



**Smart Cards;  
Test specification for UICC Application Programming  
Interface for Java Card™ for Contactless Applications;  
Test Environment and Annexes  
(Release 9)**

---

**Reference**DTS/SCP-00HCL\_API\_TESTv910

---

**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**

---

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the 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:

[http://portal.etsi.org/chaicor/ETSI\\_support.asp](http://portal.etsi.org/chaicor/ETSI_support.asp)

---

**Copyright Notification**

---

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2013.  
All rights reserved.

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

# Contents

Intellectual Property Rights .....	7
Foreword.....	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 .....	21
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.....	27
6.1.2.1.3	Initial conditions.....	27
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.....	29
6.1.2.2.4	Test procedure.....	29
6.1.2.3	Method activateEvent.....	29
6.1.2.3.1	Conformance requirements.....	29
6.1.2.3.2	Test Suite Files.....	31
6.1.2.3.3	Initial conditions.....	31
6.1.2.3.4	Test procedure.....	31
6.1.2.4	Method deactivateEvent.....	34
6.1.2.4.1	Conformance requirements.....	34
6.1.2.4.2	Test suite files.....	35
6.1.2.4.3	Initial conditions.....	35
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.....	39
6.1.2.5.4	Test procedure.....	39
6.1.2.6	Method getEventNotificationStatus.....	39
6.1.2.6.1	Conformance requirements.....	39
6.1.2.6.2	Test Suite Files.....	40
6.1.2.6.3	Initial conditions.....	40
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.....	44
6.1.3.1.3	Initial conditions.....	44
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.....	45
6.1.3.2.3	Initial condition.....	45
6.1.3.2.4	Test procedure.....	45
6.1.3.3	Method getType.....	45
6.1.3.3.1	Conformance requirements.....	45
6.1.3.3.2	Test Suite Files.....	46
6.1.3.3.3	Initial condition.....	46
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.....	47
6.1.3.4.3	initial condition.....	47
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.....	48
6.1.3.6.1	Conformance requirements.....	48
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.....	50
6.1.4.1.3	Initial conditions.....	50
6.1.4.1.4	Test procedure .....	50
6.1.5	Interface HCIListener .....	51
6.1.5.1	Method onCallback .....	51
6.1.5.1.1	Conformance requirements.....	51
6.1.5.1.2	Test Suite Files .....	52
6.1.5.1.3	Initial conditions.....	52
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 getCardRFTType .....	59
6.2.1.2.2	Test Suite Files .....	59
6.2.1.2.3	Initial conditions.....	60
6.2.1.3	Interface CardEmulationListener .....	60
6.2.1.3.1	Method onCallback .....	60
6.2.2	Package Connectivity Service.....	64
6.2.2.1	Interface ConnectivityService .....	64
6.2.2.1.1	Method prepareAndSendConnectivityEvent .....	64
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).....	70
6.2.3	Package Reader Service.....	73
6.2.3.1	Interface ReaderMessage .....	73
6.2.3.1.1	Method restartReaderModeProcedure .....	73
6.2.3.1.2	Method prepareAndSendWriteXchgDataCommand .....	75
6.2.3.1.3	Method prepareAndSendGetParameterCommand .....	78
6.2.3.2	Interface ReaderListener .....	82
6.2.3.2.1	Method onCallback .....	82
<b>Annex A (normative):</b>	<b>Class, methods and tests acronyms .....</b>	<b>85</b>
A.1	HCI framework .....	85
A.1.1	Class HCIDevice .....	85
A.1.2	Interface HCIService.....	85
A.1.3	Interface HCIMessage.....	85
A.1.4	Interface HCIListener.....	85
A.1.5	Class HCIException .....	86
A.2	HCI Services .....	86
A.2.1	Package cardemulation.....	86
A.2.1.1	Interface CardEmulationListener.....	86
A.2.1.2	Interface CardEmulationMessage.....	86
A.2.1.3	Interface CardEmulationService.....	86
A.2.2	Package connectivity .....	86
A.2.2.1	Interface ConnectivityListener.....	86
A.2.2.2	Interface ConnectivityMessage.....	86
A.2.2.3	Interface ConnectivityService.....	87
A.2.3	Readermode.....	87
A.2.3.1	Interface RaederListener.....	87
A.2.3.2	Interface ReaderMessage .....	87
A.2.3.3	Interface ReaderService .....	87

<b>Annex B (normative):</b>	<b>AIDs - to be reserved .....</b>	<b>88</b>
B.1	Package HCI framework .....	88
B.1.1	Class HCIDevice .....	88
B.1.2	Interface HCIService .....	88
B.1.3	Interface HCIMessage .....	89
B.1.4	Interface HCIListener .....	89
B.1.5	Class HCIException .....	89
B.2	HCI Services .....	90
B.2.1	Package cardemulation .....	90
B.2.1.1	Interface CardEmulationListener .....	90
B.2.1.2	Interface CardEmulationMessage .....	90
B.2.1.3	Interface CardEmulationService .....	91
B.2.2	Package connectivity .....	91
B.2.2.1	Interface ConnectivityListener .....	91
B.2.2.2	Interface ConnectivityMessage .....	91
B.2.2.3	Interface ConnectivityService .....	91
B.2.3	Package readermode .....	91
B.2.3.1	Interface ReaderMessage .....	92
B.2.3.2	Interface ReaderListener .....	92
B.2.3.3	Interface ReaderService .....	92
<b>Annex C (normative):</b>	<b>Requirements .....</b>	<b>93</b>
C.1	Non-occurrence and out-of-scope requirements .....	93
C.1.1	Package uicc.hci .....	93
C.1.2	TS 102 705 prose part .....	93
C.2	FFS requirements .....	93
C.2.1	Package uicc.hci .....	93
C.2.2	TS 102 705 prose part .....	94
<b>Annex D (normative):</b>	<b>Test Specification for Java Card™ Platform HCI API for the UICC .....</b>	<b>95</b>
<b>Annex E (normative):</b>	<b>uicc.contactless.test.util package and contactless interfaces .....</b>	<b>96</b>
<b>Annex F (normative):</b>	<b>Test Area files .....</b>	<b>97</b>
<b>Annex G (informative):</b>	<b>Core specification version information .....</b>	<b>98</b>
<b>Annex H (informative):</b>	<b>Change history .....</b>	<b>99</b>
History .....		100

---

## 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>).

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.

---

# 1 Scope

The present document covers the minimum characteristics considered necessary in order to provide compliance to TS 102 705 [1].

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

---

# 2 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.

## 2.1 Normative references

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

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 TS 102 705 [1]

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

**HCP message:** message as specified in TS 102 622 [3]

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 (TS 102 622 [3]) protocol specification

## 3.2 Symbols

For the purposes of the present document, the symbols given in TS 102 705 [1] 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 [2].

API            Application Programming Interface  
 CAT           Card Application Toolkit  
 CB            Chaining Bit  
 CLF           Contactless Front-end

NOTE:    According to TS 102 622 [3].

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  
 DUT           Device Under Test  
 EVT           EVenT  
 FFS           For Further Study  
 GND           GrouND  
 HCI           Host Controller Interface

NOTE:    According to TS 102 622 [3].

HCP           Host Controller Protocol

NOTE:    According to TS 102 622 [3].

