the 26th byte, 'b2' and the 31st byte, 'b1' in the terminal profile and disable the contactless functionality in the UICC as defined in ETSI TS 102 223 [].  - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.3.3 not applicable here - - Send APDU  (INS = '01') on ISO interface | HCIService = ReaderService  activateEvent()  event = ReaderListener.EVENT\_TARGET\_DISCOVERED | HCIException with reason code HCI\_CURRENTLY\_DISABLED shall be thrown | SW- '90 00' | P2, N8 |
|  | 2 - Precondition: The contactless interface is disabled in the UICC as defined in Global Platform Amendment C.  - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.3.3 not applicable here  - Send APDU (INS = '01') on ISO interface  - Postcondition:  The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C | setCommunicationInterface() API method of "GlobalPlatform Amendment C" [] used to disable HCI interface  HCIService = ReaderService  activateEvent()  event = ReaderListener.EVENT\_TARGET\_DISCOVERED  setCommunicationInterface() API method of  Global Platform Amendment C [] is used to enable again HCI interface | HCIException with reason code HCI\_CURRENTLY\_DISABLED shall be thrown | SW- '90 00' | P2, N8 |
| 6 | **Persistent Event Activation** | | | | |
| - Send  EVT\_SEND\_DATA (INS = '01')  - EVT\_FIELD\_OFF  - Power off  - Power on  - EVT\_FIELD\_ON  - EVT\_CARD\_ACTIVATED  - Select the applet  - Send EVT\_SEND\_DATA (INS = '02') | Service = CardEmulationService  Listener = CardEmulationListener  activateEvent(EVENT\_FIELD\_OFF) | No exception shall be thrown.  getEventNotificationStatus() = true | - EVT\_SEND\_DATA (SW ‑ '90 00')  -EVT\_SEND\_DATA (SW ‑ '90 00') | N7 |
| 7 | **Event Field Off before deselect** | | | | |
| - EVT\_FIELD\_OFF  - EVT\_FIELD\_ON  - EVT\_CARD\_ACTIVATED  - Select the applet  - Send EVT\_SEND\_DATA (INS = '01') | Service = CardEmulationService  Listener = CardEmulationListener  activateEvent(EVENT\_ON\_SEND\_DATA) during the installation  onCallback()  activateEvent()  event = EVENT\_FIELD\_OFF  deselect() | EVENT\_FIELD\_OFF raised before  deselect() | EVT\_SEND\_DATA (SW ‑ '90 00') | N9, N11, N12,  N14 |
| 8 | **No triggering after Event Field Off** | | | | |
| EVT\_FIELD\_OFF  - Send EVT\_SEND\_DATA (INS = '01') | Service = CardEmulationService  Listener = CardEmulationListener  activateEvent(EVENT\_ON\_SEND\_DATA) during the installation  onCallback()  activateEvent()  event = EVENT\_FIELD\_OFF | EVENT\_ON\_SEND\_DATA shall not be raised | No response, or EVT\_SEND\_DATA with any response except:  (SW - '90 00') | N1,  N4,  N10, N11 |
| 9 | **Check for Selection** | | | | |
| Send EVT\_SEND\_DATA (INS = '01') | Service = CardEmulationService  Listener = CardEmulationListener | Check for invocation of select() | EVT\_SEND\_DATA (SW - '90 00') | N12, N13 |
| 10 | **No triggering after Event Card Deactivate** | | | | |
| EVT\_CARD\_DEACTIVATE  - Send EVT\_SEND\_DATA (INS = '01') | Service = CardEmulationService  Listener = CardEmulationListener  activateEvent(EVENT\_ON\_SEND\_DATA) during the installation  onCallback()  activateEvent()  event = EVENT\_FIELD\_OFF | EVT\_SEND\_DATA shall not be raised | No response or EVT\_SEND\_DATA withany response except:  (SW - '90 00' or '6D 00') | N12,  N14 |

#### 6.1.2.4 Method deactivateEvent

Test Area Reference: Api\_1\_Hsr\_Dae.

##### 6.1.2.4.1 Conformance requirements

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

void deactivateEvent(byte event)

throws HCIException

6.1.2.4.1.1 Normal execution

* CRRN1: deactivates an event from the list of activated events.
* CRRN2: for all service interfaces HCIListenr.EVENT\_HCI\_TRANSMISSION\_FAILED is sent to notify the applet when the UICC failed to transmit a message sent by CLF. (See NOTE.)
* CRRN3: for all service interfaces HCIListenr HCIListener.EVENT\_HCI\_RECEPTION\_FAILED is sent to notify the applet when the UICC failed to receive a message sent by CLF. (See NOTE.)
* CRRN4: Possible values for parameter event for Card emulation service are:
* EVENT\_GET\_PARAMETER\_RESPONSE.
* EVENT\_ON\_SEND\_DATA.
* EVENT\_FIELD\_OFF.
* CRRN5: Possible values for parameter event for Reader service are:
* EVENT\_GET\_PARAMETER\_RESPONSE.
* EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE.
* EVENT\_TARGET\_DISCOVERED.
* CRRN6: Possible values for Connectivity service is EVENT\_STANDBY.

NOTE: HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED and HCIListener.EVENT\_HCI\_RECEPTION\_FAILED are deprecated in TS 102 705 [1]. However, as existing applets may still invoke the method with these events as a parameter, the invocation of the method with these events as a parameter is still tested.

6.1.2.4.1.2 Parameter errors

* CRRP1: HCIException with reason code HCIException.HCI\_WRONG\_EVENT\_TYPE in case a wrong event was deactivated for this service instance.
* CRRP2: HCIException with reason code.HCI\_WRONG\_EVENT\_TYPE in case the event wasn't activated before.

6.1.2.4.1.3 Context errors

* None.

##### 6.1.2.4.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hsr\_Dae\_1.java | 1-1 |
| Api\_1\_Hsr\_Dae\_1.java | 1-2 |
| Api\_1\_Hsr\_Dae\_2.java | 1-3 |
| Api\_1\_Hsr\_Dae\_2.java | 1-4 |
| Api\_1\_Hsr\_Dae\_5.java | 2 |
| Api\_1\_Hsr\_Dae\_4.java | 3 |
| Api\_1\_Hsr\_Dae\_1.java | 4 |
| Api\_1\_Hsr\_Dae\_3.java | 5 |
| Api\_1\_Hsr\_Dae\_6.java | 6 |
| Api\_1\_Hsr\_Dae\_7.java | 7 |

##### 6.1.2.4.3 Initial conditions

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI interface.

##### 6.1.2.4.4 Test procedure

| **Test case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Event deactivation - CardEmulationService** | | | | |
| 1 - EVT\_SEND\_DATA(INS='10') | HCIService = CardEmulationService  Event has been successfully activated  deactivateEvent()  event = HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N2 |
| 2 - EVT\_SEND\_DATA(INS='11') | HCIService = CardEmulationService  Event has been successfully activated  deactivateEvent()  event =  HCIListener.EVENT\_HCI\_RECEPTION\_FAILED | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N3 |
| 3 - EVT\_SEND\_DATA(INS='12') | HCIService = CardEmulationService  Event has been successfully activated  deactivateEvent()  event =  CardEmulationListener.EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N4 |
| 4 - EVT\_SEND\_DATA(INS='14') | HCIService = CardEmulationService  Event has been successfully activated  deactivateEvent()  event =  CardEmulationListener.EVENT\_ON\_SEND\_DATA | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N4 |
| 2 | **Event deactivation - ReaderService** | | | | |
|  | 1- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.3.3 not applicable here  - Send APDU (INS='09') | HCIService = ReaderService  deactivateEvent()  event = HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED | No exception shall be thrown | SW - '90 00' | N1, N2 |
|  | 2 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here  - Send APDU (INS='10') | HCIListener = ReaderListener  HCIService = ReaderService  Event has been successfully activated  deactivateEvent()  event = HCIListener. EVENT\_HCI\_TRANSMISSION\_FAILED | No exception shall be thrown | SW - '90 00' | N1, N3 |
|  | 3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here  - Send APDU (INS='11') | HCIListener = ReaderListener  HCIService = ReaderService  Event has been successfully activated  deactivateEvent()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown | SW - '90 00' | N1, N5 |
|  | 4 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here  - Send APDU (INS='12') | HCIListener = ReaderListener  HCIService = ReaderService  deactivateEvent()  event = EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE | No exception shall be thrown | SW - '90 00' | N1, N5 |
|  | 5- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here  - Send APDU (INS='13') | HCIListener = ReaderListener  HCIService = ReaderServiceEvent has been successfully activated  deactivateEvent()  event = EVENT\_TARGET\_DISCOVERED | No exception shall be thrown | SW - '90 00' | N1, N5 |
| 3 | **Event deactivation - ConnectivityService** | | | | |
|  | 1 - EVT\_SEND\_DATA(INS='10') | HCIListener = ConnectivityListener  HCIService = ConnectivityService  Event has been successfully activated  deactivateEvent()  event = HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N2 |
|  | 2 -  EVT\_SEND\_DATA(INS='11') | HCIListener = ConnectivityListener  HCIService = ConnectivityService  Event has been successfully activated  deactivateEvent()  event = HCIListener.EVENT\_HCI\_RECEPTION\_FAILED | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N3 |
|  | 3 -  EVT\_SEND\_DATA(INS='12') | HCIListener = ConnectivityListener  HCIService = ConnectivityService  Event has been successfully activated  deactivateEvent()  event = EVENT\_STAND\_BY | No exception shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N6 |
| 4 | **Wrong event type** | | | | |
| EVT\_SEND\_DATA (INS = '16') | HCIService = CardEmulationService  Event has been successfully activated  deactivateEvent()  event value = 02 | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | P1 |
| 5 | **Deactivate an non-active event - CardEmulationService** | | | | |
|  | 1 - EVT\_SEND\_DATA (INS = '10') | HCIService = CardEmulationService  Event was not activated before  deactivateEvent()  event =  HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | P2 |
|  | 2 - EVT\_SEND\_DATA (INS = '11') | HCIService = CardEmulationService  Event was not activated before  deactivateEvent()  event =  HCIListener.EVENT\_HCI\_RECEPTION\_FAILED | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | P2 |
|  | 3 - EVT\_SEND\_DATA (INS = '12') | HCIService = CardEmulationService  Event was not activated before  deactivateEvent()  event =  CardEmulationListener.EVENT\_GET\_PARAMETER\_RESPONSE | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | P2 |
|  | 4 -EVT\_SEND\_DATA (INS = '13') | HCIService = CardEmulationService  Event was not activated before  deactivateEvent()  event =  CardEmulationListener.EVENT\_FIELD\_OFF | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | P2 |
|  | 5- EVT\_SEND\_DATA (INS = '14') | HCIService = CardEmulationService  Event was not activated before  deactivateEvent()  event =  CardEmulationListener.EVENT\_ON\_SEND\_DATA | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | P2 |
| 6 | **Deactivate an non-active event - ConnectivityService** | | | | |
| EVT\_SEND\_DATA(INS='12') | HCIListener = ConnectivityListener  HCIService = ConnectivityService  Event was not activated before  deactivateEvent()  event = EVENT\_STAND\_BY | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | P2 |
| 7 | **Deactivate an non-active event - ReaderService** | | | | |
|  | 1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here  - Send APDU (INS='10') | HCIListener = ReaderListener  HCIService = ReaderService  Event was not activated before  deactivateEvent()  event = EVENT\_GET\_PARAMETER\_RESPONSE | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | P2 |
|  | 2 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here  - Send APDU (INS='11') | HCIListener = ReaderListener  HCIService = ReaderService  Event was not activated before  deactivateEvent()  event = EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | P2 |
|  | 3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.4.3 not applicable here  - Send APDU (INS='12') | HCIListener = ReaderListener  HCIService = ReaderService  Event was not activated before  deactivateEvent()  event = EVENT\_TARGET\_DISCOVERED | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | P2 |

#### 6.1.2.5 Method requestCallbackNotification

Test Area Reference: Api\_1\_Hsr\_Rcn.

##### 6.1.2.5.1 Conformance requirements

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

void requestCallbackNotification(byte event)

throws HCIException

6.1.2.5.1.1 Normal execution

* CRRN1: the call of this method throws HCIException with reason code HCI\_ACCESS\_NOT\_GRANTED.

6.1.2.5.1.2 Parameter errors

* None.

6.1.2.5.1.3 Context errors

* None.

##### 6.1.2.5.2 Test Suite Files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hsr\_Rcn\_7.java | 1 |

##### 6.1.2.5.3 Initial conditions

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI interface.

##### 6.1.2.5.4 Test procedure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test case | | | | | |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Request Callback Notification** | | | | |
| EVT\_SEND\_DATA (INS = '10') | Precondition:  HCIService = CardEmulationService  Corresponding listener registered  No event is activated  requestCallbackNotification ()  event = HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED | throws HCIexception with reason code  HCI\_ACCESS\_NOT\_GRANTED | EVT\_SEND\_DATA (SW ‑ '90 00') | N1 |

#### 6.1.2.6 Method getEventNotificationStatus

Test Area Reference: Api\_1\_Hsr\_Gen.

##### 6.1.2.6.1 Conformance requirements

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

boolean getEventNotificationStatus(byte event)

throws HCIException

6.1.2.6.1.1 Normal execution

* CRRN1: return the activation state of an event; if true the event is activated for this listener, else false.
* CRRN2: For CardEmulationService:EVENT\_GET\_PARAMETER\_RESPONSE, EVENT\_SEND\_DATA, and EVENT\_FIELD\_OFF.
* CRRN3: For ReaderService: EVENT\_GET\_PARAMETER\_RESPONSE, EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE, and EVENT\_TARGET\_DISCOVERED.
* CRRN4: For ConnectivityService: ConnectivityListener.EVENT\_STANDBY.
* CRRN5: For all service interfaces: HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED HCIListener.EVENT\_HCI\_RECEPTION\_FAILED. (See NOTE.)

NOTE: HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED and HCIListener.EVENT\_HCI\_RECEPTION\_FAILED are deprecated in TS 102 705 [1]. However, as existing applets may still invoke the method with these events as a parameter, the invocation of the method with these events as a parameter is still tested.

6.1.2.6.1.2 Parameter errors

* CRRP1: HCIException with reason code HCIException.HCI\_WRONG\_EVENT\_TYPE in case the event was not one of the possible values.

6.1.2.6.1.3 Context errors

* None.

##### 6.1.2.6.2 Test Suite Files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hsr\_Gen\_1.java | 1 |
| Api\_1\_Hsr\_Gen\_1.java | 2 |
| Api\_1\_Hsr\_Gen\_3.java | 3 |
| Api\_1\_Hsr\_Gen\_3.java | 4 |
| Api\_1\_Hsr\_Gen\_1.java | 5-1 |
| Api\_1\_Hsr\_Gen\_3.java | 5-2 |
| Api\_1\_Hsr\_Gen\_2.java | 5-3 |
| Api\_1\_Hsr\_Gen\_2.java | 6 |
| Api\_1\_Hsr\_Gen\_2.java | 7 |

##### 6.1.2.6.3 Initial conditions

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI Interface.

##### 6.1.2.6.4 Test procedure

| **Test case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Event notification status false - CardEmulationService** | | | | |
| 1 - EVT\_SEND\_DATA (INS = '10') | HCIService = CardEmulationService  getEventNotificationStatus()  event =  HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED | No exception shall be thrown.  getEventNotificationStatus() = false | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N5 |
| 2 - EVT\_SEND\_DATA (INS = '11') | HCIService = CardEmulationService  getEventNotificationStatus()  event =  HCIListener.EVENT\_HCI\_RECEPTION\_FAILED | No exception shall be thrown.  getEventNotificationStatus() = false | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N5 |
| 3 - EVT\_SEND\_DATA (INS = '12') | HCIService = CardEmulationService  getEventNotificationStatus()  event =  CardEmulationListener.EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown.  getEventNotificationStatus() = false | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N2 |
| 4 - EVT\_SEND\_DATA (INS = '13') | 4 HCIService = CardEmulationService  getEventNotificationStatus()  event =  CardEmulationListener. EVENT\_FIELD\_OFF | No exception shall be thrown.  getEventNotificationStatus() = false | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N2 |
| 5 - EVT\_SEND\_DATA (INS = '14') | HCIService = CardEmulationService  getEventNotificationStatus()  event =  CardEmulationListener. EVENT\_ON\_SEND\_DATA | No exception shall be thrown.  getEventNotificationStatus() = false | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N2 |
| 2 | **Event notification status true - CardEmulationService** | | | | |
| EVT\_SEND\_DATA (INS = '15') | HCIService = CardEmulationService  activateEvent()  event =  CardEmulationListener.EVENT\_ON\_SEND\_DATA  getEventNotificationStatus ()  event =  CardEmulationListener.EVENT\_ON\_SEND\_DATA | No exception shall be thrown.  getEventNotificationStatus() = true | EVT\_SEND\_DATA (SW ‑ '90 00') | N1, N2 |
| 3 | **Event notification status true - ReaderService** | | | | |
|  | 1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here  - Send APDU (INS = '21') | HCIService = ReaderService  activateEvent()  event =  ReaderListener. EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE  getEventNotificationStatus()  event =  ReaderListener. EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE | No exception shall be thrown.  getEventNotificationStatus() = true | SW - '90 00' | N1, N3 |
|  | 2 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here  - Send APDU (INS = '22') | HCIService = ReaderService  activateEvent()  event =  ReaderListener.EVENT\_TARGET\_DISCOVERED  getEventNotificationStatus()  event = ReaderListener.EVENT\_TARGET\_DISCOVERED | No exception shall be thrown.  getEventNotificationStatus() = true | SW - '90 00' | N1, N3 |
|  | 3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here  - Send APDU (INS = '23') | HCIService = ReaderService  activateEvent()  event =  ReaderListener. EVENT\_GET\_PARAMETER\_RESPONSE  getEventNotificationStatus()  event =  ReaderListener. EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown.  getEventNotificationStatus() = true | SW - '90 00' | N1, N3 |
|  | 4 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here  - Send APDU (INS = '24') | HCIService = ReaderService  activateEvent()  event =  ReaderListener. EVENT\_HCI\_RECEPTION\_FAILED  getEventNotificationStatus()  event =  ReaderListener. EVENT\_HCI\_RECEPTION\_FAILED | No exception shall be thrown.  getEventNotificationStatus() = true | SW - '90 00' | N1, N5 |
|  | 5- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here  - Send APDU (INS = '25') | HCIService = ReaderService  activateEvent()  event =  ReaderListener. EVENT\_HCI\_TRANSMISSION\_FAILED  getEventNotificationStatus()  event =  ReaderListener. EVENT\_HCI\_TRANSMISSION\_FAILED | No exception shall be thrown.  getEventNotificationStatus() = true | SW - '90 00' | N1, N5 |
| 4 | **Event notification status false - ReaderService** | | | | |
|  | 1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here  - Send APDU (INS='10') | HCIListener = ReaderListener  HCIService = ReaderServiceEvent has been successfully activated  deactivateEvent()  event = HCIListener. EVENT\_HCI\_TRANSMISSION\_FAILED | No exception shall be thrown.  getEventNotificationStatus() = false | SW - '90 00' | N1,  N5 |
|  | 2 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here  - Send APDU (INS = '11') | HCIListener = ReaderListener  HCIService = ReaderServiceEvent has been successfully activated  deactivateEvent()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown.  getEventNotificationStatus() = false | SW - '90 00' | N1,  N3 |
|  | 3 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here  - Send APDU (INS='12') | HCIListener = ReaderListener  HCIService = ReaderServiceEvent has been successfully activated  deactivateEvent()  event = EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE | No exception shall be thrown.  getEventNotificationStatus() = false | SW - '90 00' | N1,  N3 |
|  | 4- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here  - Send APDU (INS = '13') | HCIListener = ReaderListener  HCIService = ReaderServiceEvent has been successfully activated  deactivateEvent()  event = EVENT\_TARGET\_DISCOVERED | No exception shall be thrown.  getEventNotificationStatus() = false | SW - '90 00' | N1,  N3 |
|  | 5- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here  - Send APDU (INS='14') | HCIListener = ReaderListener  HCIService = ReaderServiceEvent has been successfully activated  deactivateEvent()  event = EVENT\_HCI\_RECEPTION\_FAILED | No exception shall be thrown.  getEventNotificationStatus() = false | SW - '90 00' | N1,  N3 |
| 5 | **Wrong event type** | | | | |
| 1- EVT\_SEND\_DATA (INS = '16') | HCIService = CardEmulationService  getEventNotificationStatus()  event = 0x02 | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | EVT\_SEND\_DATA (SW ‑ '90 00') | P1 |
| 2- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.3 not applicable here  - Send APDU (INS = '16') | HCIService = ReaderService  getEventNotificationStatus()  event = 0x84 | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | SW - '90 00' | P1 |
| 3 - EVT\_SEND\_DATA (INS = '16') | HCIService = ConnectivityService  getEventNotificationStatus()  event = 0x02 | HCIException with reason code HCI\_WRONG\_EVENT\_TYPE shall be thrown | EVT\_SEND\_DATA (SW - '90 00') | P1 |
| 6 | **Event notification status false - ConnectivityService** | | | | |
| 1 - EVT\_SEND\_DATA (INS = '10') | HCIListener = ConnectivityListener  HCIService = ConnectivityService  getEventNotificationStatus()  event =  HCIListener.EVENT\_HCI\_TRANSMISSION\_FAILED | No exception shall be thrown.  getEventNotificationStatus() = false | EVT\_SEND\_DATA (SW - '90 00') | N1, N5 |
| 2 - EVT\_SEND\_DATA (INS = '11') | HCIListener = ConnectivityListener  HCIService = ConnectivityService  getEventNotificationStatus()  event =  HCIListener.EVENT\_HCI\_RECEPTION\_FAILED | No exception shall be thrown.  getEventNotificationStatus() = false | EVT\_SEND\_DATA (SW - '90 00') | N1, N5 |
| 3 - EVT\_SEND\_DATA (INS = '12') | HCIListener = ConnectivityListener  HCIService = ConnectivityService  getEventNotificationStatus()  event = ConnectivityListener.EVENT\_STAND\_BY | No exception shall be thrown.  getEventNotificationStatus() = false | EVT\_SEND\_DATA (SW - '90 00') | N1, N4 |
| 7 | **Event notification status true - ConnectivityService** | | | | |
| EVT\_SEND\_DATA (INS = '15') | HCIListener = ConnectivityListener  HCIService = ConnectivityService  activateEvent()  event = ConnectivityListener.EVENT\_STAND\_BY  getEventNotificationStatus()  event = ConnectivityListener.EVENT\_STAND\_BY | No exception shall be thrown.  getEventNotificationStatus() = true | EVT\_SEND\_DATA (SW - '90 00') | N1, N4 |