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  
 TBD           To Be Defined

## 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 [8], 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 [8], 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 a 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 a 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	M	
6.1.1.1	Method getHCIService (ID5-2)	Rel-9	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, ID2, ID3, ID4)	Rel-9	SAA1	M	
6.1.2.1	Method register (ID1, ID2, ID3-1, ID4)	Rel-9	SAA1	M	
6.1.2.1	Method register (ID1-2, 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	

Clause	Test case number and description	Release	Execution requirements	Rel-9 UICC	Support
6.1.2.3	Method activateEvent (ID3)	Rel-9	SAA1, SAA2	M	
6.1.2.3	Method activateEvent (ID5-1)	Rel-9	SAA10	SAA10, SAA3	
6.1.2.3	Method activateEvent (ID5-2)	Rel-9	SAA11	SAA11, SAA3	
6.1.2.4	Method deactivateEvent (ID1, ID4, ID5)	Rel-9	SAA1	M	
6.1.2.4	Method deactivateEvent (ID2, ID7)	Rel-9	SAA3	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, ID3)	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, ID5-2)	Rel-9	SAA1	C001	
6.2.1.3.1	Method onCallback (ID4, ID5, ID6, ID5-3)	Rel-9	SAA1	C002	
6.2.1.3.1	Method onCallback (ID5-1, ID5-4)	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, SAA1, SAA10	M	
6.2.2.1.1	Method prepareAndSendConnectivityEvent (ID2-1)	Rel-9	SAA2, SAA1, SAA11	M	
6.2.2.1.1	Method prepareAndSendConnectivityEvent (ID6)	Rel-9	SAA2, SAA1, SAA9	M	
6.2.2.1.2	Method prepareAndSendTransactionEvent byte[] aid, short aidOffset, short aidLen, byte[] parameters, short parametersOffset, short parametersLen) (ID1, ID3, ID4, ID5)	Rel-9	SAA1, SAA2	M	

Clause	Test case number and description	Release	Execution requirements	Rel-9 UICC	Support
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	
6.2.2.1.3	Method prepareAndSendTransactionEvent (byte[] parameters, short parametersOffset, short parametersLen ) (ID1, ID3, ID4)	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	
6.2.3.1.1	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 (ID5-1)	Rel-9	SAA3, SAA10	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 (ID3-1)	Rel-9	SAA3, SAA10	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 (ID3-1)	Rel-9	SAA3, SAA10	M	
6.2.3.1.3	Method prepareAndSendGetParameterCommand (ID3-2)	Rel-9	SAA3, SAA11	M	
6.2.3.1.3	Method prepareAndSendGetParameterCommand (ID4)	Rel-9	SAA3	C007	
6.2.3.1.3	Method prepareAndSendGetParameterCommand (ID5)	Rel-9	SAA3	M	
6.2.3.2.1	Method onCallback	Rel-9	SAA3	M	
6.2.3.2.1	Method onCallback (ID1, ID2, ID3, ID4)	Rel-9	SAA3	C005	
6.2.3.2.1	Method onCallback (ID5, ID6, ID7, 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 TS 102 241 [6] 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 (SAA<sub>n</sub>) 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 (SAA<sub>n</sub>), 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:

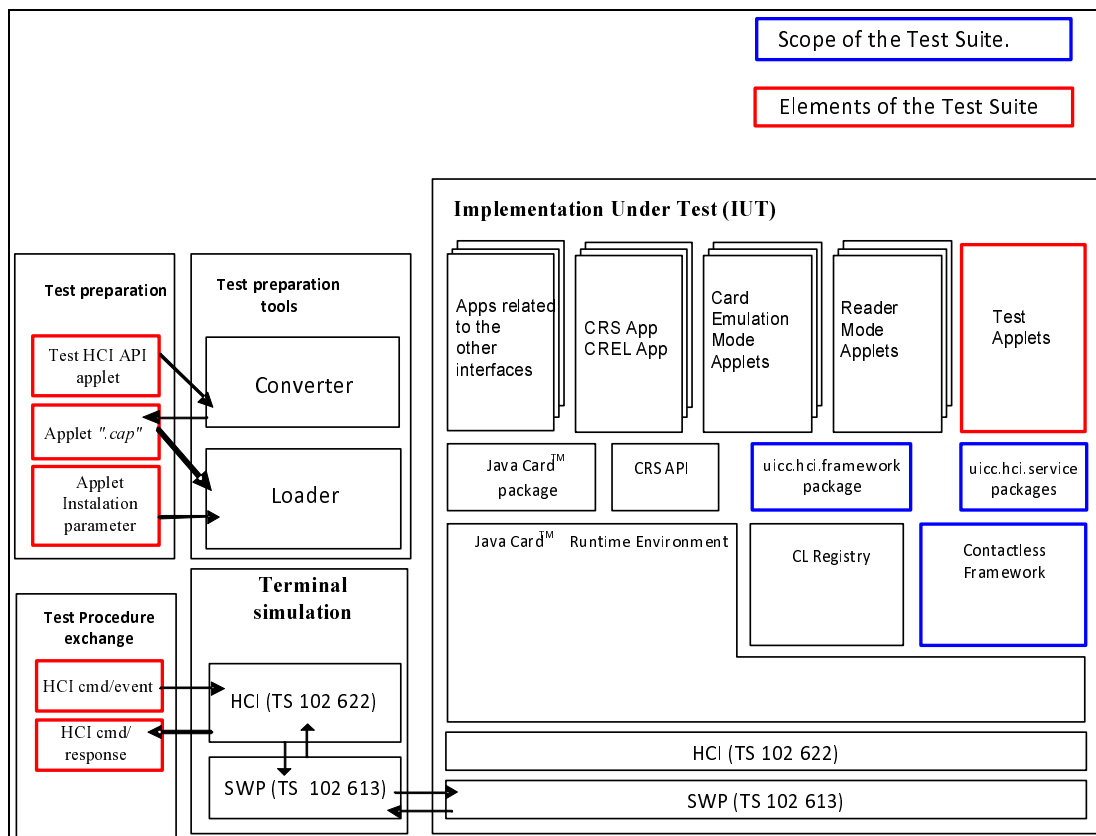


Figure 5.1

## 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 TS 102 705 [1].
- 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 [13].

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

### 5.2.1.3 Initial conditions

In addition to the general preconditions 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
	<i>Commands sent on HCI interface</i>	<i>Test Case detailed description</i>	<i>API expected behaviour</i>	<i>Expected behaviour at APDU/HCI level</i>	<i>Conformance Requirements Reference</i>

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 TS 102 622 [3].
- 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 TS 102 622 [3], 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 TS 102 705 [1] uicc.hci.framework package.

**uicc.hci.test.services.cardemulation.[Test Area Reference]:** Java Card packages containing Test Area References for the TS 102 705 [1] uicc.hci.services.cardemulation package.

**uicc.hci.test.services.connectivity.[Test Area Reference]:** Java Card packages containing Test Area References for the TS 102 705 [1] uicc.hci.services.connectivity package.

**uicc.hci.test.services.readermode.[Test Area Reference]:** Java Card packages containing Test Area References for the TS 102 705 [1] uicc.hci.services.readermode package.

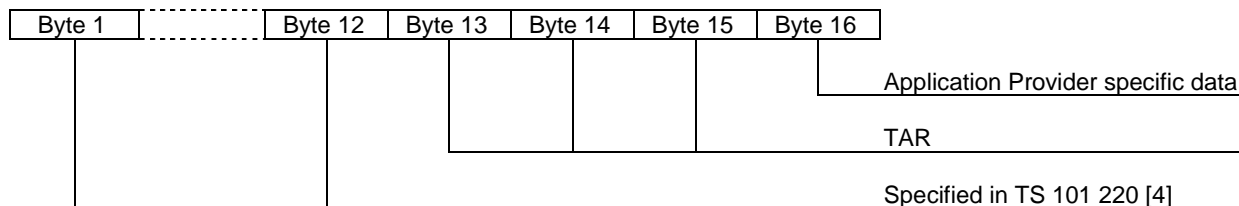
where the Test Area Reference is written in lower case.

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]`.

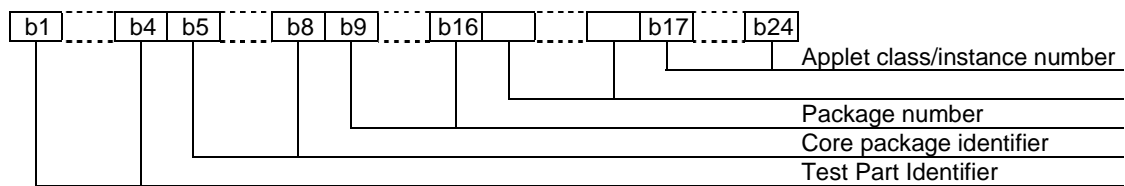
### 5.5 AID coding

The AID coding for the Test Packages, Applet classes and Applets shall be as specified in TS 101 220 [4]. In addition, the following TAR and Application Provider specific data values are defined for use within the present document:

AID coding



TAR coding (3 bytes / 24 bits):



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 0100 uicc.hci.test.servces.connectivity.
- 0010 0100 uicc.hci.test.servces.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 TS 102 622 [3];
- be able to send and receive the commands correctly on the lower layer; i.e TS 102 613 [14];
- 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 ([11], [12],[13]) 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
<b>1 Select supported and registered service</b>					
1	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; the initial conditions in clause 6.1.1.1.3 not applicable here - select applet-INS= '03'	getHCIService() serviceID = READER_SERVICE_ID	No exception shall be thrown	SW - '90 00'	N1, N2
<b>2 Applet not registered</b>					
2	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
<b>3 Select not available service</b>					
3	1 - Send command on ISO interface; the initial conditions in clause 6.1.1.1.3 not applicable here - select applet- (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; the initial conditions in clause 6.1.1.1.3 not applicable here - select applet- (INS = '03')	getHCIService() serviceID = ReaderService	Shall throw uicc.hci.framework.HCIException with error code HCI_SERVICE_NOT_AVAILABLE	SW - '90 01'	C2
<b>4 Select undefined service</b>					
4	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

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
5	<b>Access not granted</b>				
	1 - Send command on ISO interface; the initial conditions in clause 6.1.1.1.3 not applicable here - select applet- INS = '01'	getHCIService() serviceID = CardEmulationService	Shall throw uicc.hci.framework.HCIException with error code HCI_ACCESS_NOT_GRANTED	SW - '90 02'	C1
	2 - Send command on ISO interface; the initial conditions in clause 6.1.1.1.3 not applicable here - select applet- 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 TS 102 613 [14].
- CRRN2: returns the current power mode or -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 TS 102 705 [1] which do not require a CardEmulationListener registration (e.g. requesting the power mode or connectivity service).

##### 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	<b>Verify power mode</b>				
	- 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	<b>Verify power mode</b>				
	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	<b>SWP interface is not activated</b>				
	sending command on ISO interface: - select applet- INS='05'	getPowerMode()	Return '-1'	SW - '90 05'	N2

## 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	<b>Retrieve information of an available service</b>				
	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	<b>Interface not supported by terminal</b>				
	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



Test Case					
ID	HCI/APDU Command	API Description	API Expectation	HCI/APDU Response	CRR
3	<b>Service not supported by terminal</b>				
	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	<b>Access not allowed for Applet</b>				
	1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.1.1.3 not applicable here - select applet - 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 - select applet - send APDU (INS = '08')	isHCIServiceAvailable() serviceID = READER_SERVICE_ID	bit 2 set	SW - '90 00', bit 2 set in the data byte	N2
5	<b>HCI interface currently disabled in UICC</b>				
	1 - Precondition: Contactless functionality state is disabled in the UICC as defined in TS 102 223 [7]. Send on ISO interface the following commands: - Send APDU to select the applet. - Send APDU (INS = '06')	isHCIServiceAvailable()	bit 3 set	SW - '90 00', bit 3 set in the data byte	N2

Test Case					
ID	HCI/APDU Command	API Description	API Expectation	HCI/APDU Response	CRR
	2 - Precondition: Contactless functionality state 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 = '06') - Postcondition: Contactless functionality state is enabled again in the UICC as defined in Global Platform Amendment C	<pre>                     setCommunicationInterface()                     API method of Global Platform Amendment C [10]                     is used to disable HCI interface                      isHCIServiceAvailable()                      setCommunicationInterface()                     API method of Global Platform Amendment C [10]                     is used to enable again HCI interface                 </pre>	bit 3 set	SW - '90 04', 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

ID	HCI Command	Test case			
		Description	API Expectation	HCI Response	CRR
1	<b>Register Listener to a service</b>				
	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	<b>Register wrong listener type</b>				
	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	<b>Listener already registered</b>				
	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

ID	HCI Command	Description	Test case		
			API Expectation	HCI Response	CRR
	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
<b>Persistent Listener Registration</b>					
4	- 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 TS 102 705 [1] 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
<b>Deregister Listener to a Service - CardEmulation</b>					
1	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
<b>Deregister Listener to a Service - Connectivity</b>					
2	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.
- CRRN3: for all service interfaces HCIListener.EVENT\_HCI\_RECEPTIONS\_FAILED is sent to notify the applet when the UICC failed to receive a message sent by CLF.
- 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 TS 102 223 [7] and setCommunicationInterface() API method of "GlobalPlatform Amendment C" [10]), 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 TS 102 221 [5] 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" [11] and in "GlobalPlatform Amendment C" [10].
- 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" [11].
- CRRN14: If the HCI event EVT\_FIELD\_OFF or EVT\_CARD\_DEACTIVATED defined by the HCI protocol as specified in TS 101 220 [4] is received by the Contactless Framework and the UICC is still powered, the Applet instance shall be deselected according to "GlobalPlatform Amendment C" [10].

#### 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.

#### 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	<b>Event activation - CardEmulationService</b>				
1	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	2 - EVT_SEND_DATA(INS = '11')	HCIListener = CardEmulationListener HCIService = CardEmulationService  activateEvent() event = HCIListener.EVENT_HCI_RECEPTIONS_FAILED	No exception shall be thrown	EVT_SEND_DATA (SW- '90 00')	N1, N3
3	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	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

Test Case					
ID	HCI Command	Description	API Expectation	HCI Response	CRR
2	<b>Event activation - ReaderService</b>				
	1 - EVT_SEND_DATA (INS = '11')	HCIService = ReaderService activateEvent() event = HCIListener.EVENT_HCI_RECEPTIONS_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	<b>Event activation - ConnectivityListener</b>				
	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_RECEPTIONS_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	<b>Wrong event type</b>				
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	



Test Case					
ID	HCI Command	Description	API Expectation	HCI Response	CRR
5	<b>HCI is disabled</b>				
	1- Precondition: Contactless functionality state is disabled in the UICC as defined in TS 102 223 [7]. Send on ISO interface the following commands - Send APDU to select the applet. - Send APDU (INS = '01') on ISO interface	HCIService = ReaderService activateEvent() event = ReaderListener.EVENT_TARGET_D ISCOVERED	HCIException with reason code HCI_CURRENTLY_DISABLED shall be thrown	SW- '90 00'	P2, N8
	2 - Precondition: Contactless functionality state 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 = '01') on ISO interface - Postcondition: Contactless functionality state is enabled again in the UICC as defined in Global Platform Amendment C	setCommunicationInterface() API method of "GlobalPlatform Amendment C" [10] used to disable HCI interface HCIService = ReaderService activateEvent() event = ReaderListener.EVENT_TARGET_D ISCOVERED  setCommunicationInterface() API method of Global Platform Amendment C [10] is used to enable again HCI interface	HCIException with reason code HCI_CURRENTLY_DISABLED shall be thrown	SW- '90 00'	P2, N8
6	<b>Persistent Event Activation</b>				
	- 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	<b>Event Field Off before deselect</b>				
	- 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	<b>No triggering after Event Field Off</b>				
	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	<b>Check for Selection</b>				
	Send EVT_SEND_DATA (INS = '01')	Service = CardEmulationService Listener = CardEmulationListener	Check for invocation of select()	EVT_SEND_DATA (SW - '90 00')	N12, N13

Test Case					
ID	HCI Command	Description	API Expectation	HCI Response	CRR
10	<b>No triggering after Event Card Deactivate</b>				
	EVT_CARD_DEACTIVATE - Send EVT_SEND_DATA (INS = '01')	Service = CardEmulationService Listener = CardEmulationListener activateEvent(EVENT_ON_SEND_D ATA) 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.
- CRRN3: for all service interfaces HCIListenr HCIListener.EVENT\_HCI\_RECEPTIONS\_FAILED is sent to notify the applet when the UICC failed to receive a message sent by CLF.
- 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.