### 6.1.3 Interface HCIMessage

#### 6.1.3.1 Method isHeading

Test Area Reference: Api\_1\_Hme\_Mhd.

##### 6.1.3.1.1 Conformance requirements

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

boolean isHeading()

6.1.3.1.1.1 Normal execution

- CRRN1: Checks if the current content is the heading part of an HCP message The heading information indicates the first part of an HCP message; with the isComplete() method it is possible to check whether the complete message has already been received.

- CRRN2: returns true if the current message is the heading part of the message.

- CRRN3: An HCIMessage object shall encapsulate one HCP message according to the HCI protocol as specified in ETSI TS 102 622 [].

- CRRN4: HCI message for the different contactless modes shall be identified by different types of interfaces.

* CRRN5: In case the Applet instance has registered the CardEmulationListener and has activated the EVENT\_ON\_SEND\_DATA the process() method of this Applet instance shall not be invoked during the selection.

NOTE: Development of test cases for CRRN1 in the circumstance that isHeading() returns false is FFS.

6.1.3.1.1.2 Parameter errors

* None.

6.1.3.1.1.3 Context errors

* None.

##### 6.1.3.1.2 Test Suite Files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hme\_Mhd\_1.java | 1 |

##### 6.1.3.1.3 Initial conditions

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI Interface.

##### 6.1.3.1.4 Test procedure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | | | | | |
| ID | HCI Command | API Description | API Expectation | HCI Response | CRR |
| 1 | **message is complete** | | | | |
| EVT\_SEND\_DATA (INS ='01') | onCallback()  event = EVENT\_ON\_SEND\_DATA  HCIMessage.isHeading() | No exception shall be thrown.  isHeading() = true | EVT\_SEND\_DATA (SW - '90 00')  Result returned in the first response byte of the R-APDU data: b1 is set | N1,  N2, N3,  N4, N5 |

#### 6.1.3.2 Method isComplete

Test Area Reference: Api\_1\_Hme\_Mco.

##### 6.1.3.2.1 Conformance requirements

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

boolean isComplete()

6.1.3.2.1.1 Normal execution

- CRRN1: this method checks for the completeness of the HCP message, i.e. the last received HCP packet had the chaining bit CB set to 1. It returns true if the message is complete.

- CRRN2: the framework shall copy data into the receive buffer up to the end of the buffer. When the HCI message is longer than the available buffer length the HCIMessage shall be set as not complete.

NOTE 1: Development of test cases for CRRN1 in the circumstance that isComplete () returns false is FFS.

NOTE 2: Development of test cases for CRRN2 is FFS.

6.1.3.2.1.2 Parameter errors

* None.

6.1.3.2.1.3 Context errors

* None.

##### 6.1.3.2.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hme\_Mco\_1.java | 1 |

##### 6.1.3.2.3 Initial condition

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI Interface.

##### 6.1.3.2.4 Test procedure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | | | | | |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **message is complete** | | | | |
| EVT\_SEND\_DATA (INS = '01') | onCallback()  event = EVENT\_ON\_SEND\_DATA  HCIMessage.isComplete() | No exception shall be thrown.  isComplete() = true | EVT\_SEND\_DATA (SW - '90 00')  Result returned in the second response byte of the R-APDU data: b1 is set | N1 |

#### 6.1.3.3 Method getType

Test Area Reference: Api\_1\_Hme\_Mty.

##### 6.1.3.3.1 Conformance requirements

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

byte getType()

6.1.3.3.1.1 Normal execution

* CRRN1: Returns the type of the incoming HCI message, it shall be one of constant values TYPE\_defined in this interface.The returned type possibilities are: TYPE\_COMMAND Indicates the HCI message type "command", TYPE\_EVENT Indicates the HCI message type "event" and TYPE\_RESPONSE indicates the HCI message type "response".
* CRRN2: In the case of a fragmented incoming message this method shall return the HCI message type coded in the first part of the HCI message.

6.1.3.3.1.2 Parameter errors

* None.

6.1.3.3.1.3 Context errors

* None.

##### 6.1.3.3.2 Test Suite Files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hme\_Mty\_1.java | 1 |
| Api\_1\_Hme\_Mty\_2.java | 2 |

##### 6.1.3.3.3 Initial condition

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI interface.

##### 6.1.3.3.4 Test procedure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | | | | | |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **message of type event - Card emulation** | | | | |
| EVT\_SEND\_DATA (INS = '01') | onCallback()  event = EVENT\_ON\_SEND\_DATA  HCIMessage.getType() | No exception shall be thrown.  getType () = TYPE\_EVENT | EVT\_SEND\_DATA (SW - '90 00')  third response byte of the R-APDU data is '40' | N1 |
| 2 | **message of type event - Connectivity** | | | | |
| - Send command on ISO interface to select applet; the initial conditions in clause 6.1.3.3.3 not applicable here  - EVT\_STANDBY  - Send on ISO interface (INS = '01') | extends ConnectivityListener  activateEvent()  event = EVENT\_STAND\_BY  onCallback()  event = EVENT\_STAND\_BY  HCIMessage.getType() | No exception shall be thrown.  getType () = TYPE\_EVENT | SW = '90 00' | N1 |

#### 6.1.3.4 Method getInstruction

Test Area Reference: Api\_1\_Hme\_Min.

##### 6.1.3.4.1 Conformance requirements

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

byte getInstruction()

6.1.3.4.1.1 Normal execution

* CRRN1: returns the instruction part of the HCP message header of the current incoming HCP message, see ETSI TS 102 622 [] for further information. It returns the message instruction, b8,b7 of the return value are set to zero.
* CRRN2: in the case of a fragmented incoming message this method shall return the HCI message instruction coded in the first part of the HCI message.

NOTE: Development of test cases for CRRN2 is FFS.

6.1.3.4.1.2 Parameter errors

* None.

6.1.3.4.1.3 Context errors

* None.

##### 6.1.3.4.2 Test Suite Files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hme\_Min\_1.java | 1 |
| Api\_1\_Hme\_Min\_2.java | 2 |

##### 6.1.3.4.3 initial condition

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI Interface.

##### 6.1.3.4.4 Test procedure

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test case** | | | | | | | | |
| **ID** | | **HCI Command** | | **API Description** | | **API Expectation** | **HCI Response** | **CRR** |
| 1 | | **message of type event: Card emulation** | | | | | | |
|  | | EVT\_SEND\_DATA (INS = '01') | | onCallback()  event = EVENT\_ON\_SEND\_DATA  HCIMessage.getInstruction() | No exception shall be thrown.  getInstruction() = '10' | EVT\_SEND\_DATA (SW - '90 00')  the 4th response byte of the R-APDU data is '10' | N1 |
| 2 | | **message of type response: Card emulation** | | | | | | |
|  | | - EVT\_SEND\_DATA (INS = '01')  - ANY\_OK(ATQA)  - EVT\_SEND\_DATA (arbitrary data) | | onCallback()  prepareAndSendGetParameterCommand()  parameter = PARAM\_ID\_TYPE\_A\_CARD\_ATQA  onCallback()  event = EVENT\_GET\_PARAMETER\_RESPONSE  HCIMessage.getInstruction() | No exception shall be thrown.  getInstruction() = '00' | - ANY\_GET\_PARAMTER(ATQA)  - EVT\_SEND\_DATA (SW - '90 00')  - EVT\_SEND\_DATA (SW - '90 00')  the 4th response byte of the R-APDU data is '00' | N1 |

#### 6.1.3.5 Method getReceiveOffset

Test Area Reference: Api\_1\_Hme\_Mro.

##### 6.1.3.5.1 Conformance requirements

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

short getReceiveOffset()

6.1.3.5.1.1 Normal execution

* CRRN1: this method returns the offset of the received HCI message data in the receive buffer, retrieved via getReceiveBuffer().

6.1.3.5.1.2 Parameter errors

* None.

6.1.3.5.1.3 Context error

* None.

NOTE: This method is implicitly tested in clause 6.1.3.7.

#### 6.1.3.6 Method getReceiveLength

Test Area Reference: Api\_1\_Hme\_Mrl.

##### 6.1.3.6.1 Conformance requirements

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

short getReceiveLength()

6.1.3.6.1.1 Normal execution

* CRRN1: this method returns the length of the received HCI message data. It is the size of the message data available in the receive buffer.
* CRRNX: if the message is not complete then the returned value is the actual HCI message fragment length.

NOTE: Development of test cases for CRRNX is FFS.

6.1.3.6.1.2 Parameter errors

* None.

6.1.3.6.1.3 Context error

* None.

##### 6.1.3.6.2 Test Suite Files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hme\_Mrl\_1.java | 1 |

##### 6.1.3.6.3 Initial condition

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI interface.

##### 6.1.3.6.4 Test procedure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | | | | | |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Get received message length: CardEmulationService** | | | | |
| EVT\_SEND\_DATA:  Send message with the length '0A'  message is complete | onCallback()  event = EVENT\_ON\_SEND\_DATA  HCIMessage.getReceiveLength() | No exception shall be thrown.  getReceiveLength() = '0A' | EVT\_SEND\_DATA (SW - '90 00')  Result returned in the 7th and 8th response byte of the R-APDU data ('00 0A') | N1 |

#### 6.1.3.7 Method getReceiveBuffer

Test Area Reference: Api\_1\_Hme\_Mrb.

##### 6.1.3.7.1 Conformance requirements

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

byte[] getReceiveBuffer()

6.1.3.7.1.1 Normal execution

* CRRN1: returns a reference to the underlying HCI message receive buffer. The length of this buffer is system dependent; it is guaranteed to be at least 270 bytes.
* CRRN2: it returns the buffer holding the current HCI message.
* CRRN3: the content of the array starts at the location indicated by getReceiveOffset() and has the length indicated by getReceiveLength().

6.1.3.7.1.2 Parameter errors

* None.

6.1.3.7.1.3 Context errors

* None.

##### 6.1.3.7.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hme\_Mrb\_1.java | 1 |

##### 6.1.3.7.3 Initial condition

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI interface.

##### 6.1.3.7.4 Test procedure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | | | | | |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Message length < 270 bytes** | | | | |
| EVT\_SEND\_DATA (message with the length = 09: '00 01 02 03 04 05 06 07 08') | onCallback()  event = EVENT\_ON\_SEND\_DATA  getReceiveBuffer()  getReceiveOffset()  getReceiveLength() | No exception shall be thrown.  First 9 bytes of getReceiveBuffer() shall be '00 01 02 03 04 05 06 07 08 ' | EVT\_SEND\_DATA ()  First 9 bytes of the response data shall be '00 01 02 03 04 05 06 07 08' | N1, N2,  N3 |

### 6.1.4 Class HCIException

#### 6.1.4.1 Method throwIt

Test Area Reference: Api\_1\_Hxp\_Trw.

##### 6.1.4.1.1 Conformance requirements

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

public static void throwIt(short reason)

throws HCIException

6.1.4.1.1.1 Normal execution

* CRRN1: Throws the JCRE owned instance of the HCIException with the specified reason code.

6.1.4.1.1.2 Parameter errors

* None.

6.1.4.1.1.3 Context errors

* None.

##### 6.1.4.1.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hxp\_Trw\_1.java | 1 |

##### 6.1.4.1.3 Initial conditions

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI Interface.

##### 6.1.4.1.4 Test procedure

| Test Case | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| ID | HCI Command | API Description | API Expectation | | HCI Response | CRR |
| 1 | Throw Exception | | | | | |
|  | 1 - EVT\_SEND\_DATA (INS = '01') | throwIt()  reason = HCI\_ACCESS\_NOT\_GRANTED | | Throw exception with corresponding reason code | EVT\_SEND\_DATA (SW - '90 00') | N1 |
|  | 2 - EVT\_SEND\_DATA (INS = '02') | throwIt()  reason = HCI\_CONDITIONS\_NOT\_SATISFIED | | Throw exception with corresponding reason code | EVT\_SEND\_DATA (SW - '90 00') | N1 |
|  | 3 - EVT\_SEND\_DATA (INS = '03') | throwIt()  reason = HCI\_CURRENTLY\_DISABLED | | Throw exception with corresponding reason code | EVT\_SEND\_DATA (SW - '90 00') | N1 |
|  | 4 - EVT\_SEND\_DATA (INS = '04') | throwIt()  reason = HCI\_FRAGMENTED\_MESSAGE\_ONGOING | | Throw exception with corresponding reason code | EVT\_SEND\_DATA (SW - '90 00') | N1 |
|  | 5 - EVT\_SEND\_DATA (INS = '05') | throwIt()  reason = HCI\_INVALID\_LENGTH | | Throw exception with corresponding reason code | EVT\_SEND\_DATA (SW - '90 00') | N1 |
|  | 6 - EVT\_SEND\_DATA (INS = '06') | throwIt()  reason = HCI\_LISTENER\_ALREADY\_REGISTERED | | Throw exception with corresponding reason code | EVT\_SEND\_DATA (SW - '90 00') | N1 |
|  | 7 - EVT\_SEND\_DATA (INS = '07') | throwIt()  reason = HCI\_NOT\_AVAILABLE | | Throw exception with corresponding reason code | EVT\_SEND\_DATA (SW - '90 00') | N1 |
|  | 8 - EVT\_SEND\_DATA (INS = '08') | throwIt()  reason = HCI\_RESOURCES\_NOT\_AVAILABLE | | Throw exception with corresponding reason code | EVT\_SEND\_DATA (SW - '90 00') | N1 |
|  | 9 - EVT\_SEND\_DATA (INS = '09') | throwIt()  reason = HCI\_SERVICE\_NOT\_AVAILABLE | | Throw exception with corresponding reason code | EVT\_SEND\_DATA (SW - '90 00') | N1 |
|  | 10 - EVT\_SEND\_DATA (INS = '0A') | throwIt()  reason = HCI\_WRONG\_EVENT\_TYPE | | Throw exception with corresponding reason code | EVT\_SEND\_DATA (SW - '90 00') | N1 |
|  | 11 - EVT\_SEND\_DATA (INS = '0B') | throwIt() reason = HCI\_WRONG\_LISTENER\_TYPE | | Throw exception with corresponding reason code | EVT\_SEND\_DATA (SW - '90 00') | N1 |

### 6.1.5 Interface HCIListener

#### 6.1.5.1 Method onCallback

Test Area Reference: Api\_1\_Hln\_Ocb.

##### 6.1.5.1.1 Conformance requirements

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

void onCallback(byte event, HCIMessage message)

6.1.5.1.1.1 Normal execution

* CRRN1: The Contactless Framework shall invoke the method \*typeListener.onCallback only with events which are defined in this particular listener or HCIListener and only with message objects of this listener type.
* CRRN2: The Applet instance shall activate the events with CardEmulationService.activateEvent before it will receive any event notification.
* CRRN3: The context as defined in the Java Card™ specification [], [] and [] shall be set to the context of the Applet which implements the onCallback() method. The previous context (context of the caller) shall be the context of the Contactless Framework.
* CRRN4: Upon return from the onCallback() method a pending transaction shall be aborted.
* CRRN5: Void
* CRRN6: The Contactless Framework shall be able to receive one or more HCI messages while waiting for a response related to a command originated by the Applet (e.g. processing a request for parameters) especially for the EVT\_FIELD\_OFF case.
* CRRN7: All other HCI messages [except EVT\_FIELD\_OFF] shall be delivered to the Applet instance in the same order as they were received by the Contactless Framework.

NOTE: Development of test cases for CRRN7 is FFS.

6.1.5.1.1.2 Parameter errors

* None.

6.1.5.1.1.3 Context errors

* None.

##### 6.1.5.1.2 Test Suite Files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_1\_Hln\_Ocb\_4.java | 1 |
| Api\_1\_Hln\_Ocb\_5.java | 2 |
| Api\_1\_Hln\_Ocb\_2.java | 4 |
| Api\_1\_Hln\_Ocb\_3.java | 5 |

##### 6.1.5.1.3 Initial conditions

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.

##### 6.1.5.1.4 Test procedure

| **Test case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Get context** | | | | |
| 1 - Send EVT\_SEND\_DATA (Select applet)  Send  EVT\_SEND\_DATA (INS = '01')  ) | Service = CardEmulationService  Listener = CardEmulationListener  process():  JCSystem.getAID(); | No exception shall be thrown  expected AID = current AID | EVT\_SEND\_DATA (SW '90 00') | N3 |
| 2 - Send EVT\_SEND\_DATA (Select applet)  Send  EVT\_SEND\_DATA (INS = '02') | Service = CardEmulationService  Listener = CardEmulationListener  process():  JCSystem.getPreviousContextAID(); | No exception shall be thrown  previous AID = null | EVT\_SEND\_DATA (SW '90 00') | N3 |
| 2 | **Abort transaction** | | | | |
| - Send EVT\_SEND\_DATA (Select applet)  - Send  EVT\_SEND\_DATA (INS = '01')  - Send ENVELOPE command on the ISO interface which will trigger EVENT\_UNRECOGNIZED\_ENVELOPE in the applet as specified in ETSI TS 102 241 [] | Service = CardEmulationService  Listener = CardEmulationListener  onCallback():  JCsystem.beginTransaction();  Set up test array  //no JCSystem.commitTransaction!!  processToolkit():  check array modification | No exception shall be thrown  Test array shall not be modifed | EVT\_SEND\_DATA (SW '90 00')  EVT\_SEND\_DATA (SW '90 00')  SW - '90 00' | N4 |
| 3 | **Void** | | | | |
|  |  |  |  |  |
| 4 | **Receive messages while waiting for get parameter** | | | | |
| - EVT\_SEND\_DATA (Select applet)  - EVT\_FIELD\_OFF before ANY\_OK | Listener = CardEmulationListener  activateEvent()  event = EVENT\_GET\_PARAMETER\_RESPONSE  prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_B\_CARD\_ATQB | No exception shall be thrown  onCallback() shall receive ANY\_OK as EVENT\_GET\_PARAMETER\_RESPONSE | ANY\_GET\_PARAMETER(ATQB)  EVT\_SEND\_DATA ()  No error after EVT\_FIELD\_OFF | N1, N2, N6 |
| 5 | **Receive messages while waiting for get parameter** | | | | |
| - EVT\_SEND\_DATA (Select applet)  - EVT\_FIELD\_OFF  - ANY\_OK | Listener = CardEmulationListener  activateEvent()  event = EVENT\_GET\_PARAMETER\_RESPONSE  prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_A\_CARD\_ATQA | No exception shall be thrown  onCallback() shall receive ANY\_OK as EVENT\_GET\_PARAMETER\_RESPONSE | ANY\_GET\_PARAMETER(ATQA)  EVT\_SEND\_DATA ()  No error after EVT\_FIELD\_OFF | N1, N2, N6 |

## 6.2 Package uicc.hci.services

### 6.2.1 Package CardEmulation Service

#### 6.2.1.1 Interface CardEmulationMessage

##### 6.2.1.1.1 Method prepareAndSendGetParameterCommand

Test Area Reference: Api\_2\_CEm\_Sgp.

6.2.1.1.1.1 Conformance requirements

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

void prepareAndSendGetParameterCommand(byte paramID)

throws HCIException

6.2.1.1.1.1.1 Normal execution

* CRRN1: this non-blocking method builds the HCI command ANY\_GET\_PARAMETER for the indicated information and sends it to the appropriate Card RF Gate of the CLF.
* CRRN2: the paramID indicates a constant parameter of the requested value.
* CRRN3: The CardEmulationListener.onCallback method shall be called by the Contactless Framework. The HCI message that resulted in the selection of this Applet according to the rules defined in "GlobalPlatform Amendment C" [] shall be provided by the CardEmulationMessage.

6.2.1.1.1.1.2 Parameter errors

* CRRP1: throws HCIException with the reason code HCI\_CURRENTLY\_DISABLED if the HCI interface was disabled.

NOTE: Development of test cases for CRRP1 is FFS.

6.2.1.1.1.1.3 Context errors

* CRRC1: throws HCIException with the reason code HCI\_FRAGMENTED\_MESSAGE\_ONGOING if the Contactless Framework is still receiving a fragmented HCI Message.
* CRRC2: throws HCIException with the reason code HCI\_RESOURCES\_NOT\_AVAILABLE if the contactless framework does not have enough resources to process the command.

NOTE: CRRC2 is not tested, as it is not possible to force the situation where the contactless framework does not have enough resources to process the command.

6.2.1.1.1.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_2\_CEm\_Sgp\_1.java | 1-1 |
| Api\_2\_CEm\_Sgp\_2.java | 1-2 |
| Api\_2\_CEm\_Sgp\_3.java | 1-3 |
| Api\_2\_CEm\_Sgp\_4.java | 1-4 |
| Api\_2\_CEm\_Sgp\_5.java | 1-5 |
| Api\_2\_CEm\_Sgp\_6.java | 1-6 |
| Api\_2\_CEm\_Sgp\_7.java | 1-7 |
| Api\_2\_CEm\_Sgp\_8.java | 1-8 |
| Api\_2\_CEm\_Sgp\_9.java | 2-1 |
| Api\_2\_CEm\_Sgp\_10.java | 2-2 |
| Api\_2\_CEm\_Sgp\_11.java | 2-3 |
| Api\_2\_CEm\_Sgp\_12.java | 2-4 |
| Api\_2\_CEm\_Sgp\_13.java | 2-5 |
| Api\_2\_CEm\_Sgp\_14.java | 3-1 |
| Api\_2\_CEm\_Sgp\_15.java | 3-2 |

6.2.1.1.1.3 Initial condition

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.

6.2.1.1.1.4 Test procedure

| **Test case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Get parameter: Type A** | | | | |
|  | 1 - EVT\_SEND\_DATA (Sele | prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_A\_CARD\_APPLICATION\_DATA | No exception shall be thrown | GET\_PARAMETER with command parameter as indicated in the API Description column | N1,  N2, N3 |
|  | 2 - EVT\_SEND\_DATA (Select applet) | prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_A\_CARD\_ATQA | No exception shall be thrown | GET\_PARAMETER with command parameter as indicated in the API Description column | N1,  N2, N3 |
|  | 3 - EVT\_SEND\_DATA (Select applet) | prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_A\_CARD\_CID\_SUPPORT | No exception shall be thrown | GET\_PARAMETER with command parameter as indicated in the API Description column | N1,  N2, N3 |
|  | 4 - EVT\_SEND\_DATA (Select applet) | prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_A\_CARD\_CLT\_SUPPORT | No exception shall be thrown | GET\_PARAMETER with command parameter as indicated in the API Description column | N1,  N2, N3 |
|  | 5 - EVT\_SEND\_DATA (Select applet) | prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_A\_CARD\_DATARATE\_MAX | No exception shall be thrown | GET\_PARAMETER with command parameter as indicated in the API Description column | N1,  N2, N3 |
|  | 6 - EVT\_SEND\_DATA (Select applet) | prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_A\_CARD\_FWI\_SFGI | No exception shall be thrown | GET\_PARAMETER with command parameter as indicated in the API Description column | N1,  N2, N3 |
|  | 7 -EVT\_SEND\_DATA (Select applet) | prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_A\_CARD\_MODE | No exception shall be thrown | GET\_PARAMETER with command parameter as indicated in the API Description column | N1,  N2, N3 |
|  | 8 -EVT\_SEND\_DATA (Select applet) | prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_A\_CARD\_SAK | No exception shall be thrown | GET\_PARAMETER with command parameter as indicated in the API Description column | N1,  N2, N3 |
| 2 | **Get parameter: Type B** | | | | |
|  | 1- EVT\_SEND\_DATA (Select applet) | prepareAndSendGetParameterCommand()  paramID =  PARAM\_ID\_TYPE\_B\_CARD\_AFI | No exception shall be thrown | GET\_PARAMETER with command parameter as indicated in the API Description column | N1,  N2, N3 |
|  | 2 - EVT\_SEND\_DATA (Select applet) | prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_B\_CARD\_ATQB | No exception shall be thrown | GET\_PARAMETER with command parameter as indicated in the API Description column | N1,  N2, N3 |
|  | 3 - EVT\_SEND\_DATA (Select applet) | prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_B\_CARD\_DATARATE\_MAX | No exception shall be thrown | GET\_PARAMETER with command parameter as indicated in the API Description column | N1,  N2, N3 |
|  | 4 - EVT\_SEND\_DATA (Select applet) | prepareAndSendGetParameterCommand()  paramID =  PARAM\_ID\_TYPE\_B\_CARD\_HIGHER\_LAYER\_RESPONSE | No exception shall be thrown | GET\_PARAMETER with command parameter as indicated in the API Description column | N1,  N2, N3 |
|  | 5 - EVT\_SEND\_DATA (Select applet) | prepareAndSendGetParameterCommand()  paramID =  PARAM\_ID\_TYPE\_B\_CARD\_MODE | No exception shall be thrown | GET\_PARAMETER with command parameter as indicated in the API Description column | N1,  N2, N3 |
| 3 | **The Contactless Framework is receiving fragmented HCI Message** | | | | |
|  | 1 –  EVT\_SEND\_DATA (Select applet)  Send as fragmented HCI Message:  - EVT\_SEND\_DATA (data length = supported buffer size + 5 bytes, message not complete)  With CB = 1 for the last sent frame  EVT\_SEND\_DATA sent without waiting for response | prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_A\_CARD\_ATQA | Shall throw uicc.hci.framework.HCIException with error code HCI\_FRAGMENTED\_MESSAGE\_ONGOING | - EVT\_SEND\_DATA (SW – ’90 00’)  - EVT\_SEND\_DATA (No data)  - EVT\_SEND\_DATA (one byte with second least significant bit set) | C1 |
|  | 2 EVT\_SEND\_DATA (Select applet)  Send as fragmented HCI Message:  EVT\_SEND\_DATA (data length = supported buffer size + 5 bytes, message not complete)  EVT\_SEND\_DATA sent without waiting for response | prepareAndSendGetParameterCommand()  paramID =  PARAM\_ID\_TYPE\_B\_CARD\_ATQB | Shall throw uicc.hci.framework.HCIException with error code HCI\_FRAGMENTED\_MESSAGE\_ONGOING | - EVT\_SEND\_DATA (SW – ’90 00’)  - EVT\_SEND\_DATA (No data)  - EVT\_SEND\_DATA (one byte with second least significant bit set) | C1 |

##### 6.2.1.1.2 Method prepareAndSendSendDataEvent

Test Area Reference: Api\_2\_CEm\_Ssd.

6.2.1.1.2.1 Conformance requirements

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

void prepareAndSendSendDataEvent(byte[] data,

short offset,

short len)

throws HCIException,

java.lang.NullPointerException,

java.lang.ArrayIndexOutOfBoundsException

6.2.1.1.2.1.1 Normal execution

* CRRN1: This non-blocking method builds the the HCI event EVT\_SEND\_DATA sends it to the Contactless Card Emulation Gate of the CLF.
* CRRN2: the data to be sent should be formatted according to the HCI specification, the offset define the offset of data into the data buffer and the len define the length of data in the data buffer.
* CRRN3: Applet instances shall receive CardEmulationMessages after the registration of a CardEmulationListener interface to a CardEmulationService only if the EVENT\_ON\_SEND\_DATA is activated for the Applet instance.
* CRRN4: The CardEmulationListener.onCallback method shall be called by the Contactless Framework. The HCI message that resulted in the selection of this Applet according to the rules defined in "GlobalPlatform Amendment C" [] shall be provided by the CardEmulationMessage.

6.2.1.1.2.1.2 Parameter errors

* CRRP1: throws java.lang.NullPointerException - if data is null.
* CRRP2: throws java.lang.ArrayIndexOutOfBoundsException - if operation would cause access of data outside array bounds.

NOTE: Development of test cases for CRRP2 is FFS.

6.2.1.1.2.1.3 Context errors

* CRRC1: throws HCIException with the reason code HCI\_CURRENTLY\_DISABLED if the HCI interface was disabled.
* CRRC2: throws HCIException with the reason code HCI\_FRAGMENTED\_MESSAGE\_ONGOING if the Contactless Framework is still receiving a fragmented HCI Message.
* CRRC3: throws HCIException with the reason code HCI\_RESOURCES\_NOT\_AVAILABLE if the contactless framework does not have enough resources to process the command.

NOTE 1: Development of test cases for CRRC1 is FFS.

NOTE 2: CRRC3 is not tested, as it is not possible to force the situation where the contactless framework does not have enough resources to process the command.

6.2.1.1.2.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_2\_CEm\_Ssd\_1.java | 1 |
| Api\_2\_CEm\_Ssd\_1.java | 2 |
| Api\_2\_CEm\_Ssd\_1.java | 3 |
| Api\_2\_CEm\_Ssd\_4.java | 4 |
| Api\_2\_CEm\_Ssd\_2.java | 5-1 |
| Api\_2\_CEm\_Ssd\_3.java | 5-2 |

##### 6.2.1.1.2.3 Initial conditions

All test cases shall be executed in both full power mode and low power mode.

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.

6.2.1.1.2.4 Test procedure

| **Test case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI Commnad** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Send data - Type A** | | | | |
| EVT\_SEND\_DATA (Select applet) | prepareAndSendSendDataEvent()  data = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09}  offset = 2  length = 5 | No exception shall be thrown | EVT\_SEND\_DATA ({0x03, 0x04, 0x05, 0x06, 0x07}) | N1,  N2, N3, N4 |
| 2 | **Send data - Type B** | | | | |
| EVT\_SEND\_DATA (Select applet) | prepareAndSendSendDataEvent()  data = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09}  offset = 2  length = 5 | No exception shall be thrown | EVT\_SEND\_DATA ({0x03, 0x04, 0x05, 0x06, 0x07}) | N1,  N2, N3, N4 |
| 3 | **Recption of fragmented HCI Message** | | | | |
| - EVT\_SEND\_DATA to select applet  Send as fragmented HCI Message:  - EVT\_SEND\_DATA (data length = supported buffer size + 5 bytes, message not complete)  - EVT\_SEND\_DATA (INS = '01') without waiting for response  - Send EVT\_SEND\_DATA (INS = '01') | prepareAndSendSendDataEvent()  data = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09}  offset = 2  length = 5 | Shall throw uicc.hci.framework.HCIException with error code HCI\_FRAGMENTED\_MESSAGE\_ONGOING | - EVT\_SEND\_DATA (SW - '90 00')  - EVT\_SEND\_DATA (No data)  - EVT\_SEND\_DATA (SW- '90 02') | C2 |
| 4 | **Send no data** | | | | |
| EVT\_SEND\_DATA (Select applet) | prepareAndSendSendDataEvent()  data = null  offset = 2  length = 5 | Shall throw java.lang exception with error code NullPointerException | EVT\_SEND\_DATA ({0xFF, 0xFF, 0xFF, 0xFF, 0xFF }) | P1 |
| 5 | **Array Index Out Of Bounds** | | | | |
| 1 -EVT\_SEND\_DATA (Select applet) | prepareAndSendSendDataEvent()  1 - data = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09}  offset = 2  length = 15 | Shall throw java.lang exception with error code ArrayIndexOutOfBoundsException | EVT\_SEND\_DATA ({0xFF, 0xFF, 0xFF, 0xFF, 0xFF }) | P2 |
| 2 - EVT\_SEND\_DATA (Select applet) | prepareAndSendSendDataEvent()  data = {0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09}  offset = 15  length = 5 | Shall throw java.lang exception with error code ArrayIndexOutOfBoundsException | EVT\_SEND\_DATA ({0xFF, 0xFF, 0xFF, 0xFF, 0xFF }) | P2 |

##### 6.2.1.1.3 Method selectingMessage

Test Area Reference: Api\_2\_CEm\_Scm

6.2.1.1.3.1 Conformance requirements

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

boolean selectingMessage()

6.2.1.1.3.1.1 Normal execution

* CRRN1: This method is used by the applet to distinguish the HCI Message command which selected this applet from all other HCI messages.
* CRRN2: returns true if the current HCI message caused the selection of the Applet according to the rules in GlobalPlatform Amendment C [].

6.2.1.1.3.1.2 Parameter errors

* None.

6.2.1.1.3.1.3 Context errors

* None.

6.2.1.1.3.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_2\_CEm\_Scm\_1.java | 1 |
| Api\_2\_CEm\_Scm\_1.java | 2 |

##### 6.2.1.1.3.3 Initial conditions

All test cases shall be executed in both full power mode and low power mode.

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.

6.2.1.1.3.4 Test procedure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | | | | | |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Select Applet** | | | | |
| EVT\_SEND\_DATA (Select applet) | selectingMessage() | No exception shall be thrown.  selectingMessage()= true | EVT\_SEND\_DATA (SW - '90 00') | N1,  N2 |
| 2 | **Arbitrary Command** | | | | |
| - EVT\_SEND\_DATA (Select applet)  - EVT\_SEND\_DATA (INS = '01') | selectingMessage() | No exception shall be thrown.  selectingMessage() = false | - EVT\_SEND\_DATA(SW - '90 00')  - EVT\_SEND\_DATA ('01 02 03 90 00') | N1,  N2 |

#### 6.2.1.2 Interface CardEmulationService

##### 6.2.1.2.1 Method getCardRFType

Test Area Reference: Api\_2\_CEs\_RFt.

6.2.1.2.1.1 Conformance requirements

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

byte getCardRFType()

6.2.1.2.2.1.1 Normal execution

* CRRN1: returns the reader RF type on which the service is connected, shall be one of the TYPE\_\* constant values defined in this interface, or -1 if the RF type can not be evaluated.
* CRRN2: Applets communicating through the process() method shall also be able to use the API services defined in the present document which do not require a CardEmulationListener registration (e.g. requesting the power mode or connectivity service).

6.2.1.2.2.1.2 Parameter errors

* None.

6.2.1.2.2.1.3 Context errors

* None.

##### 6.2.1.2.2 Test Suite Files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_2\_CEs\_RFt\_1.java | 1 |
| Api\_2\_CEs\_RFt\_1.java | 2 |
| Api\_2\_CEs\_RFt\_1.java | 3 |
| Api\_2\_CEs\_RFt\_1.java | 4 |

##### 6.2.1.2.3 Initial conditions

All test cases shall be executed in both full power mode and low power mode.