##### 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	<b>Event deactivation - CardEmulationService</b>				
	1 - EVT_SEND_DATA(INS='10')	HCIService = CardEmulationService Event has been successfully activated  deactivateEvent() event = HCIListener.EVENT_HCI_TRA NSMISSION_FAILED	No exception shall be thrown	EVT_SEND_DAT A (SW- '90 00')	N1, N2
	2 - EVT_SEND_DATA(INS='11')	HCIService = CardEmulationService Event has been successfully activated  deactivateEvent() event = HCIListener.EVENT_HCI_REC EPTIONS_FAILED	No exception shall be thrown	EVT_SEND_DAT A (SW- '90 00')	N1, N3
	3 - EVT_SEND_DATA(INS='12')	HCIService = CardEmulationService Event has been successfully activated  deactivateEvent() event = CardEmulationListener.EVE NT_GET_PARAMETER_RESPONSE	No exception shall be thrown	EVT_SEND_DAT A (SW- '90 00')	N1, N4
	4 - EVT_SEND_DATA(INS='14')	HCIService = CardEmulationService Event has been successfully activated  deactivateEvent() event = CardEmulationListener.EVE NT_ON_SEND_DATA	No exception shall be thrown	EVT_SEND_DAT A (SW- '90 00')	N1, N4

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
2	<b>Event deactivation - ReaderService</b>				
	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	<b>Event deactivation - ConnectivityService</b>				
	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_RECEPTIONS_FAILED	No exception shall be thrown	EVT_SEND_DATA (SW- '90 00')	N1, N3

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
	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	<b>Wrong event type</b>				
	EVT_SEND_DATA (INS = '16')	HCIService = CardEmulationService Event has been successfully activated  deactivateEvent() event value = 02	HCIException with reason code HCI_WRONG_EV ENT_TYPE shall be thrown	EVT_SEND_DATA (SW- '90 00')	P1
5	<b>Deactivate an non-active event - CardEmulationService</b>				
	1 - EVT_SEND_DATA (INS = '10')	HCIService = CardEmulationService Event was not activated before deactivateEvent() event = HCIListener.EVENT_HCI_TRA NSMISSION_FAILED	HCIException with reason code HCI_WRONG_EV ENT_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_REC EPTIONS_FAILED	HCIException with reason code HCI_WRONG_EV ENT_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.EVE NT_GET_PARAMETER_RESPONSE	HCIException with reason code HCI_WRONG_EV ENT_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.EVE NT_FIELD_OFF	HCIException with reason code HCI_WRONG_EV ENT_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.EVE NT_ON_SEND_DATA	HCIException with reason code HCI_WRONG_EV ENT_TYPE shall be thrown	EVT_SEND_DATA (SW- '90 00')	P2
6	<b>Deactivate an non-active event - ConnectivityService</b>				
	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_EV ENT_TYPE shall be thrown	EVT_SEND_DATA (SW- '90 00')	P2

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
7	<b>Deactivate an non-active event - ReaderService</b>				
1 -	EVT_SEND_DATA(INS='10')	HCIListener = ReaderListener HCIService = ReaderService Event was not activated before  deactivateEvent() event = EVENT_GET_PARAMETER_RESPONSE	HCIException with reason code HCI_WRONG_EV ENT_TYPE shall be thrown	EVT_SEND_DATA (SW- '90 00')	P2
2 -	EVT_SEND_DATA(INS='11')	HCIListener = ReaderListener HCIService = ReaderService Event was not activated before  deactivateEvent() event = EVENT_WRITE_EXCHANGE_DATA RESPONSE	HCIException with reason code HCI_WRONG_EV ENT_TYPE shall be thrown	EVT_SEND_DATA (SW- '90 00')	P2
3 -	EVT_SEND_DATA(INS='12')	HCIListener = ReaderListener HCIService = ReaderService Event was not activated before  deactivateEvent() event = EVENT_TARGET_DISCOVERED	HCIException with reason code HCI_WRONG_EV ENT_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 trows 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\_RECEPTIONS\_FAILED.

##### 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	<b>Event notification status false - CardEmulationService</b>				
	1 - EVT_SEND_DATA (INS = '10')	HCIService = CardEmulationService  getEventNotificationStatus() event = HCIListener.EVENT_HCI_TRANSM ISSION_FAILED	No exception shall be thrown.  getEventNotificationStatus() = false	EVT_SEND_DAT A (SW- '90 00')	N1, N5
	2 - EVT_SEND_DATA (INS = '11')	HCIService = CardEmulationService  getEventNotificationStatus() event = HCIListener.EVENT_HCI_RECEPT IONS_FAILED	No exception shall be thrown.  getEventNotificationStatus() = false	EVT_SEND_DAT A (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_DAT A (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_DAT A (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_DAT A (SW- '90 00')	N1, N2



Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
2	<b>Event notification status true - CardEmulationService</b>				
	EVT_SEND_DATA (INS = '15')	<pre> HCIService = CardEmulationService activateEvent() event = CardEmulationListener.EVENT_ ON_SEND_DATA  getEventNotificationStatus () event = CardEmulationListener.EVENT_ ON_SEND_DATA </pre>	<p>No exception shall be thrown.</p> <p>getEventNotificationStatus() = true</p>	EVT_SEND_DATA (SW- '90 00')	N1, N2
3	<b>Event notification status true - ReaderService</b>				
	1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.5 not applicable here - Send APDU (INS = '21')	<pre> HCIService = ReaderService activateEvent() event = ReaderListener. EVENT_WRITE_EXCHANGE_DATA_RE SPONSE  getEventNotificationStatus() event = ReaderListener. EVENT_WRITE_EXCHANGE_DATA_RE SPONSE </pre>	<p>No exception shall be thrown.</p> <p>getEventNotificationStatus() = true</p>	SW - '90 00'	N1, N3
	2 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.5 not applicable here - Send APDU (INS = '22')	<pre> HCIService = ReaderService  activateEvent() event = ReaderListener.EVENT_TARGET_ DISCOVERED  getEventNotificationStatus() event = ReaderListener.EVENT_TARGET_ DISCOVERED </pre>	<p>No exception shall be thrown.</p> <p>getEventNotificationStatus() = true</p>	SW - '90 00'	N1, N3
	3 - Send APDU (INS = '23')	<pre> HCIService = ReaderService activateEvent() event = ReaderListener. EVENT_GET_PARAMETER_RESPONSE  getEventNotificationStatus() event = ReaderListener. EVENT_GET_PARAMETER_RESPONSE </pre>	<p>No exception shall be thrown.</p> <p>getEventNotificationStatus() = true</p>	SW - '90 00'	N1, N3
	4 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.5 not applicable here - Send APDU (INS = '24')	<pre> HCIService = ReaderService activateEvent() event = ReaderListener. EVENT_HCI_RECEPTIONS_FAILED  getEventNotificationStatus() event = ReaderListener. EVENT_HCI_RECEPTIONS_FAILED </pre>	<p>No exception shall be thrown.</p> <p>getEventNotificationStatus() = true</p>	SW - '90 00'	N1, N5
	5- Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.5 not applicable here - Send APDU (INS = '25')	<pre> HCIService = ReaderService activateEvent() event = ReaderListener. EVENT_HCI_TRANSMISSION_FAILE D  getEventNotificationStatus() event = ReaderListener. EVENT_HCI_TRANSMISSION_FAILE D </pre>	<p>No exception shall be thrown.</p> <p>getEventNotificationStatus() = true</p>	SW - '90 00'	N1, N5

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
4	<b>Event notification status false - ReaderService</b>				
	1 - Send command on ISO interface to select applet; the initial conditions in clause 6.1.2.6.5 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 6.1.2.6.5 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 6.1.2.6.5 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 6.1.2.6.5 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 6.1.2.6.5 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	<b>Wrong event type</b>				
	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 on ISO interface - 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	

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
<b>6</b>					
<b>Event notification status false - ConnectivityService</b>					
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_RECEPTIONS_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
<b>7</b>					
<b>Event notification status true - ConnectivityService</b>					
	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 TS 102 622 [3].

- 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.

#### 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.

##### 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	<b>message is complete</b>				
	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	<b>message of type event - Card emulation</b>				
	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	<b>message of type event - Connectivity</b>				
	- 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'  third response byte of the R-APDU data is '40'	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 TS 102 622 [3] for further information. It returns the message instruction, b8,b7 of the return value are set to zero.

## 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	<b>message of type event: Card emulation</b>				
	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	<b>message of type response: Card emulation</b>				
	- 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'	- EVT_SEND_DATA (SW - '90 00') - ANY_GET_PARAMETER(ATQA) - 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.