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI interface.

6.2.1.2.3.1 Test procedure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test case** | | | | | |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **RF type A** | | | | |
| EVT\_SEND\_DATA  (INS = '01') | getCardRFType() | No exception shall be thrown  getCardRFType() = TYPE\_A\_CARD\_RF | EVT\_SEND\_DATA  (SW = '90 01') | N1  N2 |
| 2 | **RF type B** | | | | |
| EVT\_SEND\_DATA  (INS = '01') | getCardRFType() | No exception shall be thrown  getCardRFType() = TYPE\_B\_CARD\_RF | EVT\_SEND\_DATA  (SW = '90 02') | N1  N2 |
| 3 | **RF type F** | | | | |
| EVT\_SEND\_DATA  (INS = '01') | getCardRFType() | No exception shall be thrown  getCardRFType() = TYPE\_F\_CARD\_RF | EVT\_SEND\_DATA  (SW = '90 04') | N1  N2 |
| 4 | **RF type B'** | | | | |
| EVT\_SEND\_DATA  (INS = '01') | getCardRFType() | No exception shall be thrown  getCardRFType() = TYPE\_B\_PRIM\_CARD\_RF | EVT\_SEND\_DATA  (SW = '90 03') | N1  N2 |

#### 6.2.1.3 Interface CardEmulationListener

##### 6.2.1.3.1 Method onCallback

Test Area Reference: Api\_2\_CEl\_Ocb.

6.2.1.3.1.1 Conformance requirements

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

void onCallback(byte event, HCIMessage message)

6.2.1.3.1.1.1 Normal execution

* CRRN1: The Contactless Framework shall invoke the method CardEmulationListener.onCallback only with events which are defined in this particular listener or CardEmulationListener and only with CardEmulationMessage objects.
* CRRN2: The Applet instance shall activate the events with CardEmulationService.activateEvent before it will receive any event notification. In case the Applet instance has registered the *CardEmulationListener* and has activated the EVENT\_ON\_SEND\_DATA the *process()* method of this Applet instance shall not be invoked during the selection. The *CardEmulationListener.onCallback* method shall be called by the Contactless Framework.
* CRRN3: If the EVENT\_ON\_SEND\_DATA is deactivated for the Applet instance and an APDU is received via the EVT\_SEND\_DATA, the javacard.framework.APDU class and the process() method of the Applet instance shall be invoked.
* CRRN4: It shall not be possible to switch between the usage of the CardEmulationListener interface and the invocation through the process() method within a contactless application session, i.e. not before the Applet has been deselected and selected again.

6.2.1.3.1.1.2 Parameter errors

* None.

6.2.1.3.1.1.3 Context errors

* None.

6.2.1.3.1.2 Test Suite Files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_2\_CEl\_Ocb\_1.java | 1 |
| Api\_2\_CEl\_Ocb\_4.java | 3 |
| Api\_2\_CEl\_Ocb\_1.java | 4 |
| Api\_2\_CEl\_Ocb\_3.java | 6 |
| Api\_2\_CEl\_Ocb\_6.java | 7-2 |
| Api\_2\_CEl\_Ocb\_7.java | 7-3 |
| Api\_2\_CEl\_Ocb\_8.java | 7-4 |

6.2.1.3.1.3 Initial conditions

All test cases shall be executed in both full power mode and low power mode.

* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI interface.

6.2.1.3.1.4 Test procedure

| **Test case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Activate EVT\_SEND\_DATA - Type A** | | | | |
| - EVT\_SEND\_DATA(INS = '14')  - EVT\_SEND\_DATA(INS = '24')-  - deselect the applet  - select the applet | HCIListener = CardEmulationListener  HCIService = CardEmulationService  activateEvent()  event =  CardEmulationListener.EVENT\_ON\_SEND\_DATA | - process() method: No exception shall be thrown  - process() method: No exception shall be thrown  onCallback() method shall notice the event EVENT\_ON\_SEND\_DATA | - EVT\_SEND\_DATA (SW ‑ '90 00')  - EVT\_SEND\_DATA (SW ‑ '90 00')  - EVT\_SEND\_DATA (SW ‑ '90 00') | N1,  N2,  N3,  N4, |
| 2 | Void | | | | |
| 3 | **Activate EVENT\_GET\_PARAMETER\_RESPONSE - Type A** | | | | |
| -EVT\_SEND\_DATA (INS = '01')  - ANY\_OK as response to GET\_PARAMETER  - deselect/select the applet  - EVT\_SEND\_DATA(INS='22') | Listener = CardEmulationListener  activateEvent()  event = EVENT\_GET\_PARAMETER\_RESPONSE  prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_A\_CARD\_ATQA | onCallback() method shall notice the event EVENT\_ON\_SEND\_DATA  onCallback() shall receive ANY\_OK as EVENT\_GET\_PARAMETER\_RESPONSE  -process() method  No exception shall be thrown  - process() method: ANY\_OK reception is verified | GET\_PARAMETER with the PARAM\_ID\_TYPE\_A\_CARD\_ATQA  - No exception after ANY\_OK reception.  EVT\_SEND\_DATA (SW ‑ '90 00')  - EVT\_SEND\_DATA (SW ‑ '90 00')  - EVT\_SEND\_DATA (SW ‑ '90 00') | N1,  N2,  N3,  N4, |
| 4 | **Activate EVT\_SEND\_DATA - Type B** | | | | |
| - EVT\_SEND\_DATA(INS = '14')  - EVT\_SEND\_DATA(INS = '24')-  - deselect the applet  - select the applet | HCIListener = CardEmulationListener  HCIService = CardEmulationService  activateEvent()  event =  CardEmulationListener.EVENT\_ON\_SEND\_DATA | - process() method: No exception shall be thrown  - process() method: No exception shall be thrown  onCallback() method shall notice the event EVENT\_ON\_SEND\_DATA | - EVT\_SEND\_DATA (SW ‑ '90 00')  - EVT\_SEND\_DATA (SW ‑ '90 00')  - EVT\_SEND\_DATA (SW ‑ '90 00') | N1,  N2,  N3,  N4, |
| 5 |  | | | | |
|  |  |  |  |  |
| 6 | **Activate EVENT\_GET\_PARAMETER\_RESPONSE - Type B** | | | | |
| - EVT\_SEND\_DATA (INS = '01')  - ANY\_OK as response to GET\_PARAMETER  - deselect/select the applet  - EVT\_SEND\_DATA(INS = '22') | Listener = CardEmulationListener  activateEvent()  event = EVENT\_GET\_PARAMETER\_RESPONSE  prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_B\_CARD\_ATQB | onCallback() method shall notice the event EVENT\_ON\_SEND\_DATA  onCallback() shall receive ANY\_OK as EVENT\_GET\_PARAMETER\_RESPONSE  -process() method  No exception shall be thrown  - process() method: ANY\_OK reception is verified | - GET\_PARAMETER with the PARAM\_ID\_TYPE\_B\_CARD\_ATQB  - No exception after ANY\_OK reception.  EVT\_SEND\_DATA (SW ‑ '90 00')  - EVT\_SEND\_DATA (SW ‑ '90 00')  - EVT\_SEND\_DATA (SW ‑ '90 00') | N1,  N2,  N3,  N4 |
| 7 | **Deactivate events** | | | | |
|  | 1 - Void |  |  |  |  |
|  | 2 - EVT\_SEND\_DATA(INS='12')  - ANY\_OK as response to GET\_PARAMETER  - deselect/select the applet  - EVT\_SEND\_DATA(INS='20') | HCIService = CardEmulationService  Event has been successfully activated  deactivateEvent()  event =  CardEmulationListener.EVENT\_GET\_PARAMETER\_RESPONSE  prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_A\_CARD\_ATQA | No exception shall be thrown  onCallback() method shall not notice the event | - ignore the first response  - EVT\_SEND\_DATA (SW ‑ '90 00')  - EVT\_SEND\_DATA (SW ‑ '90 00') | N2 |
|  | 3 - EVT\_SEND\_DATA(INS='12')  - ANY\_OK as response to GET\_PARAMETER  - deselect/select the applet  - EVT\_SEND\_DATA(INS='20') | HCIService = CardEmulationService  Event has been successfully activated  deactivateEvent()  event =  CardEmulationListener.EVENT\_GET\_PARAMETER\_RESPONSE  prepareAndSendGetParameterCommand()  paramID = PARAM\_ID\_TYPE\_B\_CARD\_ATQB | No exception shall be thrown  onCallback() method shall not notice the event | - ignore the first response  - EVT\_SEND\_DATA (SW ‑ '90 00')  - EVT\_SEND\_DATA (SW ‑ '90 00') | N2 |
|  | 4 -  -EVT\_SEND\_DATA(INS='14')  - deselect  /select the applet  - EVT\_SEND\_DATA(INS='20') | HCIService = CardEmulationService  Event has been successfully activated  deactivateEvent()  event =  CardEmulationListener.EVENT\_ON\_SEND\_DATA | No exception shall be thrown  onCallback() method shall not notice the event | - EVT\_SEND\_DATA (SW ‑ '90 00')  - EVT\_SEND\_DATA (SW ‑ '90 00')  - EVT\_SEND\_DATA (SW- '90 01') | N2 |

### 6.2.2 Package Connectivity Service

#### 6.2.2.1 Interface ConnectivityService

##### 6.2.2.1.1 Method prepareAndSendConnectivityEvent

Test Area Reference: Api\_2\_CNs\_Sce.

6.2.2.1.1.1 Conformance requirements

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

void prepareAndSendConnectivityEvent()

throws HCIException

6.2.2.1.1.1.1 Normal execution

* CRRN1: This non-blocking method builds the HCI event EVT\_CONNECTIVITY which notifies the terminal host that it shall send a "HCI connectivity event" as defined in ETSI TS 102 223 [].
* CRRN2: If the Applet wants to use proactive functionality it shall use the Connectivity Service defined above to send an HCI event EVT\_CONNECTIVITY to the terminal, register for EVENT\_EVENT\_DOWNLOAD\_HCI\_CONNECTIVITY and return. All the proactive functionality of the UICC API defined in ETSI TS 102 241 [] is then available to the Applet when that Applet instance is triggered with the processToolkit() method defined in ETSI TS 102 241 [].
* CRRN3: the contactless runtime environment shall bind the services defined in uicc.hci.services.connectivity to the corresponding resources (e.g. gates and pipes) specified by the HCI protocol [] for the connectivity service.
* CRRN4: The Contactless Framework shall only send the HCI event EVT\_CONNECTIVITY or EVT\_TRANSACTION specified by the HCI protocol [] to an Applet instance, when it is the selected Applet in card emulation mode or when this Applet instance is in the state ACTIVATED (according to "GlobalPlatform Amendment C" []) for the reader mode.
* CRRN5: The ProactiveHandler defined in ETSI TS 102 241 [] shall not be available when the contactless Applet is invoked with the callback methods defined in the present document, or when the Applet is invoked with the process() method of the Applet class defined in Application Programming Interface, Java Card™ Platform [] (in card emulation mode).

6.2.2.1.1.1.2 Parameter errors

* None.

6.2.2.1.1.1.3 Context errors

* CRRC1: throw HCIException with error code reason HCI\_CURRENTLY\_DISABLED if the HCI interface was disabled.
* CRRC2: throw HCIException with error code reason HCI\_RESOURCES\_NOT\_AVAILABLE if the contactless framework does not have enough resources to process the command.
* CRRC3: throw HCIException with error code reason HCI\_CONDITIONS\_NOT\_SATISFIED if the conditions to call this method are not satisfied.

NOTE: CRRC2 is not tested, as it is not possible to force the situation where the contactless framework does not have enough resources to process the command.

6.2.2.1.1.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_2\_CNs\_Sce\_1.java | 1 |
| Api\_2\_CNs\_Sce\_1.java | 2-1 |
| Api\_2\_CNs\_Sce\_6.java | 2-2 |
| Api\_2\_CNs\_Sce\_2.java | 3 |
| Api\_2\_CNs\_Sce\_3.java | 4 |
| Api\_2\_CNs\_Sce\_4.java | 5 |
| Api\_2\_CNs\_Sce\_5.java | 6 |
| Api\_2\_CNs\_Sce\_7.java | 7 |

6.2.2.1.1.3 Initial condition

Initial conditions for ID 1, ID 3, ID 4, ID 5, ID 6:

* Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.
* In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.
* In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.
* UICC has created and opened a pipe to connectivity gate in the terminal host
* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI interface.

Initial conditions for ID 2-1:

* Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.
* In terminal profile set 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.
* In terminal profile the following CAT facilities shall be supported:
* Set the 25th byte, 'b6' to indicate that class m is supported
* Set the 26th byte, 'b2' and the 31st byte, 'b1' to indicate that class r is supported.
* Contactless functionality state is disabled in the UICC as defined in ETSI TS 102 223 [].

Initial conditions for ID 2-2:

* Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.
* In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.
* In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

Initial conditions for ID 7:

* Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.
* In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.
* In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.
* UICC has created and opened a pipe to connectivity gate in the terminal host
* According applet has been successfully installed and selected using ISO Interface and the applet is set into ACTIVATED state.

6.2.2.1.1.4 Test procedure

| **Test case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI commands** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | Send HCI Connectivity Event (card emulation) | | | | |
| - Send EVT\_SEND\_DATA (INS = '01') | Listener = CardEmulationListener  process():  prepareAndSendConnectivityEvent() | No exception shall be thrown | EVT\_SEND\_DATA (SW – '90 00')  HCI event EVT\_CONNECTIVITY sent. This may be sent before or after the EVT\_SEND\_DATA indicated above. | N1  N3 |
| 2 | HCI interface disabled | | | | |
| 1  Send on ISO interface:  - Select applet  - Send APDU (INS = '02') | Listener = CardEmulationListener  process():  prepareAndSendConnectivityEvent() | throw HCIException with error code reason HCI\_CURRENTLY\_DISABLED | SW - '90 00'  No EVT\_ CONNECTIVITY shall be sent | C1  N3 |
| 2 - The contactless interface is disabled in the UICC as defined in Global Platform Amendment C Send on ISO interface send the following commands:  - Send APDU to select the applet.  - Send APDU (INS = '02')  - Postcondition:  The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C | setCommunicationInterface() API method of Global Platform Amendment C [] is used to disable HCI interface in the CRS  Listener = CardEmulationListener  process():  prepareAndSendConnectivityEvent()  setCommunicationInterface() API method of  Global Platform Amendment C [] is used to enable again HCI interface | throw HCIException with error code reason HCI\_CURRENTLY\_DISABLED | SW - '90 00'  No EVT\_ CONNECTIVITY shall be sent | C1  N3 |
| 3 | Wrong Precondition Connectivity Event | | | | |
| The initial conditions in clause 6.2.2.1.1.3 related to card emulation are not applicable here.  Send on ISO:  - Select applet  - Send  APDU with (INS = '01') | Service = ConnectivityService  Listener = ConnectivityListener  process()  ConnectivityService. prepareAndSendConnectivityEvent() | Shall throw exception  with error code reason HCI\_CONDITIONS\_NOT\_SATISFIED | No EVT\_ CONNECTIVITY shall be sent | N4,  C3 |
| 4 | No Proactive Handler | | | | |
| - Send  EVT\_SEND\_DATA (INS = '01') | Service = CardEmulationService  Listener = CardEmulationListener  process()  ProactiveHandlerSystem.getTheHandler() | ProactiveHandler shall not be available | EVT\_SEND\_DATA (SW – '90 00') | N5 |
| 5 | No Proactive Handler onCallback | | | | |
| - Send  EVT\_SEND\_DATA (INS = '01') | Service = CardEmulationService  Listener = CardEmulationListener  activateEvent(EVENT\_ON\_SEND\_DATA) during the installation  onCallback()  ProactiveHandlerSystem.getTheHandler() | ProactiveHandler shall not be available | EVT\_SEND\_DATA (SW – '90 00') | N5 |
| 6 | Use proactive functionality | | | | |
| - Send EVT\_SEND\_DATA with INS = '01'  - Send ENVELOPE (HCI Connectivity) on ISO interface after receiving HCI Connectivity Event | Listener = CardEmulationListener  process():  prepareAndSendConnectivityEvent()  register for EVENT\_EVENT\_DOWNLOAD\_HCI\_CONNECTIVITY  processToolkit():  use arbitrary method of the UICC API (ETSI TS 102 241 []) | No exception shall be thrown | - EVT\_SEND\_DATA (SW – '90 00')  - HCI event EVT\_CONNECTIVITY sent. This may be sent before or after the EVT\_SEND\_DATA indicated above.  - response to ENVELOPE | N1  N2 N3 |
| 7 | Send HCI Connectivity Event (reader mode) | | | | |
| - Send APDU(INS = '01') on ISO interface | process():  prepareAndSendConnectivityEvent() | No exception shall be thrown | HCI event EVT\_CONNECTIVITY sent.  - SW - '90 00' | N1  N3 |

##### 6.2.2.1.2 Method prepareAndSendTransactionEvent(byte[] aid, short aidOffset, short aidLen, byte[] parameters, short parametersOffset, short parametersLen)

Test Area Reference: Api\_2\_CNs\_Ste.

6.2.2.1.2.1 Conformance requirements

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

void prepareAndSendTransactionEvent(byte [] aid,

short aidOffset,

short aidLen,

byte[] parameters,

short parametersOffset,

short parametersLen)

throws HCIException

java.lang.ArrayIndexOutOfBoundsException,

java.lang.NullPointerException

6.2.2.1.2.1.1 Normal execution

* CRRN1: this non-blocking method builds the HCI event EVT\_TRANSACTION which notifies the terminal host that it shall launch an application on the terminal which is associated to an Applet in the UICC host identified.
* CRRN2: the contactless runtime environment shall bind the services defined in uicc.hci.services.connectivity to the corresponding resources (e.g. gates and pipes) specified by the HCI protocol [] for the connectivity service.
* CRRN3: The Contactless Framework shall only send the HCI event EVT\_CONNECTIVITY or EVT\_TRANSACTION specified by the HCI protocol [] to an Applet instance, when it is the selected Applet in card emulation mode or when this Applet instance is in the state ACTIVATED (according to "GlobalPlatform Amendment C" []) for the reader mode.

6.2.2.1.2.1.2 Parameter errors

* CRRP1: throw HCIException with error code reason HCI\_INVALID\_LENGTH if the parameter length or the AID is not compliant to ETSI TS 102 622 [].
* CRRP2: throw java.lang.ArrayIndexOutOfBoundsException - if operation would cause access of data outside array bounds.
* CRRP3: throw java.lang.NullPointerException - if parameters is null.

NOTE: Development of test cases for CRRP1 is FFS.

6.2.2.1.2.1.3 Context errors

* CRRC1: throw HCIException with error code reason HCI\_CURRENTLY\_DISABLED if the HCI interface was disabled.
* CRRC2: throw HCIException with error code reason HCI\_RESOURCES\_NOT\_AVAILABLE if the contactless framework does not have enough resources to process the command.
* CRRC3: throw HCIException with error code reason HCI\_CONDITIONS\_NOT\_SATISFIED if the conditions to call this method are not satisfied.

NOTE: CRRC2 is not tested, as it is not possible to force the situation where the contactless framework does not have enough resources to process the command.

6.2.2.1.2.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_2\_CNs\_Ste\_1.java | 1 |
| Api\_2\_CNs\_Ste\_1.java | 2-1 |
| Api\_2\_CNs\_Ste\_5.java | 2-2 |
| Api\_2\_CNs\_Ste\_4.java | 3 |
| Api\_2\_CNs\_Ste\_2.java | 4 |
| Api\_2\_CNs\_Ste\_6.java | 6 |

6.2.2.1.2.3 Initial condition

Initial conditions for ID 1, ID 3, ID 4:

* Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.
* In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.
* In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.
* UICC has created and opened a pipe to connectivity gate in the terminal host
* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI interface.

Initial conditions for ID 2-1:

* Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.
* In terminal profile set 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.
* In terminal profile the following CAT facilities shall be supported :
* Set the 25th byte, 'b6' to indicate that class m is supported
* Set the 26th byte, 'b2' and the 31st byte, 'b1' to indicate that class r is supported.
* Contactless functionality state is disabled in the UICC as defined in ETSI TS 102 223 [].

Initial conditions for ID2-2:

* Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.
* In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.
* In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

Initial conditions for ID 6:

* Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.
* In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.
* In terminal profile set 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.
* UICC has created and opened a pipe to connectivity gate in the terminal host
* According applet has been successfully installed and selected using ISO Interface and the applet is set into ACTIVATED state.

6.2.2.1.2.4 Test procedure

| **Test case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI commands** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Send HCI Transaction Event (card emulation)** | | | | |
| Send EVT\_SEND\_DATA with INS = '01' | extends CardEmulationListener  process():  prepareAndSendTransactionEvent()  aid = 'A00000000901…01'  aidOffset = 0  aidLen = 16  parameters = '01…01'  parametersOffset = 0  parametersLen = 10 | No exception shall be thrown | EVT\_SEND\_DATA (SW - '90 00')  HCI event EVT\_TRANSACTION sent. This may be sent before or after the EVT\_SEND\_DATA indicated above. | N1, N2,  N3 |
| 2 | **HCI interface disabled** | | | | |
| 1  Send on ISO interface:  - Send APDU to select the applet  - Send APDU (INS = '02') | extends CardEmulationListener  process():  prepareAndSendTransactionEvent() | throw HCIException with error code reason HCI\_CURRENTLY\_DISABLED | SW – '90 00'  No EVT\_TRANSACTION shall be sent | C1 |
| 2 The contactless interface is disabled in the UICC as defined in Global Platform Amendment C Send on ISO interface send the following commands:  - Send APDU to select the applet.  - Send APDU (INS = '02')  - Postcondition:  The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C | setCommunicationInterface() API method of Global Platform Amendment C [] is used to disable HCI interface  extends CardEmulationListener  process():  prepareAndSendTransactionEvent()  setCommunicationInterface() API method of  Global Platform Amendment C [] is used to enable again HCI interface | throw HCIException with error code reason HCI\_CURRENTLY\_DISABLED | SW – '90 00'  No EVT\_TRANSACTION shall be sent | C1 |
| 3 | **Wrong conditions** | | | | |
| The initial conditions in clause 6.2.2.1.2.3 related to card emulation are not applicable here.  Send on ISO:  - Select applet  - Send  APDU with (INS = '03') | Service = ConnectivityService  Listener = ConnectivityListener  process():  prepareAndSendTransactionEvent() | throw HCIException with error code reason HCI\_CONDITIONS\_NOT\_SATISFIED | SW - '90 00'  No EVT\_ TRANSACTION shall be sent | C3 |
| 4 | **Wrong parameters** | | | | |
|  | 1 - Send EVT\_SEND\_DATAwith INS = '01' | extends CardEmulationListener  process():  prepareAndSendTransactionEvent()  aid = 'A00000000901…01'  aidOffset = 20  aidLen = 16  parameters = '01…01'  parametersOffset = 0  parametersLen = 10 | throw java.lang.ArrayIndexOutOfBoundsException | EVT\_SEND\_DATA (SW - '90 00')  No EVT\_ TRANSACTION shall be sent | P2,  P3 |
|  | 2 - Send EVT\_SEND\_DATAwith INS = '02' | extends CardEmulationListener  process():  prepareAndSendTransactionEvent()  aid = 'A00000000901…01'  aidOffset = 0  aidLen = 20  parameters = '01…01'  parametersOffset = 0  parametersLen = 10 | throw java.lang.ArrayIndexOutOfBoundsException | EVT\_SEND\_DATA (SW - '90 00')  No EVT\_ TRANSACTION shall be sent | P2,  P3 |
|  | 3 - Send EVT\_SEND\_DATAwith INS = '03' | extends CardEmulationListener  process():  prepareAndSendTransactionEvent()  aid = 'A00000000901…01'  aidOffset = 0  aidLen = 16  parameters = '01…01'  parametersOffset = 20  parametersLen = 10 | throw java.lang.ArrayIndexOutOfBoundsException | EVT\_SEND\_DATA (SW - '90 00')  No EVT\_ TRANSACTION shall be sent | P2,  P3 |
|  | 4 - Send EVT\_SEND\_DATAwith INS = '04' | extends CardEmulationListener  process():  prepareAndSendTransactionEvent()  aid = 'A00000000901…01'  aidOffset = 0  aidLen = 16  parameters = '01…01'  parametersOffset = 0  parametersLen = 20 | throw java.lang.ArrayIndexOutOfBoundsException | EVT\_SEND\_DATA (SW - '90 00')  No EVT\_ TRANSACTION shall be sent | P2,  P3 |
|  | 5 - Send EVT\_SEND\_DATAwith INS = '05' | extends CardEmulationListener  process():  prepareAndSendTransactionEvent()  aid = null  aidOffset = 0  aidLen = 16  parameters = '01…01'  parametersOffset = 0  parametersLen = 10 | throw java.lang.NullPointerException | EVT\_SEND\_DATA (SW - '90 00')  No EVT\_ TRANSACTION shall be sent | P2,  P3 |
|  | 6 - Send EVT\_SEND\_DATAINS = '06' | extends CardEmulationListener  process():  prepareAndSendTransactionEvent()  aid = 'A00000000901…01'  aidOffset = 0  aidLen = 16  parameters = null  parametersOffset = 0  parametersLen = 10 | throw java.lang.NullPointerException | EVT\_SEND\_DATA (SW - '90 00')  No EVT\_ TRANSACTION shall be sent | P2,  P3 |
| 5 | **Void** | | | | |
| 6 | **Send HCI Transaction Event (reader mode)** | | | | |
|  | - Send APDU(INS = '01') on ISO interface | process():  prepareAndSendTransactionEvent()  aid = 'A00000000901…01'  aidOffset = 0  aidLen = 16  parameters = '01…01'  parametersOffset = 0  parametersLen = 10 | No exception shall be thrown | HCI event EVT\_TRANSACTION sent.  - SW - '90 00' | N1, N2,  N3 |

##### 6.2.2.1.3 Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen)

Test Area Reference: Api\_2\_CNs\_Stt.

6.2.2.1.3.1 Conformance requirements

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

void prepareAndSendTransactionEvent(byte[] parameters,

short parametersOffset,

short parametersLen)

throws HCIException,

java.lang.ArrayIndexOutOfBoundsException,

java.lang.NullPointerException

6.2.2.1.3.1.1 Normal execution

* CRRN1: this non-blocking method builds the HCI event EVT\_TRANSACTION which notifies the terminal host that it shall launch an application on the terminal which is associated to an Applet in the UICC host identified.
* CRRN2: the contactless runtime environment shall bind the services defined in uicc.hci.services.connectivity to the corresponding resources (e.g. gates and pipes) specified by the HCI protocol [] for the connectivity service.
* CRRNX: The Contactless Framework shall add the AID of the calling Applet instance to the HCI event passed to the CLF

NOTE: Development of test cases for CRRNX is FFS.

6.2.2.1.3.1.2 Parameter errors

* CRRP1: throw HCIException with error code reason HCI\_INVALID\_LENGTH if the parameter length or the AID is not compliant to ETSI TS 102 622 [].
* CRRP2: throw java.lang.ArrayIndexOutOfBoundsException - if operation would cause access of data outside array bounds.
* CRRP3: throw java.lang.NullPointerException - if parameters is null.

NOTE: Development of test cases for CRRP1 is FFS.

6.2.2.1.3.1.3 Context errors

* CRRC1: throw HCIException with error code reason HCI\_CURRENTLY\_DISABLED if the HCI interface was disabled.
* CRRC2: throw HCIException with error code reason HCI\_RESOURCES\_NOT\_AVAILABLE if the contactless framework does not have enough resources to process the command.
* CRRC3: throw HCIException with error code reason HCI\_CONDITIONS\_NOT\_SATISFIED if the conditions to call this method are not satisfied.

NOTE: CRRC2 is not tested, as it is not possible to force the situation where the contactless framework does not have enough resources to process the command.

6.2.2.1.3.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_2\_CNs\_Stt\_1.java | 1 |
| Api\_2\_CNs\_Stt\_1.java | 2-1 |
| Api\_2\_CNs\_Stt\_4.java | 2-2 |
| Api\_2\_CNs\_Stt\_3.java | 3 |
| Api\_2\_CNs\_Stt\_2.java | 4 |
| Api\_2\_CNs\_Stt\_5.java | 5 |

6.2.2.1.3.3 Initial condition

Initial conditions for ID 1, ID 3, ID 4:

* Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.
* In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.
* In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.
* UICC has created and opened a pipe to connectivity gate in the terminal host
* EVT\_FIELD\_ON has been sent on HCI interface.
* EVT\_CARD\_ACTIVATED has been sent on HCI interface.
* According applet has been successfully installed and selected using HCI interface.

Initial conditions for ID 2-1:

* Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.
* In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.
* In terminal profile the following CAT facilities shall be supported :
* Set the 25th byte, 'b6' to indicate that class m is supported
* Set the 26th byte, 'b2' and the 31st byte,'b1' to indicate that class r is supported.
* Contactless functionality state is disabled in the UICC as defined in ETSI TS 102 223 [].

Initial conditions for ID2-2:

* Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.
* In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.
* In terminal profile set the 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.

Initial conditions for ID 5:

* Prior to this test the Terminal shall have been powered on and performed the PROFILE DOWNLOAD procedure.
* In terminal profile set the 25th byte, 'b6' to indicate that class m is supported.
* In terminal profile set 5th byte, 'b1' to indicate proactive UICC: SET UP EVENT LIST.
* UICC has created and opened a pipe to connectivity gate in the terminal host
* According applet has been successfully installed and selected using ISO Interface and the applet is set into ACTIVATED state.

6.2.2.1.3.4 Test procedure

| **Test case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI commands** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Send HCI Transaction Event (card emulation)** | | | | |
| Send EVT\_SEND\_DATA with INS = '01' | extends CardEmulationListener  process():  prepareAndSendTransactionEvent()  parameters = '01…01'  parametersOffset = 0  parametersLen = 10 | No exception shall be thrown | EVT\_SEND\_DATA (SW - '90 00')  HCI event EVT\_TRANSACTION sent. This may be sent before or after the EVT\_SEND\_DATA indicated above. | N1, N2 |
| 2 | **HCI interface disabled** | | | | |
|  | 1  Send on ISO interface:  - Send APDU to select applet  - Send APDU (INS = '02') | extends CardEmulationListener  process():  prepareAndSendTransactionEvent() | throw HCIException with error code reason HCI\_CURRENTLY\_DISABLED | SW – '90 00'  No EVT\_TRANSACTION shall be sent | C1 |
|  | 2 The contactless interface is disabled in the UICC as defined in Global Platform Amendment C Send on ISO interface send the following commands:  - Send APDU to select the applet.  - Send APDU (INS = '02')  - Postcondition:  The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C | setCommunicationInterface() API method of Global Platform Amendment C [] is used to disable HCI interface  extends CardEmulationListener  process():  prepareAndSendTransactionEvent()  setCommunicationInterface() API method of  Global Platform Amendment C [] is used to enable again HCI interface | throw HCIException with error code reason HCI\_CURRENTLY\_DISABLED | SW – '90 00'  No EVT\_TRANSACTION shall be sent | C1 |
| 3 | **Wrong conditions** | | | | |
| The initial conditions in clause 6.2.2.1.3.3 related to card emulation are not applicable here.  Send on ISO interface:  - Select applet  - APDU data with INS = '03' | Service = ConnectivityService  Listener = ConnectivityListener    process():  prepareAndSendTransactionEvent() | throw HCIException with error code reason HCI\_CONDITIONS\_NOT\_SATISFIED | SW - '90 00'  No EVT\_ TRANSACTION shall be sent | C3 |
| 4 | **Wrong parameters** | | | | |
| 1 - Send EVT\_SEND\_DATA with INS = '01' | extends CardEmulationListener  process():  prepareAndSendTransactionEvent()  parameters = '01…01'  parametersOffset = 20  parametersLen = 10 | throw java.lang.ArrayIndexOutOfBoundsException | SW - '90 00'  No EVT\_ TRANSACTION shall be sent | P2,  P3 |
|  | 2 - Send EVT\_SEND\_DATA with INS = '02' | extends CardEmulationListener  process():  prepareAndSendTransactionEvent()  parameters = '01…01'  parametersOffset = 0  parametersLen = 20 | throw java.lang.ArrayIndexOutOfBoundsException | SW - '90 00'  No EVT\_ TRANSACTION shall be sent | P2,  P3 |
|  | 3 - Send EVT\_SEND\_DATA with INS = '03' | extends CardEmulationListener  process():  prepareAndSendTransactionEvent()  parameters = null  parametersOffset = 0  parametersLen = 10 | throw java.lang.NullPointerException | SW - '90 00'  No EVT\_ TRANSACTION shall be sent | P2,  P3 |
| 5 | **Send HCI Transaction Event (reader mode)** | | | | |
| - Send APDU(INS = '01') on ISO interface | process():  prepareAndSendTransactionEvent()  parameters = '01…01'  parametersOffset = 0  parametersLen = 10 | No exception shall be thrown | HCI event EVT\_TRANSACTION sent.  - SW - '90 00' | N1, N2,  N3 |

### 6.2.3 Package Reader Service

#### 6.2.3.1 Interface ReaderMessage

##### 6.2.3.1.1 Method restartReaderModeProcedure

Test Area Reference: Api\_2\_RMm\_Rrp.

6.2.3.1.1.1 Conformance requirements

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

void restartReaderModeProcedure()

throws HCIException

6.2.3.1.1.1.1 Normal execution

* CRRN1: this method sends first the HCI event EVT\_END\_OPERATION and then the HCI event EVT\_READER\_REQUESTED.
* CRRN2: this method is intended to recover from the following error states:
* reception of EVT\_TARGET\_DISCOVERED with a MULTIPLE\_TARGET\_STATUS.
* reception of RESP\_WR\_RF\_ERROR.
* reception of RESP\_ANY\_E\_TIMEOUT.
* CRRN3: an Applet shall only be able to activate ReaderListener.EVENT\_TARGET\_DISCOVERED or to use the restartReadermodeProcedure method if it is in lifecycle state ACTIVATED.

6.2.3.1.1.1.2 Parameter errors

* None.

6.2.3.1.1.1.3 Context errors

* CRRC1: throw HCIException with error code reason HCI\_CURRENTLY\_DISABLED if the HCI interface is disabled.
* CRRC2: throw HCIException with error code reason HCI\_CONDITIONS\_NOT\_SATISFIED if the Applet is not ACTIVATED.

NOTE 1: CRRC1 is only tested using the mechanism defined in Global Platform Amendment C [10]. For the mechanism defined in ETSI TS 102 223 [7], it is impossible to exercise this requirement: HCI would need to be disabled in advance, but this would mean that onCallback(ReaderMessage) could not be called, and the methods in ReaderMessage could therefore not be invoked.

NOTE 2: Development of test cases for CRRC2 is FFS.

6.2.3.1.1.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_2\_RMm\_Rrp\_1.java | 1 |
| Api\_2\_RMm\_Rrp\_1.java | 2 |
| Api\_2\_RMm\_Rrp\_1.java | 3 |
| Api\_2\_RMm\_Rrp\_2.java | 4 |
| Api\_2\_RMm\_Rrp\_3.java | 5-2 |

6.2.3.1.1.3 Initial condition

* According applet has been successfully installed and selected using ISO Interface and the applet is set into ACTIVATED state.
* The UICC has sent EVT\_READER\_REQUESTED on HCI interface.

6.2.3.1.1.4 Test procedure

| **Test case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **MULTIPLE\_TARGET\_STATUS is received - Type A** | | | | |
| Send EVT\_TARGET\_DISCOVERED(Status = '03') | restartReaderModeProcedure()  event = EVT\_END\_OPERATION  event = EVT\_READER\_REQUESTED | No exception shall be thrown | - EVT\_END\_OPERATION  - EVT\_READER\_REQUESTED | N1,  N2,  N3 |
| 2 | **MULTIPLE\_TARGET\_STATUS is received - Type B** | | | | |
| Send EVT\_TARGET\_DISCOVERED(Status = '03') | restartReaderModeProcedure()  event = EVT\_END\_OPERATION  event = EVT\_READER\_REQUESTED | No exception shall be thrown | - EVT\_END\_OPERATION  - EVT\_READER\_REQUESTED | N1,  N2,  N3 |
| 3 | **RESP\_WR\_RF\_ERROR is received** | | | | |
| - Send EVT\_TARGET\_DISCOVERED (status = '00')  -WR\_RF\_ERROR | - prepareAndSendWriteXchgDataCommand()  timeout = -1  data = '01…01'  offset = 0  len = 10  - restartReaderModeProcedure()  event = EVT\_END\_OPERATION  event = EVT\_READER\_REQUESTED | No exception shall be thrown | -  WR\_XCHG\_DATA  -EVT\_END\_OPERATION  - EVT\_READER\_REQUESTED | N1,  N2,  N3 |
| 4 | **RESP\_ANY\_E\_TIMEOUT is received** | | | | |
| - Send EVT\_TARGET\_DISCOVERED (status = '00')  - ANY\_E\_TIMEOUT | - prepareAndSendWriteXchgDataCommand()  timeout = -1  data = '01…01'  offset = 0  len = 10  - restartReaderModeProcedure()  event = EVT\_END\_OPERATION  event = EVT\_READER\_REQUESTED | No exception shall be thrown | - WR\_XCHG\_DATA  - EVT\_END\_OPERATION  - EVT\_READER\_REQUESTED | N1,  N2,  N3 |
| 5 | **HCI interface is disabled** | | | | |
| 1 - Void |  |  |  |  |
|  | 2 - Precondition: The contactless interface is disabled in the UICC as defined in Global Platform Amendment C Send on ISO interface send the following commands:  - Send EVT\_TARGET\_DISCOVERED (status = '03')  - Send APDU on ISO interface ('01')  - Postcondition:  The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C | setCommunicationInterface() API method of Global Platform Amendment C [] is used to disable HCI interface  - restartReaderModeProcedure()  event = EVT\_END\_OPERATION  event = EVT\_READER\_REQUESTED  setCommunicationInterface() API method of  Global Platform Amendment C [] is used to enable again HCI interface | Shall throw uicc.hci.framework.HCIException with error code HCI\_CURRENTLY\_DISABLED | - SW - '90 03'  No EVT\_READER\_REQUESTED shall be sent | C1 |

##### 6.2.3.1.2 Method prepareAndSendWriteXchgDataCommand

Test Area Reference: Api\_2\_RMm\_Srx

6.2.3.1.2.1 Conformance requirements

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

void prepareAndSendWriteXchgDataCommand(byte timeout,

byte[] data,

short offset,

short len)

throws HCIException,

java.lang.NullPointerException,

java.lang.ArrayIndexOutOfBoundsException,

javacard.framework.SystemException

6.2.3.1.2.1.1 Normal execution

* CRRN1: this non-blocking method builds the HCI command WR\_XCHG\_DATA with the data passed in the parameter data and sends it to the Contactless Reader Gate of the CLF.
* CRRN2: The response data from the CLF will be transferred to the UICC with the event ReaderListener.EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE.
* CRRN3: to be able to receive and send messages over the contactless interface in reader mode the applet shall activate the ReaderListener.EVENT\_TARGET\_DISCOVERED.
* CRRN4: an Applet has to be in the selectable state (according to the Java Card™ specification [], [] and []) to act as a contactless Applet in reader mode.