##### 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	<b>Get received message length: CardEmulationService</b>				
	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	<b>Throw Exception</b>				
	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

Test Case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
	10 - EVT_SEND_DATA (INS = '0A')	throwIt() reason = HCI_WRONG_EVENT_TYPE	Throw exception with corresponding reason code	EVT_SEND_DA TA (SW - '90 00')	N1
	11 - EVT_SEND_DATA (INS = '0B')	throwIt() reason = HCI_WRONG_LISTENER_TYPE	Throw exception with corresponding reason code	EVT_SEND_DA TA (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, HCI_Message 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 [11], [12] and [13] 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: As stated in TS 102 622 [3], a reset of the underlying data link layer shall be transparent to the application layer if the data link layer recovers without any loss of data.
- 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.

##### 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_6.java	3
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	<b>Get context</b>				
	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	<b>Abort transaction</b>				
	Send EVT_SEND_DATA (Select applet) - Send EVT_SEND_DATA (INS = '01') - Send ENVELOPE command on the ISO interface which will trigger EVENT_UNREGISTERED_ENVELOPE in the applet as specified in TS 102 241 [6]	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 modified	EVT_SEND_DATA (SW '90 00')	N4
3	<b>Data link layer recovery</b>				
	- Send EVT_SEND_DATA (Select applet) - Send EVT_SEND_DATA (INS = '01') - reset the underlying data link layer while receiving the response	Service = CardEmulationService Listener = CardEmulationListener activateEvent(EVENT_ON_SEND_DATA) during the installation onCallback()  prepareAndSendSendDataEvent() data = {0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0, 0x00, 0x00, 0x00, ..., 0x90, 0x00}  length = 256B	No exception shall be thrown	- EVT_SEND_DATA (SW - '90 00') in the response to selecting applet - EVT_SEND_DATA (0x00, 0x00, 0x00, 0x00, 0x00, 0x0, 0x00, 0x00, 0x00, ..., 0x90, 0x00)  No loss of data expected	N5

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
4	<b>Receive messages while waiting for get parameter</b>				
	- 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	EVT_SEND_DATA ()  ANY_GET_PARAMETER(ATQB)  No error after EVT_FIELD_OFF	N1, N2, N6
5	<b>Receive messages while waiting for get parameter</b>				
	- EVT_SEND_DATA (Select applet) - EVT_FIELD_OFF before 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	EVT_SEND_DATA ()  ANY_GET_PARAMETER(ATQA)  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" [10] 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.

## 6.2.1.1.1.3 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.

## 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	<b>Get parameter: Type A</b>				
	1 - EVT_SEND_DATA (Select applet)	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

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
	6 - EVT_SEND_DATA (Select applet)	prepareAndSendGetParameterCommand( ) paramID = PARAM_ID_TYPE_A_CARD_FWI_SFGE	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	<b>Get parameter: Type B</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_DATA_RATE_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	<b>The Contactless Framework is receiving fragmented HCI Message</b>				
	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.HCIE exception 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

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
2	EVT_SEND_DATA (Select applet)	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
	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				

### 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" [10] 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.



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.

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	<b>Send data - Type A</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
2	<b>Send data - Type B</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	<b>Reception of fragmented HCI Message</b>				
	- 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

Test case					
ID	HCI Commnad	API Description	API Expectation	HCI Response	CRR
4	<b>Send no data</b>				
	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	<b>Array Index Out Of Bounds</b>				
	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 [10].

##### 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	<b>Select Applet</b>				
	EVT_SEND_DATA (Select applet)	selectingMessage()	No exception shall be thrown.  selectingMessage()= true	EVT_SEND_DATA (SW - '90 00')	N1, N2
2	<b>Arbitrary Command</b>				
	- 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 getCardRFTType

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 getCardRFTType()
```

## 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	<b>RF type A</b>				
	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	<b>RF type B</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	<b>RF type F</b>				
	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	<b>RF type B'</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\_CEI\_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.
- CRRN3: If the EVENT\_ON\_SEND\_DATA is deactivated for the Applet instance the javacard.framework.APDU class and the process() method shall be used if the current application protocol is APDU based. Otherwise, if the EVENT\_ON\_SEND\_DATA is deactivated and the current application protocol is not APDU based the process() method will not be invoked, i.e. the Applet will not receive any messages from the external reader.
- 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_CEI_Ocb_1.java	1
Api_2_CEI_Ocb_2.java	2
Api_2_CEI_Ocb_4.java	3
Api_2_CEI_Ocb_1.java	4
Api_2_CEI_Ocb_2.java	5
Api_2_CEI_Ocb_3.java	6
Api_2_CEI_Ocb_5.java	7-1
Api_2_CEI_Ocb_6.java	7-2
Api_2_CEI_Ocb_7.java	7-3
Api_2_CEI_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	<b>EVT_SEND_DATA - Type A</b>				
	- EVT_SEND_DATA A(INS = '14')	HCIListener = CardEmulationListener	- No exception shall be thrown	- EVT_SEND_DATA (SW- '90 00')	N1, N2,
	- EVT_SEND_DATA A(INS = '24')-	HCIService = CardEmulationService activateEvent() event = CardEmulationListener.EVENT_ON_ SEND_DATA	- No exception shall be thrown	- EVT_SEND_DATA (SW- '90 00')	N3, N4,
	- deselect the applet - select the applet		onCallback() method shall notice the event EVENT_ON_SEND_D ATA	- No response or no data in the response to the select applet command	

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
<b>EVENT_HCI_RECEPTIONS_FAILED - Type A</b>					
2	- EVT_SEND_DATA(INS = '11') - EVT_SEND_DATA(INS = '14') - - deselect/select the applet - EVT_SEND_DATA(INS = '11', RF error indicator is set to '01') - - deselect/select the applet - EVT_SEND_DATA(INS = '21')	HCIListener = CardEmulationListener HCIService = CardEmulationService activateEvent() event = HCIListener.EVENT_HCI_RECEPTIONS_FAILED	No exception shall be thrown onCallback() method shall notice the event	-EVT_SEND_DATA (SW- '90 00') - EVT_SEND_DATA (SW- '90 00') - EVT_SEND_DATA (SW- '90 00') - Response to RF error message not verified - EVT_SEND_DATA (SW- '90 00') - EVT_SEND_DATA (SW- '90 00')	N1, N2, N3, N4
<b>EVENT_GET_PARAMETER_RESPONSE - Type A</b>					
3	- 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	No exception shall be thrown onCallback() shall receive ANY_OK as EVENT_GET_PARAMETER_RESPONSE	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')	N1, N2, N3, N4,
<b>EVT_SEND_DATA - Type B</b>					
4	- 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	- No exception shall be thrown - 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') - No response or no data in the response to the select applet command	N1, N2, N3, N4,

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
5	<b>EVENT_HCI_RECEPTIONS_FAILED - Type B</b>				
	- EVT_SEND_DATA (INS = '11') - EVT_SEND_DATA (INS = '14') - deselect/select the applet - EVT_SEND_DATA (INS = '11', RF error indicator is set to '01') - deselect/select the applet - EVT_SEND_DATA (INS = '21')	HCIListener = CardEmulationListener HCIService = CardEmulationService  activateEvent () event = HCIListener.EVENT_HCI_RECEPTIONS_FAILED	No exception shall be thrown  onCallback() method shall notice the event	-EVT_SEND_DATA (SW- '90 00') - EVT_SEND_DATA (SW- '90 00') - EVT_SEND_DATA (SW- '90 00')  - Response to RF error message not verified  - EVT_SEND_DATA (SW- '90 00')  - EVT_SEND_DATA (SW- '90 00')	N1, N2, N3, N4
6	<b>EVENT_GET_PARAMETER_RESPONSE - Type B</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	No exception shall be thrown.  onCallback() shall receive ANY_OK as EVENT_GET_PARAMETER_RESPONSE	- 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')	N1, N2, N3, N4
7	<b>Deactivate events</b>				
	1 - EVT_SEND_DATA (INS='11') - deselect/select the applet - EVT_SEND_DATA (INS='20', RF error indicator is set to '01') - deselect/select the applet - EVT_SEND_DATA (INS='20')	HCIService = CardEmulationService Event has been successfully activated  deactivateEvent () event = HCIListener.EVENT_HCI_RECEPTIONS_FAILED	No exception shall be thrown.  onCallback() method shall not notice the event	EVT_SEND_DATA (SW- '90 00')  - EVT_SEND_DATA (SW- '90 00')  - Response to RF error message not verified  - EVT_SEND_DATA (SW- '90 00')  - EVT_SEND_DATA (SW- '90 00')	N2

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
	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\_CN\_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 TS 102 223 [7].



- 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 TS 102 241 [6] is then available to the Applet when that Applet instance is triggered with the processToolkit() method defined in TS 102 241 [6].
- 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 [3] for the connectivity service.
- CRRN4: The Contactless Framework shall only send the HCI event EVT\_CONNECTIVITY or EVT\_TRANSACTION specified by the HCI protocol [3] to an Applet instance, when it is the selected Applet, which is in card emulation mode or when it is in the state ACTIVATED (according to "GlobalPlatform Amendment C" [10]) for the reader mode.
- CRRN5: The ProactiveHandler defined in TS 102 241 [6] 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 [6] (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.

#### 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

#### 6.2.2.1.1.3 Initial condition

- 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' indicates that class m is supported.
- In terminal profile 5th byte, b1 indicates 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.

## 6.2.2.1.1.4 Test procedure

Test case					
ID	HCI commands	API Description	API Expectation	HCI Response	CRR
1	<b>Send HCI Connectivity Event</b>				
	- 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	N1 N3
2	<b>HCI interface disabled</b>				
	1 - Precondition: Contactless functionality state is disabled in the UICC as defined in TS 102 223 [7]. 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 - Precondition: Contactless functionality state 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: Contactless functionality state is enabled again in the UICC as defined in Global Platform Amendment C	setCommunicationInterface() API method of Global Platform Amendment C [10] is used to disable HCI interface in the CRS  Listener = CardEmulationListener process(): prepareAndSendConnectivityEvent()  setCommunicationInterface() API method of Global Platform Amendment C [10] 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	<b>Wrong Precondition Connectivity Event</b>				
	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	<b>No Proactive Handler</b>				
	- 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	<b>No Proactive Handler onCallback</b>				
	- 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

Test case					
ID	HCI commands	API Description	API Expectation	HCI Response	CRR
6	Use proactive functionality				
	- Send EVT_SEND_DATA with INS = '01' - Send ENVELOPE (HCI Connectivity) on ISO interface after receiving HCI Connectivity Event	<pre> Listener = CardEmulationListener process(): prepareAndSendConnectivityEvent() register for EVENT_EVENT_DOWNLOAD_HCI_CONNECTIVITY  processToolkit(): use arbitrary method of the UICC API (TS 102 241 [6])           </pre>	No exception shall be thrown	- EVT_SEND_DATA (SW – '9000')  - HCI event EVT_CONNECTIVITY sent  - response to ENVELOPE	N1 N2 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\_CN\_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 [3] for the connectivity service.
- CRRN3: The Contactless Framework shall only send the HCI event EVT\_CONNECTIVITY or EVT\_TRANSACTION specified by the HCI protocol [3] to an Applet instance, when it is the selected Applet, which is in card emulation mode or when it is in the state ACTIVATED (according to "GlobalPlatform Amendment C" [10]) 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 TS 102 622 [3].
- CRRP2: throw java.lang.ArrayIndexOutOfBoundsException - if operation would cause access of data outside array bounds.
- CRRP3: throw java.lang.NullPointerException - if parameters is null.

## 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.

## 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_3.java	5

## 6.2.2.1.2.3 Initial condition

- 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' indicates that class m is supported.
- In terminal profile 5th byte, b1 indicates 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.

## 6.2.2.1.2.4 Test procedure

Test case					
ID	HCI commands	API Description	API Expectation	HCI Response	CRR
1	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	N1, N2,

Test case					
ID	HCI commands	API Description	API Expectation	HCI Response	CRR
<b>2</b>					
<b>HCI interface disabled</b>					
1	Precondition: Contactless functionality state is disabled in the UICC as defined in TS 102 223 [7]. 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	Precondition: Contactless functionality state 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: Contactless functionality state is enabled again in the UICC as defined in Global Platform Amendment C	setCommunicationInterface() API method of Global Platform Amendment C [10] is used to disable HCI interface  extends CardEmulationListener process(): prepareAndSendTransactionEvent()  setCommunicationInterface() API method of Global Platform Amendment C [10] 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
<b>3</b>					
<b>Wrong conditions</b>					
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	
<b>4</b>					
<b>Wrong parameters</b>					
1 - Send EVT_SEND_DATA with INS = '01'	extends CardEmulationListener process():  prepareAndSendTransactionEvent() aid = 'A0000000901...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_DATA with INS = '02'	extends CardEmulationListener process():  prepareAndSendTransactionEvent() aid = 'A0000000901...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	

Test case					
ID	HCI commands	API Description	API Expectation	HCI Response	CRR
	3 - Send EVT_SEND_DATA Awith INS = '03'	extends CardEmulationListener process():  prepareAndSendTransactionEvent() aid = 'A00000000901...01' aidOffset = 0 aidLen = 16 parameters = '01...01' parametersOffset = 20 parametersLen = 10	throw java.lang.ArrayIndexOut OfBoundsException	EVT_SEND_DATA TA (SW - '90 00') No EVT_TRANSACTION shall be sent	P2, P3