6.2.3.1.2.1.2 Parameter errors

* CRRP1: throw java.lang.NullPointerException if data is null.
* CRRP2: throw java.lang.ArrayIndexOutOfBoundsException if operation would cause access of data outside array bounds.
* CRRP3: throw javacard.framework.SystemException with the error code ILLEGAL\_VALUE if the timeout value does not match with the predefined values.

NOTE : Development of test cases for CRRP2 is FFS.

6.2.3.1.2.1.3 Context errors

* CRRC1: throw HCIException with error code reason HCI\_CURRENTLY\_DISABLED if the HCI interface was disabled.
* CRRC2: throw HCIException with error code reason HCI\_RESOURCES\_NOT\_AVAILABLE if the contactless framework does not have enough resources to process the command.

CRRC3: throws HCIException with the reason code HCI\_FRAGMENTED\_MESSAGE\_ONGOING if the Contactless Framework is still receiving a fragmented HCI Message.NOTE 1: CRRC1 is only tested using the mechanism defined in Global Platform Amendment C [10]. For the mechanism defined in ETSI TS 102 223 [7], it is impossible to exercise this requirement: HCI would need to be disabled in advance, but this would mean that onCallback(ReaderMessage) could not be called, and the methods in ReaderMessage could therefore not be invoked.

NOTE 2: CRRC2 is not tested, as it is not possible to force the situation where the contactless framework does not have enough resources to process the command.

NOTE 3: Development of test cases for CRRC3 is FFS.

6.2.3.1.2.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_2\_RMm\_Srx\_1.java | 1 |
| Api\_2\_RMm\_Srx\_1.java | 2 |
| Api\_2\_RMm\_Srx\_2.java | 3-1 |
| Api\_2\_RMm\_Srx\_3.java | 3-2 |
| Api\_2\_RMm\_Srx\_2.java | 4 |
| Api\_2\_RMm\_Srx\_2.java | 5 |
| Api\_2\_RMm\_Srx\_2.java | 6 |
| Api\_2\_RMm\_Srx\_2.java | 7 |

6.2.3.1.2.3 Initial condition

* According applet has been successfully installed and selected using ISO Interface and the applet is set into ACTIVATED state.
* The UICC has sent EVT\_READER\_REQUESTED on HCI interface.

6.2.3.1.2.4 Test procedure

| **Test case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Send Data - Type A** | | | | |
|  | 1 - Send APDU(INS = '01') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('90 00') | prepareAndSendWriteXchgDataCommand()  timeout = -1  data = '00 01 00 00'  offset = 0  len = 4 | No exception shall be thrown | WR\_XCHG\_DATA, with CTR = '0X' (where X is any value) and data sent = '00 01 00 00' | N1,  N2,  N3, N4 |
|  | 2 - Send APDU(INS = '02') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('01 90 00 00') | prepareAndSendWriteXchgDataCommand()  timeout = -1  data = '80 02 01 02 00'  offset = 0  len = 5 | No exception shall be thrown | -  WR\_XCHG\_DATA, with CTR = '0X' (where X is any value) and data sent = '80 02 01 02 00' | N1,  N2,  N3, N4 |
|  | 3 - Send APDU(INS = '03') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('90 00 00') | prepareAndSendWriteXchgDataCommand()  timeout = -1  data = 'A0 03 FE FF 02 3F 00'  offset = 0  len = 7 | No exception shall be thrown | -  WR\_XCHG\_DATA, with CTR = '0X' (where X is any value) and data sent = 'A0 03 FE FF 02 3F 00' | N1,  N2,  N3, N4 |
|  | 4 - Send APDU(INS = '04') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('01 90 00 00') | prepareAndSendWriteXchgDataCommand()  timeout = -1  data = '00 04 00 00 F8 01 … F8 00'  offset = 0  len = 254 | No exception shall be thrown | - WR\_XCHG\_DATA, with CTR = '0X' (where X is any value) and data sent = '00 04 00 00 F8 01 … F8 00' | N1,  N2,  N3, N4 |
|  | 5 - Send APDU(INS = '05') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('01 90 00 00') | prepareAndSendWriteXchgDataCommand()  timeout = 0  data = '00 05 00 00 02 3F 00 00'  offset = 0  len = 8 | No exception shall be thrown | -  WR\_XCHG\_DATA, with CTR = '10' and data sent = '00 05 00 00 02 3F 00 00' | N1,  N2,  N3, N4 |
|  | 6 - Send APDU(INS = '06') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('01 90 00 00') | prepareAndSendWriteXchgDataCommand()  timeout = 5  data = '00 06 00 00 02 3F 00 00'  offset = 0  len = 8 | No exception shall be thrown | -  WR\_XCHG\_DATA, with CTR = '15' and data sent = '00 06 00 00 02 3F 00 00' | N1,  N2,  N3, N4 |
|  | 7 - Send APDU(INS = '07') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('01 90 00 00') | prepareAndSendWriteXchgDataCommand()  timeout = 14  data = '00 07 00 00 02 3F 00 00'  offset = 0  len = 8 | No exception shall be thrown | -  WR\_XCHG\_DATA, with CTR = '1E' and data sent = '00 07 00 00 02 3F 00 00' | N1,  N2,  N3, N4 |
| 2 | **Send Data - Type B** | | | | |
|  | 1 - Send APDU(INS = '01') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('90 00') | prepareAndSendWriteXchgDataCommand()  timeout = -1  data = '00 01 00 00'  offset = 0  len = 4 | No exception shall be thrown | WR\_XCHG\_DATA, with CTR = '0X' (where X is any value) and data sent = '00 01 00 00' | N1,  N2,  N3, N4 |
|  | 2 - Send APDU(INS = '02') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('01 90 00 00') | prepareAndSendWriteXchgDataCommand()  timeout = -1  data = '80 02 01 02 00'  offset = 0  len = 5 | No exception shall be thrown | - WR\_XCHG\_DATA, with CTR = '0X' (where X is any value) and data sent = '80 02 01 02 00' | N1,  N2,  N3, N4 |
|  | 3 - Send APDU(INS = '03') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('90 00 00') | prepareAndSendWriteXchgDataCommand()  timeout = -1  data = 'A0 03 FE FF 02 3F 00'  offset = 0  len = 7 | No exception shall be thrown | - WR\_XCHG\_DATA, with CTR = '0X' (where X is any value) and data sent = 'A0 03 FE FF 02 3F 00' | N1,  N2,  N3, N4 |
|  | 4 - Send APDU(INS = '04') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('01 90 00 00') | prepareAndSendWriteXchgDataCommand()  timeout = -1  data = '00 04 00 00 F8 01 … F8 00'  offset = 0  len = 254 | No exception shall be thrown | - WR\_XCHG\_DATA, with CTR = '0X' (where X is any value) and data sent = '00 04 00 00 F8 01 … F8 00' | N1,  N2,  N3, N4 |
|  | 5 - Send APDU(INS = '05') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('01 90 00 00') | prepareAndSendWriteXchgDataCommand()  timeout = 0  data = '00 05 00 00 02 3F 00 00'  offset = 0  len = 8 | No exception shall be thrown | -  WR\_XCHG\_DATA, with CTR = '10' and data sent = '00 05 00 00 02 3F 00 00' | N1,  N2,  N3, N4 |
|  | 6 - Send APDU(INS = '06') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('01 90 00 00') | prepareAndSendWriteXchgDataCommand()  timeout = 5  data = '00 06 00 00 02 3F 00 00'  offset = 0  len = 8 | No exception shall be thrown | -  WR\_XCHG\_DATA, with CTR = '15' and data sent = '00 06 00 00 02 3F 00 00' | N1,  N2,  N3, N4 |
|  | 7 - Send APDU(INS = '07') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK('01 90 00 00') | prepareAndSendWriteXchgDataCommand()  timeout = 14  data = '00 07 00 00 02 3F 00 00'  offset = 0  len = 8 | No exception shall be thrown | -  WR\_XCHG\_DATA, with CTR = '1E' and data sent = '00 07 00 00 02 3F 00 00' | N1,  N2,  N3, N4 |
| 3 | **HCI interface is disabled** | | | | |
|  | 1 - Precondition:  The terminal shall indicate the support of class r by setting the 26th byte, 'b2' and the 31st byte, 'b1' in the terminal profile and disable the contactless functionality in the UICC as defined in ETSI TS 102 223 []  - Send APDU on ISO interface to select the applet  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send APDU on ISO interface (INS = '11') | prepareAndSendWriteXchgDataCommand()  timeout = 14  data = '01…01'  offset = 0  len = 10 | Shall throw uicc.hci.framework.HCIException with error code HCI\_CURRENTLY\_DISABLED | - SW - '90 01' | C1 |
|  | 2 - Precondition: The contactless interface is disabled in the UICC as defined in Global Platform Amendment C Send on ISO interface send the following commands:  - Send APDU on ISO interface to select the applet  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send APDU on ISO interface (INS = '01')  - Postcondition:  The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C | setCommunicationInterface() API method of Global Platform Amendment C [] is used to disable HCI interface  prepareAndSendWriteXchgDataCommand()  timeout = 14  data = '01…01'  offset = 0  len = 10  setCommunicationInterface() API method of  Global Platform Amendment C [] is used to enable again HCI interface | Shall throw uicc.hci.framework.HCIException with error code HCI\_CURRENTLY\_DISABLED | - SW - '90 01' | C1 |
| 4 | **Data is null** | | | | |
| - Send APDU on ISO interface (INS = '02')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send APDU on ISO interface ('12') | prepareAndSendWriteXchgDataCommand()  timeout = 14  data = null  offset = 0  len = 10 | Shall throw java.lang.NullPointerException | - SW - '90 02' | P1 |
| 5 | **ArrayIndexOutOfBounds Exception** | | | | |
| - Send APDU on ISO interface (INS = '03')  - Send EVT\_TARGET\_DISCOVERED (status = 00)  - Send APDU on ISO interface ('13') | prepareAndSendWriteXchgDataCommand()  timeout = 14  data = '01…01'  offset = -1  len = 10 | Shall throw java.lang. ArrayIndexOutOfBoundsException | - SW - '90 03' | P2 |
| 6 | **ArrayIndexOutOfBounds Exception** | | | | |
| - Send APDU on ISO interface (INS = '04')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send APDU on ISO interface (INS = '14') | prepareAndSendWriteXchgDataCommand()  timeout = 14  data = '01…01'  offset = 0  len = 11 | Shall throw java.lang. ArrayIndexOutOfBoundsException | - SW - '90 04' | P2 |
| 7 | **Timeout has illegal value** | | | | |
| - Send APDU on ISO interface (INS = '05')  - Send EVT\_TARGET\_DISCOVERED (status = 00)  - Send APDU on ISO interface (INS = '15') | prepareAndSendWriteXchgDataCommand()  timeout = -2  data = '01…01'  offset = 0  len = 10 | Shall throw uicc.hci.framework.HCIException with error code ILLEGAL\_VALUE | - SW - '90 05' | P3 |

##### 6.2.3.1.3 Method prepareAndSendGetParameterCommand

Test Area Reference: Api\_2\_RMm\_Sgp.

6.2.3.1.3.1 Conformance requirements

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

void prepareAndSendGetParameterCommand(byte paramid)

throws HCIException,

javacardframework.SystemException

6.2.3.1.3.1.1 Normal execution

* CRRN1: This non-blocking method builds the HCI command ANY\_GET\_PARAMETER for the indicated information and sends it to the Contactless Reader Gate of the CLF.
* CRRN2: this method should only be called when the CLF has successfully discovered a card in the field, i.e. after reception of the HCI event EVT\_TARGET\_DISCOVERED.
* CRRN3: paramid shall be one of the PARAM\_\* values defined in this interface.

6.2.3.1.3.1.2 Parameter errors

* CRRP1: throw javacard.framework.SystemException - with the following reason code ILLEGAL\_VALUE if the paramID does not match with the predefined values.

6.2.3.1.3.1.3 Context errors

* CRRC1: throw HCIException with error code reason HCI\_CURRENTLY\_DISABLED if the HCI interface was disabled.
* CRRC2: throw HCIException with error code reason HCI\_RESOURCES\_NOT\_AVAILABLE if the contactless framework does not have enough resources to process the command.
* CRRC3: throws HCIException with the reason code HCI\_FRAGMENTED\_MESSAGE\_ONGOING if the Contactless Framework is still receiving a fragmented HCI Message.

NOTE 1: CRRC1 is only tested using the mechanism defined in Global Platform Amendment C [10]. For the mechanism defined in ETSI TS 102 223 [7], it is impossible to exercise this requirement: HCI would need to be disabled in advance, but this would mean that onCallback(ReaderMessage) could not be called, and the methods in ReaderMessage could therefore not be invoked.

NOTE 2: CRRC2 is not tested, as it is not possible to force the situation where the contactless framework does not have enough resources to process the command.

NOTE 3: Development of test cases for CRRC3 is FFS.

6.2.3.1.3.2 Test suite files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_2\_RMm\_Sgp\_1.java | 1 |
| Api\_2\_RMm\_Sgp\_1.java | 2 |
| Api\_2\_RMm\_Sgp\_3.java | 3-2 |
| Api\_2\_RMm\_Sgp\_2.java | 5 |

6.2.3.1.3.3 Initial condition

* According applet has been successfully installed and selected using ISO Interface and the applet is set into ACTIVATED state.
* The UICC has sent EVT\_READER\_REQUESTED on HCI interface.

6.2.3.1.3.4 Test procedure

| **Test case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **Get Parameter: Type A** | | | | |
|  | 1 - Send APDU on ISO interface (INS = '01')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK as response to GET\_PARAMETER  - Send APDU on ISO interface (INS = '20') | prepareAndSendGetParameterCommand()  paramid = PARAM\_ID\_TYPE\_A\_READER\_UID    Response received:  onCallback()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown | - GET\_PARAMETER with command parameter as indicated in the API Description column  - APDU-Response '0100' | N1,  N2,  N3 |
|  | 2 - Send APDU on ISO interface (INS = '02')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK as response to GET\_PARAMETER  - Send APDU on ISO interface (INS = '20') | prepareAndSendGetParameterCommand()  paramid = PARAM\_ID\_TYPE\_A\_READER\_ATQA  Response received:  onCallback()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown | - GET\_PARAMETER with command parameter as indicated in the API Description column  - APDU-Response '0200' | N1,  N2,  N3 |
|  | 3 - Send APDU on ISO interface (INS = '03')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK as response to GET\_PARAMETER  - Send APDU on ISO interface (INS = '20') | prepareAndSendGetParameterCommand()  paramid = PARAM\_ID\_TYPE\_A\_READER\_APPLICATION\_DATA  Response received:  onCallback()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown | - GET\_PARAMETER with command parameter as indicated in the API Description column  - APDU-Response '0300' | N1,  N2,  N3 |
|  | 4 - Send APDU on ISO interface (INS = '04')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK as response to GET\_PARAMETER  - Send APDU on ISO interface (INS = '20') | prepareAndSendGetParameterCommand()  paramid = PARAM\_ID\_TYPE\_A\_READER\_SAK  Response received:  onCallback()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown. | - GET\_PARAMETER with command parameter as indicated in the API Description column  - APDU-Response '0400' | N1,  N2,  N3 |
|  | 5 - Send APDU on ISO interface (INS = '05')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK as response to GET\_PARAMETER  - Send APDU on ISO interface (INS = '20') | - prepareAndSendGetParameterCommand()  paramid = PARAM\_ID\_TYPE\_A\_READER\_FWI  Response received:  onCallback()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown | - GET\_PARAMETER with command parameter as indicated in the API Description column  - APDU-Response '0500' | N1,  N2,  N3 |
|  | 6 - Send APDU on ISO interface (INS = '06')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK as response to GET\_PARAMETER  - Send APDU on ISO interface (INS = '20') | prepareAndSendGetParameterCommand()  paramid =  PARAM\_ID\_TYPE\_A\_READER\_DATARATE\_MAX  Response received:  onCallback()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown | - GET\_PARAMETER with command parameter as indicated in the API Description column  - APDU-Response '0600' | N1,  N2,  N3 |
| 2 | **Get Parameter: Type B** | | | | |
|  | 1 - Send APDU on ISO interface (INS = '07')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK as response to GET\_PARAMETER  - Send APDU on ISO interface (INS = '20') | prepareAndSendGetParameterCommand()  paramid =  PARAM\_ID\_TYPE\_B\_READER\_PUPI  Response received:  onCallback()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown | - GET\_PARAMETER with command parameter as indicated in the API Description column  - APDU-Response '0700' | N1,  N2,  N3 |
|  | 2 - Send APDU on ISO interface (INS = '08')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK as response to GET\_PARAMETER  - Send APDU on ISO interface (INS = '20') | prepareAndSendGetParameterCommand()  paramid =  PARAM\_ID\_TYPE\_B\_READER\_APPLICATION\_DATA  Response received:  onCallback()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown | - GET\_PARAMETER with command parameter as indicated in the API Description column  - APDU-Response '0800' | N1,  N2,  N3 |
|  | 3 - Send APDU on ISO interface (INS = '09')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK as response to GET\_PARAMETER  - Send APDU on ISO interface (INS = '20') | prepareAndSendGetParameterCommand()  paramid =  PARAM\_ID\_TYPE\_B\_READER\_AFI  Response received:  onCallback()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown | - GET\_PARAMETER with command parameter as indicated in the API Description column  - APDU-Response '0900' | N1,  N2,  N3 |
|  | 4 - Send APDU on ISO interface (INS = '0A')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK as response to GET\_PARAMETER  - Send APDU on ISO interface (INS = '20') | prepareAndSendGetParameterCommand()  paramid =  PARAM\_ID\_TYPE\_B\_READER\_HIGHER\_LAYER\_RESPONSE  Response received:  onCallback()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown | - GET\_PARAMETER with command parameter as indicated in the API Description column  - APDU-Response '0A00' | N1,  N2,  N3 |
|  | 5 - Send APDU on ISO interface (INS = '0B')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK as response to GET\_PARAMETER  - Send APDU on ISO interface (INS = '20') | prepareAndSendGetParameterCommand()  paramid =  PARAM\_ID\_TYPE\_B\_READER\_HIGHER\_LAYER\_DATA  Response received:  onCallback()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown | - GET\_PARAMETER with command parameter as indicated in the API Description column  - APDU-Response '0B00' | N1,  N2,  N3 |
| 3 | **HCI interface is disabled** | | | | |
|  | 1 - Void |  |  |  |  |
|  | 2 - Precondition: The contactless interface is disabled in the UICC as defined in Global Platform Amendment C Send on ISO interface send the following commands:  - Send APDU on ISO interface to select the applet  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send APDU on ISO interface (INS = '11')  - Postcondition:  The contactless interface is enabled again in the UICC as defined in Global Platform Amendment C | setCommunicationInterface() API method of Global Platform Amendment C [] is used to disable HCI interface  prepareAndSendGetParameterCommand()  paramid =  PARAM\_ID\_TYPE\_A\_READER\_UID  setCommunicationInterface() API method of  Global Platform Amendment C [] is used to enable again HCI interface | Shall throw uicc.hci.framework.HCIException with error code  HCI\_CURRENTLY\_DISABLED | - SW - '90 03' | C1 |
| 4 | **Void** | | | | |
| 5 | **The parameter has an illegal value** | | | | |
| - Send APDU on ISO interface (INS = '05')  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send APDU on ISO interface (INS = '15') | - prepareAndSendGetParameterCommand()  paramid = -1 | Shall throw uicc.hci.framework.HCIException with error code  ILLEGAL\_VALUE | - SW - '90 05' | P1 |

#### 6.2.3.2 Interface ReaderListener

##### 6.2.3.2.1 Method onCallback

Test Area Reference: Api\_2\_Rml\_Ocb.