	4 - Send EVT_SEND_DATA Awith INS = '04'	extends CardEmulationListener process():  prepareAndSendTransactionEvent() aid = 'A00000000901...01' aidOffset = 0 aidLen = 16 parameters = '01...01' parametersOffset = 0 parametersLen = 20	throw java.lang.ArrayIndexOut OfBoundsException	EVT_SEND_DATA TA (SW - '90 00') No EVT_TRANSACTION shall be sent	P2, P3
	5 - Send EVT_SEND_DATA Awith INS = '05'	extends CardEmulationListener process():  prepareAndSendTransactionEvent() aid = null aidOffset = 0 aidLen = 16 parameters = '01...01' parametersOffset = 0 parametersLen = 10	throw java.lang.NullPointerExc eption	EVT_SEND_DATA TA (SW - '90 00') No EVT_TRANSACTION shall be sent	P2, P3
	6 - Send EVT_SEND_DATA AINS = '06'	extends CardEmulationListener process():  prepareAndSendTransactionEvent() aid = 'A00000000901...01' aidOffset = 0 aidLen = 16 parameters = null parametersOffset = 0 parametersLen = 10	throw java.lang.NullPointerExc eption	EVT_SEND_DATA TA (SW - '90 00') No EVT_TRANSACTION shall be sent	P2, P3
5	<b>Wrong Precondition Transaction Event</b>				
	- Send EVT_SEND_DATA A (INS = '01')	Service = ConnectivityService Listener = ConnectivityListener  process() ConnectivityService. prepareAndSendTransactionEvent()	Shall throw exception	EVT_SEND_DATA TA (SW - '90 00') No EVT_TRANSACTION shall be sent	N3

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

Test Area Reference: Api\_2\_CN\_S\_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 [3] for the connectivity service.

## 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 TS 102 622 [3].
- CRRP2: throw java.lang.ArrayIndexOutOfBoundsException - if operation would cause access of data outside array bounds.
- CRRP3: throw java.lang.NullPointerException - if parameters is null.

## 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.

## 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

## 6.2.2.1.3.3 Initial condition

- 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' indicates that class m is supported.
- In terminal profile 5th byte, b1 indicates 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.

## 6.2.2.1.3.4

## Test procedure

Test case					
ID	HCI commands	API Description	API Expectation	HCI Response	CRR
1	<b>Send HCI Transaction Event</b>				
	Send EVT_SEND_DATA with INS = '01'	extends CardEmulationListener process(): prepareAndSendTransactionEvent() parameters = '01...01' parametersOffset = 0 parametersLen = 10	No exception shall be thrown	SW - '90 00'  HCI event EVT_TRANSACTION sent	N1, N2
2	<b>HCI interface disabled</b>				
	1 - Precondition: Contactless functionality state is disabled in the UICC as defined in TS 102 223 [7]. 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 - Precondition: Contactless functionality state 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: Contactless functionality state is enabled again in the UICC as defined in Global Platform Amendment C	setCommunicationInterface() API method of Global Platform Amendment C [10] is used to disable HCI interface  extends CardEmulationListener process(): prepareAndSendTransactionEvent()  setCommunicationInterface() API method of Global Platform Amendment C [10] 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	<b>Wrong conditions</b>				
	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	<b>Wrong parameters</b>				
	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



Test case					
ID	HCI commands	API Description	API Expectation	HCI Response	CRR
	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

## 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.

## 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_2.java	5-1
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	<b>MULTIPLE_TARGET_STATUS is received - Type A</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
2	<b>MULTIPLE_TARGET_STATUS is received - Type B</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	<b>RESP_WR_RF_ERROR is received</b>				
	- 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	<b>RESP_ANY_E_TIMEOUT is received</b>				
	- 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

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
5	<b>HCI interface is disabled</b>				
	1 - Precondition: Contactless functionality state is disabled in the UICC as defined in TS 102 223 [7] - Send APDU to select the applet. - Send EVT_TARGET_DISCOVERED (status = '03') - Send APDU on ISO interface ('01')	restartReaderModeProcedure()  event = EVT_END_OPERATION event = EVT_READER_REQUESTED	Shall throw uicc.hci.framework.HCIException with error code HCI_CURRENTLY_DISABLED	- SW - '90 03' No EVT_READER_REQUESTED shall be sent	C1
	2 - Precondition: Contactless functionality state 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: Contactless functionality state is enabled again in the UICC as defined in Global Platform Amendment C	setCommunicationInterface() API method of Global Platform Amendment C [10] is used to disable HCI interface  - restartReaderModeProcedure()  event = EVT_END_OPERATION event = EVT_READER_REQUESTED  setCommunicationInterface() API method of Global Platform Amendment C [10] 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 [11], [12] and [13]) 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.

## 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: throw HCIException with error code reason HCI\_FRAGMENTED\_MESSAGE\_ONGOING if the Contactless Framework is still receiving a fragmented HCI Message.

## 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.
- TheUICC 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	<b>Send Data - Type A</b>				
	- Send EVT_TARGET_DI SCOVERED (status = '00') - Send ANY_OK	prepareAndSendWriteXchgDataCommand() timeout = -1 data = '01...01' offset = 0 len = 10	No exception shall be thrown	- command WR_XCHG_DAT A Is received	N1, N2, N3, N4
2	<b>Send Data - Type B</b>				
	- Send EVT_TARGET_DI SCOVERED (status = '00') - Send ANY_OK	prepareAndSendWriteXchgDataCommand() timeout = -1 data = '01...01' offset = 0 len = 10	No exception shall be thrown	- command WR_XCHG_DAT A Is received	N1, N2, N3, N4
3	<b>HCI interface is disabled</b>				
	1 - Precondition: Contactless functionality state is disabled in the UICC as defined in TS 102 223 [7] - Send APDU on ISO interface to select the applet - Send EVT_TARGET_DI SCOVERED (status = '00') - Send APDU on ISO interface (INS = '11')	prepareAndSendWriteXchgDataCommand() timeout = 14 data = '01...01' offset = 0 len = 10	Shall throw uicc.hci.framework.HCIEx ception with error code HCI_CURRENTLY_DISA BLED	- SW - '90 01'	C1
	2 - Precondition: Contactless functionality state 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_DI SCOVERED (status = '00') - Send APDU on ISO interface (INS = '01') - Postcondition: Contactless functionality state is enabled again in the UICC as defined in Global Platform Amendment C	setCommunicationInterface() API method of Global Platform Amendment C [10] is used to disable HCI interface  prepareAndSendWriteXchgDataCommand() timeout = 14 data = '01...01' offset = 0 len = 10  setCommunicationInterface() API method of Global Platform Amendment C [10] is used to enable again HCI interface	Shall throw uicc.hci.framework.HCIEx ception with error code HCI_CURRENTLY_DISA BLED	- SW - '90 01'	C1

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
4	<b>Data is null</b>				
	- 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	<b>ArrayIndexOutOfBoundsException Exception</b>				
	- 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	<b>ArrayIndexOutOfBoundsException Exception</b>				
	- Send APDU on ISO interface (INS = '03') - Send EVT_TARGET_DISCOVERED (status = '00') - Send APDU on ISO interface (INS = '13')	prepareAndSendWriteXchgDataCommand() timeout = 14 data = '01...01' offset = 0 len = 11	Shall throw java.lang.ArrayIndexOutOfBoundsException	- SW - '90 04'	P2
7	<b>Timeout has illegal value</b>				
	- Send EVT_TARGET_DISCOVERED (status = 00) - Send APDU on ISO interface (INS = '01')	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: throw HCIException with error code reason HCI\_FRAGMENTED\_MESSAGE\_ONGOING if the Contactless Framework is still receiving a fragmented HCI Message.

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_2.java	3-1
Api_2_RMm_Sgp_3.java	3-2
Api_2_RMm_Sgp_2.java	4
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	<b>Get Parameter: Type A</b>				
	1 - Send APDU on ISO interface (INS = '01') - Send EVT_TARGET_DISCOVERED (status = '00') - 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 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

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
	3 - Send APDU on ISO interface (INS = '03') - Send EVT_TARGET_DISCOVERED (status = '00') - 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 EVT_TARGET_DISCOVERED (status = '00') - 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 EVT_TARGET_DISCOVERED (status = '00') - 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 EVT_TARGET_DISCOVERED (status = '00') - 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	<b>Get Parameter: Type B</b>				
	1 - Send APDU on ISO interface (INS = '07') - Send EVT_TARGET_DISCOVERED (status = '00') - 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 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 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



Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
	4 - Send APDU on ISO interface (INS = '0A') - Send EVT_TARGET_DISCOVERED (status = '00') - 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 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	<b>HCI interface is disabled</b>				
	1 - Precondition: Contactless functionality state is disabled in the UICC as defined in TS 102 223 [7] - Send APDU on ISO interface (INS = '01') - Send EVT_TARGET_DISCOVERED (status = '00') - Send APDU on ISO interface (INS = '11')	- prepareAndSendGetParameterCommand() paramid = PARAM_ID_TYPE_A_READER_UID	Shall throw uicc.hci.framework.HCIException with error code HCI_CURRENTLY_DISABLED	- SW - '90 03'	C1
	2 - Precondition: Contactless functionality state 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: Contactless functionality state is enabled again in the UICC as defined in Global Platform Amendment C	setCommunicationInterface() API method of Global Platform Amendment C [10] is used to disable HCI interface  prepareAndSendGetParameterCommand() paramid = PARAM_ID_TYPE_A_READER_UID  setCommunicationInterface() API method of Global Platform Amendment C [10] is used to enable again HCI interface	Shall throw uicc.hci.framework.HCIException with error code HCI_CURRENTLY_DISABLED	- SW - '90 03'	C1

Test case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
4	<b>The Contactless Framework is receiving fragmented HCI Message</b>				
	- Send APDU on ISO interface (INS = '02') - Send EVT_TARGET_DISCOVERED (status = '00')  - Send ANY_OK as fragmented HCI Message (data length = supported buffer size + 5 bytes, message not complete)  - Send APDU on ISO interface (INS = '12')	- prepareAndSendGetParameterCommand() paramid = PARAM_ID_TYPE_A_READER_UID	Shall throw uicc.hci.framework.HCIException with error code HCI_FRAGMENTED_MESSAGE_Ongoing	- SW - '90 04'	C3
5	<b>The parameter has an illegal value</b>				
	- 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	3
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	7
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
<b>EVENT_TARGET_DISCOVERED - Type A</b>					
1	Send command on ISO interface to select applet. - select applet APDU (INS = '10') - 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') is received	N1, N2
<b>EVENT_WRITE_EXCHANGE_DATA_RESPONSE - Type A</b>					
2	Send command on ISO interface to select applet. - select applet APDU(INS = '11') - 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
<b>HCIListener.EVENT_HCI_TRANSMISSION_FAILED - Type A</b>					
3	Send command on ISO interface to select applet. - select applet APDU (INS = '12') - Send EVT_TARGET_DISCOVERED (status = '00') - Send ANY_OK ('11 22 33 44 00'), RF error indicator is set to '01' - Send APDU (INS = '2') on ISO interface	HCIService = ReaderService activateEvent() event = HCIListener.EVENT_HCI_TRANSMISSION_FAILED	No exception shall be thrown.  onCallback() method shall notice the event	- No WR_XCHG_DATA () is received.  - SW - '90 00'	N1, N2.
<b>EVENT_GET_PARAMETER_RESPONSE - Type A</b>					
4	Send command on ISO interface to select applet. - select applet APDU(INS = '13') - 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 is received	N1, N2

Test Case					
ID	HCI Command	API Description	API Expectation	HCI Response	CRR
<b>EVENT_TARGET_DISCOVERED - Type B</b>					
5	Send command on ISO interface to select applet. - select applet APDU (INS = '10') - 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') is received	N1, N2
<b>EVENT_WRITE_EXCHANGE_DATA_RESPONSE - Type B</b>					
6	Send command on ISO interface to select applet. - select applet APDU(INS = '11') - 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
<b>HCIListener.EVENT_HCI_TRANSMISSION_FAILED - Type B</b>					
7	Send command on ISO interface to select applet. - select applet APDU (INS = '12') - Send EVT_TARGET_DISCOVERED (status = '00') - Send ANY_OK ('11 22 33 44 00'), RF error indicator is set to '01' - Send APDU (INS = '2') on ISO interface	HCIService = ReaderService activateEvent() event = HCIListener.EVENT_HCI_TRANSMISSION_FAILED	No exception shall be thrown.  onCallback() method shall notice the event	- No WR_XCHG_DATA () is received.  - SW - '90 00'	N1, N2.
<b>EVENT_GET_PARAMETER_RESPONSE - Type B</b>					
8	Send command on ISO interface to select applet. - select applet APDU(INS = '13') - 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 is received	N1, N2

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

### A.1 HCI framework

Class name	Acronyms
HCIDevice	Hdv
HCIService	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_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



Applet name	AID
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_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 21 11 01 02
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_CEI_Ocb	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 00 00

Applet name	AID
Api_2_CEI_Ocb_1.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 01 02
Api_2_CEI_Ocb_2.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 02 02
Api_2_CEI_Ocb_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 03 02
Api_2_CEI_Ocb_4.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 04 02
Api_2_CEI_Ocb_5.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 05 02
Api_2_CEI_Ocb_6.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 06 02
Api_2_CEI_Ocb_7.java	A0 00 00 00 09 00 05 FF FF FF FF 89 25 05 07 02
Api_2_CEI_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_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_3.java	A0 00 00 00 09 00 05 FF FF FF FF 89 26 02 03 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_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

### 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 Package uicc.hci

Requirement	Reference
A mechanism to launch an application on the terminal host that is related to the Applet running in the UICC	Api_2_CNs_Ste Api_2_CNs_Stt
The Contactless Framework shall add the AID of the calling Applet instance to the HCI event passed to the CLF	Api_2_CNs_Stt
A mechanism that allows a UICC to start a proactive session as defined in TS 102 223 [7]	Api_2_CNs_Sce
Content of the buffer outside of the area which is used for the HCI message part is undefined	Api_1_Hme_Mrb
This method throw HCIException with error code reason HCI_RESOURCES_NOT_AVAILABLE if the contactless framework does not have enough resources to process the command	Api_2_RMm_Srx Api_2_RMm_Sgp Api_2_CNs_Sce Api_2_CEm_Ssd Api_2_CNs_Ste Api_2_CNs_Stt

#### C.1.2 TS 102 705 prose part

Requirement	Reference
To release the CLF control at the end of a transaction an Applet shall deactivate the ReaderListener.EVENT_TARGET_DISCOVERED	4.3
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 [3]	4.2

### C.2 FFS requirements

#### C.2.1 Package uicc.hci

Requirement	Reference
The application may use the whole receive buffer for its internal purposes. If the buffer is used for manipulation of sensitive data it shall be cleared by the Applet before returning to the contactless framework	Api_1_Hme_Mrb
Throw HCIException with error code reason HCI_FRAGMENTED_MESSAGE_ONGOING if the Contactless Framework is still receiving a fragmented HCI Message	Api_2_RMm_Srx
Throw java.lang.ArrayIndexOutOfBoundsException if operation would cause access of data outside array bounds	Api_2_RMm_Srx
Throws HCIException with the reason code HCI_CURRENTLY_DISABLED if the HCI interface was disabled	Api_2_CEm_Sgp Api_2_CEm_Ssd
Throw HCIException with error code reason HCI_INVALID_LENGTH if the parameter length or the AID is not compliant to TS 102 622 [3]	Api_2_CNs_Ste Api_2_CNs_Stt
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	Api_1_Hme_Min
If the message is not complete then the returned value is the actual HCI message fragment length	Api_1_Hme_Mrl
HCIException with reason code HCI_CONDITIONS_NOT_SATISFIED if one or more conditions to activate the event are not satisfied	Api_1_Hsr_Ace
All other HCI messages shall be delivered to the Applet instance in the same order as they were received by the Contactless Framework	Api_1_Hln_Ocb

## C.2.2 TS 102 705 prose part

Requirement	Reference
When the contactless interface is disabled (cf. "state of contactless functionality" in TS 102 223 [7] and setCommunicationInterface() API method of "GlobalPlatform Amendment C" [10]), the Contactless Framework shall throw an HCIException with reason code HCI_CURRENTLY_DISABLED	4.1
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
To be able to receive and send messages over the contactless interface in reader mode the applet shall activate the ReaderListener.EVENT_TARGET_DISCOVERED	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 TS 102 226 [9]	4.3
Reader mode Applets shall follow the extended lifecycle model that is defined in "GlobalPlatform Amendment C" [10] 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" [10]) at any time	4.3
When the state of a reader mode Applet changes to lifecycle ACTIVATED (according to "GlobalPlatform Amendment C" [10]) 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): uicc.contactless.test.util package and contactless interfaces

The use of Java as a test scripting language is optional. However, if Java is used as the test scripting language, then the test descriptions shall be implemented according to annex E.

The content of this annex is TBD.



---

## Annex F (normative): Test Area files

The use of Java as a test scripting language is optional. However, if Java is used as the test scripting language, then the test descriptions shall be implemented according to annex F.

The content of this annex is TBD.

---

## Annex G (informative): Core specification version information

Unless otherwise specified, the versions of TS 102 705 [1] from which conformance requirements have been extracted are as follows.

<b>Release</b>	<b>Latest version from which conformance requirements have been extracted</b>
9	V9.3.0

## 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
2013-02	SCP#58	SCP(13)000039	002	-	F	Clarification of initial conditions activity performed by UICC	9.0.0	9.1.0
2013-02	SCP#58	SCP(13)000040	003	-	F	Test case 6.1.5.1: corrected definition of unrecognised Envelope	9.0.0	9.1.0
2013-02	SCP#58	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
2013-02	SCP#58	SCP(13)000042	005	-	F	Clarification of prepareAndSendGetParameterCommand test cases	9.0.0	9.1.0
2013-02	SCP#58	SCP(13)000043	006	-	F	Correction of test suite files	9.0.0	9.1.0
2013-02	SCP#58	SCP(13)000044	007	-	F	Correction of wrong Events deactivation	9.0.0	9.1.0
2013-02	SCP#58	SCP(13)000046	008	-	F	Clarification of initial conditions related to power mode	9.0.0	9.1.0
2013-02	SCP#58	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
2013-04	SCP#59	SCP(13)000087r1	012	-	F	Clarification of EVT_READER_REQUESTED transmission	9.0.0	9.1.0
2013-04	SCP#59	SCP(13)000088	013	1	F	Correction of applet selection in TC 6.2.1.1.1 ID3	9.0.0	9.1.0
2013-04	SCP#59	SCP(13)000089	014	-	F	Correction of API expected received data in TC 6.1.3.7	9.0.0	9.1.0
2013-04	SCP#59	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
2013-04	SCP#59	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
2013-04	SCP#59	SCP(13)000092	017	-	F	Addition of Annexes E and F contents into TS 103 115	9.0.0	9.1.0
2013-04	SCP#59	SCP(13)000093	018	-	F	Corrections to avoid failed deletions of test applets	9.0.0	9.1.0
2013-04	SCP#59	SCP(13)000037	019	-	F	Corrections on Reader Mode applets	9.0.0	9.1.0

---

## History

<b>Document history</b>		
V9.0.0	April 2013	Publication
V9.1.0	July 2013	Publication