6.2.3.2.1.1 Conformance requirements

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

void onCallback(byte event, HCIMessage message)

6.2.3.2.1.1.1 Normal execution

* CRRN1: The Contactless Framework shall invoke the method ReaderListener.onCallback only with events which are defined in ReaderListener or HCIListener and only with ReaderMessage objects.
* CRRN2: The Applet instance shall activate the events with ReaderService.activateEvent before it will receive any event notification.

6.2.3.2.1.1.2 Parameter errors

* None.

6.2.3.2.1.1.3 Context errors

* None.

6.2.3.2.1.2 Test Suite Files

|  |  |
| --- | --- |
| Applet Name | Test case ID |
| Api\_2\_Rml\_Ocb\_1.java | 1 |
| Api\_2\_Rml\_Ocb\_1.java | 2 |
| Api\_2\_Rml\_Ocb\_1.java | 4 |
| Api\_2\_Rml\_Ocb\_1.java | 5 |
| Api\_2\_Rml\_Ocb\_1.java | 6 |
| Api\_2\_Rml\_Ocb\_1.java | 8 |

6.2.3.2.1.3 Initial conditions

* According applet has been successfully installed and selected using ISO Interface and the applet is set into ACTIVATED state.
* The UICC has sent EVT\_READER\_REQUESTED on HCI interface.

6.2.3.2.1.4 Test procedure

| **Test Case** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **ID** | **HCI Command** | **API Description** | **API Expectation** | **HCI Response** | **CRR** |
| 1 | **EVENT\_TARGET\_DISCOVERED - Type A** | | | | |
| - send APDU (INS = '10') on ISO interface  - Send EVENT\_TARGET\_DISCOVERED (status = '00') | HCIService = ReaderService  activateEvent()  event = EVENT\_TARGET\_DISCOVERED | No exception shall be thrown.  onCallback() method shall notice the event | WR\_XCHG\_DATA ('00 01 00 00') | N1,  N2 |
| **EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE - Type A** | | | | | |
| 2 | - send APDU(INS = '11') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK () | HCIService = ReaderService  activateEvent()  event = EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE | No exception shall be thrown.  onCallback() method shall notice the event | Start bulk data transfer on HCI interface. | N1,  N2 |
| **Void** | | | | | |
| **EVENT\_GET\_PARAMETER\_RESPONSE - Type A** | | | | | |
| 4 | - send APDU(INS = '13') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK () | HCIService = ReaderService  activateEvent()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown.  onCallback() method shall notice the event | ANY\_GET\_PARAMETER | N1,  N2 |
| 5 | **EVENT\_TARGET\_DISCOVERED - Type B** | | | | |
| - send APDU (INS = '10') on ISO interface  - Send EVENT\_TARGET\_DISCOVERED (status = '00') | HCIService = ReaderService  activateEvent()  event = EVENT\_TARGET\_DISCOVERED | No exception shall be thrown.  onCallback() method shall notice the event | WR\_XCHG\_DATA ('00 01 00 00') | N1,  N2 |
| **EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE - Type B** | | | | | |
| 6 | - send APDU(INS = '11') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK () | HCIService = ReaderService  activateEvent()  event = EVENT\_WRITE\_EXCHANGE\_DATA\_RESPONSE | No exception shall be thrown.  onCallback() method shall notice the event | Start bulk data transfer on HCI interface. | N1,  N2 |
| **7** | **Void** | | | | |
| **EVENT\_GET\_PARAMETER\_RESPONSE - Type B** | | | | | |
| 8 | - send APDU(INS = '13') on ISO interface  - Send EVT\_TARGET\_DISCOVERED (status = '00')  - Send ANY\_OK () | HCIService = ReaderService  activateEvent()  event = EVENT\_GET\_PARAMETER\_RESPONSE | No exception shall be thrown.  onCallback() method shall notice the event | ANY\_GET\_PARAMETER | N1,  N2 |

Annex A (normative):  
Class, methods and tests acronyms

# A.1 HCI framework

|  |  |
| --- | --- |
| Class name | Acronyms |
| HCIDevice | Hdv |
| HCISevice | HSr |
| HCIMessage | Hme |
| HCIListener | Hln |
| HCIException | Hxp |

## A.1.1 Class HCIDevice

|  |  |
| --- | --- |
| Method name | Acronyms |
| public static HCIService getHCIService() | Gsr |
| public static byte getPowerMode() | Gpm |
| public static byte isHCIServiceAvailable() | Isa |

## A.1.2 Interface HCIService

|  |  |
| --- | --- |
| Method name | Acronyms |
| void register() | Reg |
| void deregister() | Drg |
| void activateEvent() | Ace |
| void deactivateEvent() | Dae |
| void requestCallbackNotification() | Rcn |
| boolean getEventNotificationStatus() | Gen |

## A.1.3 Interface HCIMessage

|  |  |
| --- | --- |
| Method Name | Acronyms |
| boolean isHeading() | Mhd |
| boolean isComplete() | Mco |
| byte getType() | Mty |
| byte getInstruction() | Min |
| short getReceiveOffset() | Mro |
| short getReceiveLength() | Mrl |
| byte[] getReceiveBuffer() | Mrb |

## A.1.4 Interface HCIListener

|  |  |
| --- | --- |
| Method Name | Acronyms |
| void onCallback() | Ocb |

## A.1.5 Class HCIException

| Method Name | Acronyms |
| --- | --- |
| public static void throwIt() | Trw |

# A.2 HCI Services

## A.2.1 Package cardemulation

|  |  |
| --- | --- |
| Interface Name | Acronyms |
| CardEmulationMessage | CEm |
| CardEmulationService | CEs |

### A.2.1.1 Interface CardEmulationListener

|  |  |
| --- | --- |
| Method Name | Acronyms |
| void onCallback() | Ocb |

### A.2.1.2 Interface CardEmulationMessage

|  |  |
| --- | --- |
| Method Name | Acronyms |
| void prepareAndSendGetParameterCommand() | Sgp |
| void prepareAndSendSendDataEvent() | Ssd |
| boolean selectingMessage() | Scm |

### A.2.1.3 Interface CardEmulationService

|  |  |
| --- | --- |
| Method Name | Acronyms |
| byte getCardRFType() | Rft |

## A.2.2 Package connectivity

|  |  |
| --- | --- |
| Interface Name | Acronyms |
| ConnectivityMessage | CNm |
| ConnectivityService | CNs |

### A.2.2.1 Interface ConnectivityListener

FFS

### A.2.2.2 Interface ConnectivityMessage

FFS

### A.2.2.3 Interface ConnectivityService

|  |  |
| --- | --- |
| Method Name | Acronyms |
| prepareAndSendConnectivityEvent() | Sce |
| prepareAndSendTransactionEvent(byte[] aid, short aidOffset, short aidLen, byte[] parameters, short parametersOffset, short parametersLen) | Ste |
| prepareAndSendTransactionEvent(byte[] parameters, short parametersOffset, short parametersLen) | Stt |

## A.2.3 Readermode

### A.2.3.1 Interface RaederListener

|  |  |
| --- | --- |
| Method Name | Acronyms |
| void onCallback() | Ocb |

### A.2.3.2 Interface ReaderMessage

|  |  |
| --- | --- |
| Method Name | Acronyms |
| void prepareAndSendGetParameterCommand() | Sgp |
| void prepareAndSendWriteXchgDataCommand() | Srx |
| void restartReaderModeProcedure() | Rrp |

### A.2.3.3 Interface ReaderService

FFS

Annex B (normative):  
AIDs - to be reserved

# B.1 Package HCI framework

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.framework | A0 00 00 00 09 00 05 FF FF FF FF 89 21 00 00 00 |

## B.1.1 Class HCIDevice

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.framework. Api\_1\_Hdv\_Gsr | A0 00 00 00 09 00 05 FF FF FF FF 89 21 01 00 00 |
| uicc.test.hci.framework. Api\_1\_Hdv\_Gpm | A0 00 00 00 09 00 05 FF FF FF FF 89 21 02 00 00 |
| uicc.test.hci.framework. Api\_1\_Hdv\_Isa | A0 00 00 00 09 00 05 FF FF FF FF 89 21 03 00 00 |

|  |  |
| --- | --- |
| Applet name | AID |
| Api\_1\_Hdv\_Gsr\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 01 01 02 |
| Api\_1\_Hdv\_Gsr\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 01 02 02 |
| Api\_1\_Hdv\_Gpm\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 02 01 02 |
| Api\_1\_Hdv\_Isa\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 03 01 02 |
| Api\_1\_Hdv\_Isa\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 03 02 02 |
| Api\_1\_Hdv\_Isa\_3.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 03 03 02 |

## B.1.2 Interface HCIService

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.framework. Api\_1\_Hsr\_Reg | A0 00 00 00 09 00 05 FF FF FF FF 89 21 04 00 00 |
| uicc.test.hci.framework. Api\_1\_Hsr\_Drg | A0 00 00 00 09 00 05 FF FF FF FF 89 21 05 00 00 |
| uicc.test.hci.framework. Api\_1\_Hsr\_Ace | A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 00 00 |
| uicc.test.hci.framework. Api\_1\_Hsr\_Dae | A0 00 00 00 09 00 05 FF FF FF FF 89 21 07 00 00 |
| uicc.test.hci.framework. Api\_1\_Hsr\_Rcn | A0 00 00 00 09 00 05 FF FF FF FF 89 21 08 00 00 |
| uicc.test.hci.framework. Api\_1\_Hsr\_Gen | A0 00 00 00 09 00 05 FF FF FF FF 89 21 09 00 00 |

| Applet name | AID |
| --- | --- |
| Api\_1\_Hsr\_Reg\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 04 01 02 |
| Api\_1\_Hsr\_Reg\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 04 02 02 |
| Api\_1\_Hsr\_Reg\_3.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 04 03 02 |
| Api\_1\_Hsr\_Reg\_4.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 04 04 02 |
| Api\_1\_Hsr\_Drg\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 05 01 02 |
| Api\_1\_Hsr\_Drg\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 05 02 02 |
| Api\_1\_Hsr\_Ace\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 01 02 |
| Api\_1\_Hsr\_Ace\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 02 02 |
| Api\_1\_Hsr\_Ace\_3.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 03 02 |
| Api\_1\_Hsr\_Ace\_4.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 04 02 |
| Api\_1\_Hsr\_Ace\_5.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 05 02 |
| Api\_1\_Hsr\_Ace\_6.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 06 02 |
| Api\_1\_Hsr\_Ace\_7.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 07 02 |
| Api\_1\_Hsr\_Ace\_8.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 08 02 |
| Api\_1\_Hsr\_Ace\_9.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 06 09 02 |
| Api\_1\_Hsr\_Dae\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 07 01 02 |
| Api\_1\_Hsr\_Dae\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 07 02 02 |
| Api\_1\_Hsr\_Dae\_3.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 07 03 02 |
| Api\_1\_Hsr\_Dae\_4.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 07 04 02 |
| Api\_1\_Hsr\_Dae\_6.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 07 06 02 |
| Api\_1\_Hsr\_Rcn\_7.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 08 07 02 |
| Api\_1\_Hsr\_Gen\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 09 01 02 |
| Api\_1\_Hsr\_Gen\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 09 02 02 |

## B.1.3 Interface HCIMessage

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.framework. Api\_1\_Hme\_Mhd | A0 00 00 00 09 00 05 FF FF FF FF 89 21 0A 00 00 |
| uicc.test.hci.framework. Api\_1\_Hme\_Mco | A0 00 00 00 09 00 05 FF FF FF FF 89 21 0B 00 00 |
| uicc.test.hci.framework. Api\_1\_Hme\_Mty | A0 00 00 00 09 00 05 FF FF FF FF 89 21 0C 00 00 |
| uicc.test.hci.framework. Api\_1\_Hme\_Min | A0 00 00 00 09 00 05 FF FF FF FF 89 21 0D 00 00 |
| uicc.test.hci.framework. Api\_1\_Hme\_Mro | A0 00 00 00 09 00 05 FF FF FF FF 89 21 0E 00 00 |
| uicc.test.hci.framework. Api\_1\_Hme\_Mrl | A0 00 00 00 09 00 05 FF FF FF FF 89 21 0F 00 00 |
| uicc.test.hci.framework. Api\_1\_Hme\_Mrb | A0 00 00 00 09 00 05 FF FF FF FF 89 21 10 00 00 |

|  |  |
| --- | --- |
| Applet name | AID |
| Api\_1\_Hme\_Mhd\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 0A 01 02 |
| Api\_1\_Hme\_Mco\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 0B 01 02 |
| Api\_1\_Hme\_Mty\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 0C 01 02 |
| Api\_1\_Hme\_Mty\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 0C 02 02 |
| Api\_1\_Hme\_Min\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 0D 01 02 |
| Api\_1\_Hme\_Min\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 0D 02 02 |
| Api\_1\_Hme\_Mrl\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 0F 01 02 |
| Api\_1\_Hme\_Mrb\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 10 01 02 |

## B.1.4 Interface HCIListener

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.framework. Api\_1\_Hln\_Ocb | A0 00 00 00 09 00 05 FF FF FF FF 89 21 11 00 00 |

|  |  |
| --- | --- |
| Applet name | AID |
| Api\_1\_Hln\_Ocb\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 11 02 02 |
| Api\_1\_Hln\_Ocb\_3.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 11 03 02 |
| Api\_1\_Hln\_Ocb\_4.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 11 04 02 |
| Api\_1\_Hln\_Ocb\_5.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 11 05 02 |
| Api\_1\_Hln\_Ocb\_6.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 11 06 02 |

## B.1.5 Class HCIException

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.framework. Api\_1\_Hxp\_Trw | A0 00 00 00 09 00 05 FF FF FF FF 89 21 12 00 00 |

|  |  |
| --- | --- |
| Applet name | AID |
| Api\_1\_Hxp\_Trw\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 21 12 01 02 |

# B.2 HCI Services

## B.2.1 Package cardemulation

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.services.cardemulation | A0 00 00 00 09 00 05 FF FF FF FF 89 25 00 00 00 |

### B.2.1.1 Interface CardEmulationListener

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.services.cardemulation.Api\_2\_CEl\_Ocb | A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 00 00 |

|  |  |
| --- | --- |
| Applet name | AID |
| Api\_2\_CEl\_Ocb\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 01 02 |
| Api\_2\_CEl\_Ocb\_3.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 03 02 |
| Api\_2\_CEl\_Ocb\_4.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 04 02 |
| Api\_2\_CEl\_Ocb\_5.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 05 02 |
| Api\_2\_CEl\_Ocb\_6.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 06 02 |
| Api\_2\_CEl\_Ocb\_7.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 07 02 |
| Api\_2\_CEl\_Ocb\_8.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 08 02 |

### B.2.1.2 Interface CardEmulationMessage

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.services.cardemulation.Api\_2\_CEm\_Sgp | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 00 00 |
| uicc.test.hci.services.cardemulation.Api\_2\_CEm\_Ssd | A0 00 00 00 09 00 05 FF FF FF FF 89 25 02 00 00 |
| uicc.test.hci.services.cardemulation.Api\_2\_CEm\_Scm | A0 00 00 00 09 00 05 FF FF FF FF 89 25 03 00 00 |

|  |  |
| --- | --- |
| Applet name | AID |
| Api\_2\_CEm\_Sgp\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 01 02 |
| Api\_2\_CEm\_Sgp\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 02 02 |
| Api\_2\_CEm\_Sgp\_3.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 03 02 |
| Api\_2\_CEm\_Sgp\_4.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 04 02 |
| Api\_2\_CEm\_Sgp\_5.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 05 02 |
| Api\_2\_CEm\_Sgp\_6.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 06 02 |
| Api\_2\_CEm\_Sgp\_7.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 07 02 |
| Api\_2\_CEm\_Sgp\_8.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 08 02 |
| Api\_2\_CEm\_Sgp\_9.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 09 02 |
| Api\_2\_CEm\_Sgp\_10.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 0A 02 |
| Api\_2\_CEm\_Sgp\_11.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 0B 02 |
| Api\_2\_CEm\_Sgp\_12.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 0C 02 |
| Api\_2\_CEm\_Sgp\_13.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 0D 02 |
| Api\_2\_CEm\_Sgp\_14.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 0E 02 |
| Api\_2\_CEm\_Sgp\_15.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 01 0F 02 |
| Api\_2\_CEm\_Ssd\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 02 01 02 |
| Api\_2\_CEm\_Ssd\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 02 02 02 |
| Api\_2\_CEm\_Ssd\_3.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 02 03 02 |
| Api\_2\_CEm\_Ssd\_4.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 02 04 02 |
| Api\_2\_CEm\_Scm\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 03 01 02 |

### B.2.1.3 Interface CardEmulationService

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.services.cardemulation.Api\_2\_CEs\_RFt | A0 00 00 00 09 00 05 FF FF FF FF 89 25 04 00 00 |

|  |  |
| --- | --- |
| Applet name | AID |
| Api\_2\_CEs\_RFt\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 25 04 01 02 |

## B.2.2 Package connectivity

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.services.connectivity | A0 00 00 00 09 00 05 FF FF FF FF 89 26 00 00 00 |

### B.2.2.1 Interface ConnectivityListener

FFS

### B.2.2.2 Interface ConnectivityMessage

FFS

### B.2.2.3 Interface ConnectivityService

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.services.connectivity.Api\_2\_CNs\_Sce | A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 00 00 |
| uicc.test.hci.services.connectivity.Api\_2\_CNs\_Ste | A0 00 00 00 09 00 05 FF FF FF FF 89 26 02 00 00 |
| uicc.test.hci.services.connectivity.Api\_2\_CNs\_Stt | A0 00 00 00 09 00 05 FF FF FF FF 89 26 03 00 00 |

|  |  |
| --- | --- |
| Applet name | AID |
| Api\_2\_CNs\_Sce\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 01 02 |
| Api\_2\_CNs\_Sce\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 02 02 |
| Api\_2\_CNs\_Sce\_3.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 03 02 |
| Api\_2\_CNs\_Sce\_4.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 04 02 |
| Api\_2\_CNs\_Sce\_5.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 05 02 |
| Api\_2\_CNs\_Sce\_6.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 06 02 |
| Api\_2\_CNs\_Sce\_7.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 01 07 02 |
| Api\_2\_CNs\_Ste\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 02 01 02 |
| Api\_2\_CNs\_Ste\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 02 02 02 |
| Api\_2\_CNs\_Ste\_4.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 02 04 02 |
| Api\_2\_CNs\_Ste\_5.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 02 05 02 |
| Api\_2\_CNs\_Ste\_6.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 02 06 02 |
| Api\_2\_CNs\_Stt\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 03 01 02 |
| Api\_2\_CNs\_Stt\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 03 02 02 |
| Api\_2\_CNs\_Stt\_3.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 03 03 02 |
| Api\_2\_CNs\_Stt\_4.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 03 04 02 |
| Api\_2\_CNs\_Stt\_5.java | A0 00 00 00 09 00 05 FF FF FF FF 89 26 03 05 02 |

## B.2.3 Package readermode

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.services.reader | A0 00 00 00 09 00 05 FF FF FF FF 89 27 00 00 00 |

### B.2.3.1 Interface ReaderMessage

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.services.reader.Api\_2\_RMm\_Rrp | A0 00 00 00 09 00 05 FF FF FF FF 89 27 01 00 00 |
| uicc.test.hci.services.reader.Api\_2\_RMm\_Srx | A0 00 00 00 09 00 05 FF FF FF FF 89 27 02 00 00 |
| uicc.test.hci.services.reader.Api\_2\_RMm\_Sgp | A0 00 00 00 09 00 05 FF FF FF FF 89 27 03 00 00 |

|  |  |
| --- | --- |
| Applet name | AID |
| Api\_2\_RMm\_Rrp\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 27 01 01 02 |
| Api\_2\_RMm\_Rrp\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 27 01 02 02 |
| Api\_2\_RMm\_Rrp\_3.java | A0 00 00 00 09 00 05 FF FF FF FF 89 27 01 03 02 |
| Api\_2\_RMm\_Srx\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 27 02 01 02 |
| Api\_2\_RMm\_Srx\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 27 02 02 02 |
| Api\_2\_RMm\_Srx\_3.java | A0 00 00 00 09 00 05 FF FF FF FF 89 27 02 03 02 |
| Api\_2\_RMm\_Sgp\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 27 03 01 02 |
| Api\_2\_RMm\_Sgp\_2.java | A0 00 00 00 09 00 05 FF FF FF FF 89 27 03 02 02 |
| Api\_2\_RMm\_Sgp\_3.java | A0 00 00 00 09 00 05 FF FF FF FF 89 27 03 03 02 |

### B.2.3.2 Interface ReaderListener

|  |  |
| --- | --- |
| Package name | AID |
| uicc.test.hci.services.reader.Api\_2\_RMl\_Ocb | A0 00 00 00 09 00 05 FF FF FF FF 89 27 04 00 00 |

|  |  |
| --- | --- |
| Applet name | AID |
| Api\_2\_Rml\_Ocb\_1.java | A0 00 00 00 09 00 05 FF FF FF FF 89 27 04 01 02 |
|  |  |

### B.2.3.3 Interface ReaderService

FFS

Annex C (normative):  
Requirements

# C.1 Non-occurrence and out-of-scope requirements

## C.1.1 Void



## C.1.2 ETSI TS 102 705 prose part

|  |  |
| --- | --- |
| Requirement | Reference |
| The Contactless Framework shall bind the services defined in the uicc.hci.services.cardemulation package to the underlying HCI resources (e.g. gates and pipes) defined by the HCI protocol as specified in [] | 4.2 |

# C.2 FFS requirements

## C.2.1 Void



## C.2.2 ETSI TS 102 705 prose part

|  |  |
| --- | --- |
| Requirement | Reference |
| When the contactless interface is disabled (cf. "state of contactless functionality" in ETSI TS 102 223 [] and setCommunicationInterface() API method of "GlobalPlatform Amendment C" []), the Contactless Framework shall throw an HCIException with reason code HCI\_CURRENTLY\_DISABLED | 4.1 | |
| In case of a communication error on the RF interface (i.e. the RF error indicator is set), messages are not propagated to the application layer in CardEmulation Mode. | 4.2 | |
| In case of a communication error on the RF interface (i.e. the RF error indicator is set), messages are propagated to the application layer in ReaderMode. | 4.3 | |
| The Contactless Framework shall request the reader mode control on the CLF by sending the HCI events EVT\_READER\_REQUESTED and EVT\_END\_OPERATION according to the state of the reader mode Applet | 4.3 | |
| The Contactless Framework shall resend the EVT\_READER\_REQUESTED to the CLF if another Applet instance exists with the ReaderListener.EVENT\_TARGET\_DISCOVERED event activated | 4.3 | |
| The EVT\_READER\_REQUESTED shall be sent by the Contactless Framework if an Applet instance activates the event ReaderListener.EVENT\_TARGET\_DISCOVERED and no other Applet instance has the event activated, i.e. it shall not be sent if the Contactless Framework has earlier sent an EVT\_READER\_REQUESTED due to the request from another Applet instance, which was not yet ended by an EVT\_END\_OPERATION | 4.3 | |
| The Contactless Framework shall ensure that the ReaderListener.EVENT\_TARGET\_DISCOVERED is deactivated for all Applets when access to the interface is disabled on the UICC level | 4.3 | |
| When an Applet lifecycle state changes from ACTIVATED to DEACTIVATED the Contactless Framework shall enforce that the ReaderListener.EVENT\_TARGET\_DISCOVERED is deactivated | 4.3 | |
| The HCI event EVT\_END\_OPERATION shall be sent to the CLF when an Applet instance or the Contactless Framework deactivates the event ReaderListener.EVENT\_TARGET\_DISCOVERED | 4.3 | |
| The Contactless Framework shall inform the Applet instance which has activated the  ReaderListener.EVENT\_TARGET\_DISCOVERED when a target is discovered on one of the RF technologies the Applet instance is registered to with its installation parameters as specified in ETSI TS 102 226 [] | 4.3 | |
| Reader mode Applets shall follow the extended lifecycle model that is defined in "GlobalPlatform Amendment C" [] for contactless Applets in card emulation mode (i.e. following Application Availability States and the related transition rules) | 4.3 | |
| Per RF technology there shall be only one reader mode Applet in the state ACTIVATED (according to "GlobalPlatform Amendment C" []) at any time | 4.3 | |
| When the state of a reader mode Applet changes to lifecycle ACTIVATED (according to "GlobalPlatform  Amendment C" []) the Contactless Framework shall ensure that the HCI gates and pipes are setup for the RF technologies that are supported by the reader mode Applet | 4.3 | |

Annex D (normative):  
Test Specification for Java Card™ Platform HCI API for the UICC

The source files for the HCI API for the UICC Application Programming Interface for Java Card™ for contactless Applets are contained in Annex\_D\_TestAppletsSourceCode.zip, which accompanies the present document.

Annex E (normative):  
Void

Annex F (informative):  
Void

Annex G (informative):  
Core specification version information

Unless otherwise specified, the versions of ETSI TS 102 705 [] from which conformance requirements have been extracted are as follows.

| Release | Latest version from which conformance requirements have been extracted |
| --- | --- |
| 9 | V9.3.0 + CR043 (SCP(14)000285) |

Annex H (informative):  
Change history

The table below indicates all changes that have been incorporated into the present document since it was placed under change control.

| Change history | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Date | Meeting | Plenary Doc | CR | Rev | Cat | Subject/Comment | Old | New |
|  |  | SCP(13)000037 |  |  |  | Creation of the specification |  | 9.0.0 |
| 2013-02 | SCP#58 | SCP(13)000038 | 001 | - | F | Added definition of test case IDs | 9.0.0 | 9.1.0 |
| SCP(13)000039 | 002 | - | F | Clarification of initial conditions activity performed by UICC | 9.0.0 | 9.1.0 |
| SCP(13)000040 | 003 | - | F | Test case 6.1.5.1: corrected definition of unrecognised Envelope | 9.0.0 | 9.1.0 |
| SCP(13)000041 | 004 | - | F | Correction of expected HCI event for TC 6.2.2.1.2.4 and 6.2.2.1.3.4 ID2-1 and ID2-2 | 9.0.0 | 9.1.0 |
| SCP(13)000042 | 005 | - | F | Clarification of prepareAndSendGetParameterCommand test cases | 9.0.0 | 9.1.0 |
| SCP(13)000043 | 006 | - | F | Correction of test suite files | 9.0.0 | 9.1.0 |
| SCP(13)000044 | 007 | - | F | Correction of wrong Events deactivation | 9.0.0 | 9.1.0 |
| SCP(13)000046 | 008 | - | F | Clarification of initial conditions related to power mode | 9.0.0 | 9.1.0 |
| SCP(13)000085r1 | 010 | - | F | Modification and addition of test applets to test HCI is disabled condition | 9.0.0 | 9.1.0 |
| 2013-04 | SCP#59 | SCP(13)000086 | 011 | 1 | F | Addition of references to GlobalPlatform APIs. | 9.0.0 | 9.1.0 |
| SCP(13)000087r1 | 012 | - | F | Clarification of EVT\_READER\_REQUESTED transmission | 9.0.0 | 9.1.0 |
| SCP(13)000088 | 013 | 1 | F | Correction of applet selection in TC 6.2.1.1.1 ID3 | 9.0.0 | 9.1.0 |
| SCP(13)000089 | 014 | - | F | Correction of API expected received data in TC 6.1.3.7 | 9.0.0 | 9.1.0 |
| SCP(13)000090 | 015 | - | F | Correction of applicability for test cases 6.1.1.1 (ID3-2) 6.1.1.2 and 6.1.2.3 (ID5) | 9.0.0 | 9.1.0 |
| SCP(13)000091 | 016 | - | F | Modification of test procedure 6.1.1.3.4 ID3-1 and 6.2.2.1.1.4 | 9.0.0 | 9.1.0 |
| SCP(13)000092 | 017 | - | F | Addition of Annexes E and F contents into ETSI TS 103 115 | 9.0.0 | 9.1.0 |
| SCP(13)000093 | 018 | - | F | Corrections to avoid failed deletions of test applets | 9.0.0 | 9.1.0 |
| SCP(13)000037 | 019 | - | F | Corrections on Reader Mode applets | 9.0.0 | 9.1.0 |
| 2013-07 | SCP#60 | SCP(13)000045r1 | 009 | 1 |  | Usage of Annex F is made fully optional | 9.1.0 | 9.2.0 |
| SCP(13)000188 | 021 | - |  | Deletion of getResponse() method from Annex F | 9.1.0 | 9.2.0 |
| SCP(13)000140r1 | 022 | 1 | F | Corrections on Reader Mode test cases and proposal of Test Case 4 from 6.2.3.1.3.4 for FFS | 9.1.0 | 9.2.0 |
| SCP(13)000141r1 | 023 | 1 | F | Correction of 6.2.2.1.2 method prepareAndSendTransactionEvent() test case ID5 | 9.1.0 | 9.2.0 |
| SCP(13)000138 | 020 | - | F | Annex E update with Reader Mode features used in Annex F | 9.1.0 | 9.2.0 |
| 2013-10 | SCP#61 | SCP(13)000217 | 024 | - | F | Correction of applet selection issues | 9.2.0 | 9.3.0 |
| SCP(13)000218 | 025 | - | F | Corrections in applicability table | 9.2.0 | 9.3.0 |
| SCP(13)000220 | 026 | - | F | Correction of reader mode test cases regarding applet selection and usage of ISO interface | 9.2.0 | 9.3.0 |
| SCP(13)000222 | 027 | - | F | Correction of TC 6.1.3.3.4 ID2 | 9.2.0 | 9.3.0 |
| SCP(13)000219 | 028 | - | F | Test cases 6.1.1.1/5: correction of execution requirements | 9.2.0 | 9.3.0 |
| SCP(13)000221 | 029 | - | F | Modification of Terminal profile in the initial conditions for some test cases | 9.2.0 | 9.3.0 |
| SCP(13)000223 | 030 | - | F | Removal of redundant entries in applicability table | 9.2.0 | 9.3.0 |
| 2014-02 | SCP#62 | SCP(14)000019r1 | 031 | 1 | F | Test case 6.1.5.1 ID2: specification of Terminal Profile and response to ENVELOPE command | 9.3.0 | 9.4.0 |
| SCP(14)000020r1 | 032 | 1 | F | Correction of some test cases to add SAA1 | 9.3.0 | 9.4.0 |
| SCP(14)000021r1 | 033 | 1 | F | Corrections on AID coding section | 9.3.0 | 9.4.0 |
| SCP(14)000022r1 | 034 | 1 | F | Test case 6.1.1.1 ID5-2: correction of execution requirements | 9.3.0 | 9.4.0 |
| SCP(14)000023r1 | 035 | 1 | F | Test case 6.1.2.1 ID2: correction of execution requirements | 9.3.0 | 9.4.0 |
| SCP(14)000024r1 | 036 | 1 | F | Test case 6.1.1.3 ID 5-2: correction of expected status word | 9.3.0 | 9.4.0 |
| SCP(14)000025r1 | 037 | 1 | F | Connectivity test cases: correction related to ordering of events | 9.3.0 | 9.4.0 |
| SCP(14)000026r1 | 038 | 1 | F | Connectivity test cases involving wrong precondition: correction of initial conditions | 9.3.0 | 9.4.0 |
| SCP(14)000027r1 | 039 | 1 | F | Correction of uninstall() test cases to implement AppletEvent interface | 9.3.0 | 9.4.0 |
| SCP(14)000029r1 | 041 | 1 | F | Test case 6.1.1.3 ID5-1/2: HCI selection removed | 9.3.0 | 9.4.0 |
| SCP(14)000030r1 | 043 | 1 | D | Alignment of enable/disable contactless interface terminology with ETSI TS 102 705 | 9.3.0 | 9.4.0 |
| 2014-04 | SCP#63 | SCP(14)000119r1 | 040 | 2 | F | Correction and clarification of initial conditions and test cases. | 9.3.0 | 9.4.0 |
| SCP(14)000114r1 | 044 | 1 | F | Corrections on CRRN3 of onCallback method (Card Emulation) | 9.3.0 | 9.4.0 |
| SCP(14)000115r1 | 045 | 1 | F | Corrections on test case Id1 of onCallback method (Card Emulation) | 9.3.0 | 9.4.0 |
| SCP(14)000116 | 046 | - | F | Addition of AID for Api\_1\_Hsr\_Ace\_3 | 9.3.0 | 9.4.0 |
| SCP(14)000117r1 | 047 | 1 | D | Corrections on the wording in test cases 6.2.3.1.2.4 and 6.2.3.2.1.4 | 9.3.0 | 9.4.0 |
| SCP(14)000118 | 048 | - | F | Reader mode test cases: corrections related to ISO APDUs | 9.3.0 | 9.4.0 |
| 2014-06 | SCP#64 | SCP(14)000158 | 049 | - | F | prepareAndSendWriteXchgDataCommand test case: improvement of test case coverage | 9.3.0 | 9.4.0 |
| SCP(14)000159 | 050 | - | F | Clarification on test cases ID3 and ID6 of onCallback method (Card Emulation) | 9.3.0 | 9.4.0 |
| SCP(14)000161 | 051 | - |  | Unknown\_Power\_Mode | 9.3.0 | 9.4.0 |
| SCP(14)000160 | 052 | - | F | RF\_Transmission\_Errror | 9.3.0 | 9.4.0 |
| 2014-12 | SCP#66 | SCP(14)000316 | 053 | - | F | Test case 6.1.2.3: correction of execution requirements and applicability | 9.3.0 | 9.4.0 |
| 2015-02 | SCP#67 | SCP(15)000027 | 055 | - | F | Removal of Annexes E and F | 9.3.0 | 9.4.0 |
| SCP(15)000028r1 | 056 | 1 | F | Correction of requirements CRRN4 and CRRN3 in 6.2.2.1.1/2 | 9.3.0 | 9.4.0 |
| SCP(15)000029 | 057 | - | B | Connectivity test cases: addition of reader mode scenarios | 9.3.0 | 9.4.0 |
| SCP(15)000030 | 058 | - |  | Correction of Api\_2\_CEl\_Ocb\_1 (used in test case 6.2.1.3.1) | 9.3.0 | 9.4.0 |
| 2015-04 | SCP#68 | SCP(15)000103 | 060 | - | D | Consolidation of Annex C and non-tested requirements | 9.4.0 | 9.5.0 |
| 2015-04 | SCP#68 | SCP(15)000104 | 061 | - | F | Applets Api\_2\_RMm\_Sgp\_2, Api\_2\_RMm\_Srx\_2: correction of int to byte | 9.4.0 | 9.5.0 |
| 2015-04 | SCP#68 | SCP(15)000105 | 062 | - | F | Test cases 6.1.3.4/ID2, 6.1.5.1/ID[4,5]: corrected location of ANY\_GET\_PARAMETER | 9.4.0 | 9.5.0 |
| 2015-04 | SCP#68 | SCP(15)000130 | 063 | - | F | Removal of ReaderMessage test cases for HCI disabled using TS 102 223 | 9.4.0 | 9.5.0 |
| 2015-04 | SCP#68 | SCP(15)000106 | 064 | - | F | Inclusion of changes in CR043 for TS 102 705 | 9.4.0 | 9.5.0 |

# History

|  |  |  |
| --- | --- | --- |
| **Document history** | | |
| V9.0.0 | April 2013 | Publication |
| V9.1.0 | July 2013 | Publication |
| V9.2.0 | October 2013 | Publication |
| V9.3.0 | December 2013 | Publication |
| V9.4.0 | April 2015 | Publication |